Digital Forensic Analysis of Amazon Linux EC2 Instances

GIAC GFCA Gold Certification

Author: Kenneth G. Hartman, ken@kennethghartman.com
Advisor: Sally Vandeven

Accepted: January 2018

Abstract

Companies continue to shift business-critical workloads to cloud services such as Amazon Web Services Elastic Cloud Computing (EC2). With demand for skilled security engineers at an all-time high, many organizations do not have the capability to do an adequate forensic analysis to determine the root cause of an intrusion or to identify indicators of compromise. To help organizations improve their incident response capability, this paper presents specific tactics for the forensic analysis of Amazon Linux that align with the SANS "Finding Malware – Step by Step" process for Microsoft Windows.

1. Introduction

Companies are shifting more and more of their business-critical workloads to the cloud and often, the choice is Amazon Web Services (AWS) Elastic Cloud Computing (EC2). This is supported by industry metrics that indicate that AWS is the undisputed market leader with a 47.1% market share, followed by Microsoft Azure at 10.0% and Google Cloud Platform with 3.95% (Coles, n.d.). Infrastructure as a Service (IaaS) offerings, such as EC2, are projected to experience a combined annual growth rate of 29.7% (Coles, n.d.). Therefore, it behooves companies to develop security capabilities in response to this important trend.

Staffing and skills gaps hamper security operations at many organizations. Some predict that by 2019, there will be a shortage of two million information security professionals (Kauflin, 2917). As a result, many organizations do not have the capability to do an adequate forensic analysis to determine the root cause of an intrusion or to identify indicators of compromise (Oltsik, 2017a). The shortage of skilled engineers is exacerbated by the fact that the number of connected devices is exploding at an exponential rate—requiring organizations to find innovative ways to scale up security operations (Oltsik, 2017b).

SANS DFIR (digital-forensics.sans.org) provides training and resources to equip companies and organizations with necessary skills and tools to perform intrusion analysis and digital forensics. Two examples are the SANS FOR508: Advanced Incident Response training course (SANS Institute, n.d.a) and the SANS Investigative Forensic Toolkit (SIFT) Workstation (SANS DFIR, n.d.). ThreatResponse is an open-source project team that develops tools and promotes techniques focused on improving incident response in Amazon Web Services (Krug, McCormack, Ferrier, & Parr, n.d.).

To help organizations improve their forensic capabilities in the cloud, this paper presents specific tactics for the forensic analysis of Amazon Linux that align with the SANS "Finding Malware – Step by Step" process for Microsoft Windows and leverage the tools from the ThreatResponse project. Improved cloud forensic capabilities help an organization to adapt to adversaries and result in improved overall security.

Kenneth G. Hartman,
ken@kennethghartman.com

2. Incident Response Resources

A review of the information currently available on the Internet pertaining to the forensic analysis of EC2 instances focuses primarily on how to acquire drive and memory images from an EC2 instance (Dykstra, 2013; Olsen, 2014). There are also ample references on how to obtain valuable information from the logs generated by AWS services, such as CloudTrail (McGeehan, 2016). When enabled, AWS CloudTrail creates event logs of all interactions with other AWS services including the Management Console, command line tools and the application programing interface (Amazon Web Services, n.d.a). However, there is not a concise document that provides guidance on what to do with the forensic images of an Amazon Linux EC2 Instance once the images are acquired. This document aims to be one such resource.

This paper assumes a basic understanding of incident response procedures, covered by documents such as the SANS paper titled, *Incident Handler's Handbook* (Kral, 2011) and *NIST SP 800-61R2* (NIST special publication 800-61 revision 2, computer security incident handling guide, 2012). For prerequisite knowledge on incident handling in AWS, the following two documents are highly recommended:

- "Incident Response in Amazon EC2: First Responders Guide to Security Incidents in the Cloud" (Arnold, 2016), and
- "Hardening AWS Environments and Automating Incident Response for AWS Compromises" (Krug, 2016).

3. Incident Response Life Cycle Model

To understand the importance of the role of digital forensics in incident response, it is helpful to consider the Incident Response Life Cycle. Once such model presented in NIST SP 800-61R2 is depicted below in Figure 1. Another very popular model is the traditional PICERL model, which has the following six phases—Preparation, Identification, Containment, Eradication, Recovery, and Lessons Learned (Kral, 2011). However, the NIST model shown in Figure 1 illustrates the iterative nature of the Digital Forensics Incident Response (DFIR) process best, especially when dealing with ephemeral assets in the public cloud.

Kenneth G. Hartman,
ken@kennethghartman.com

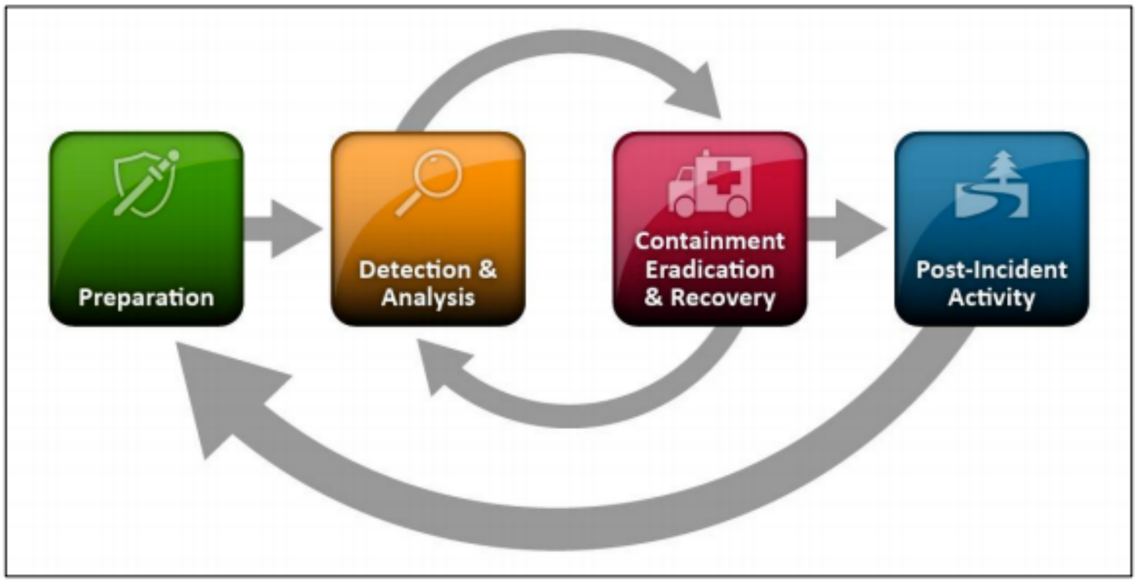


***Figure 1** – Incident Response Life Cycle as Depicted in NIST 800 SP-61R2*

In the ***Detection & Analysis*** phase, knowledge is produced to be fed into ***Containment, Eradication & Recovery*** phase to ensure that the scope of the incident is known and properly contained and eradicated. This knowledge is the synthesis of forensic analysis, security event information management (SEIM) data, intrusion detection systems (IDS) events, and external data feeds, among others. Discoveries resulting from the activities performed in the *Containment, Eradication and Recovery* phase feed back into the *Detection & Analysis* phase, including the identification of additional systems requiring forensic analysis. Similarly, lessons learned as a result of incident handling should improve an organization's preparedness (NIST SP800-61R2).

Digital forensic techniques are typically used to gather and preserve evidence, reconstruct events, determine the *how*, *when*, and *where* the incident occurred (NIST SP800-61R2) and to generate threat information. Threat information helps an organization protect itself and includes *indicators of compromise* (IOC) and *tactics, techniques, and procedures* (TTPs). Questions of attribution, i.e. *who* and *why*, are very difficult (Goutam, 2015), a distraction that is best left to law enforcement, and do not improve an organization's security posture, according to at least one CISO (Stilgherrian, 2015). Therefore, this paper focuses on gathering evidence, reconstructing events, and threat information.

Kenneth G. Hartman,
ken@kennethghartman.com

4. Acquiring Forensic Evidence in EC2

The virtual hard drives used by an EC2 virtual machine are referred to as an *Elastic Block Store* (EBS) volume (Amazon Web Services, n.d.b). An EBS volume is used to boot the operating system, but an EC2 instance can have one more than one volume. The Elastic Cloud Computing service provides a facility for making a snapshot of an EBS volume that can be used to create a second EBS volume which may then be used for forensic analysis in a manner similar to the approach of cloning physical hard drives in a conventional forensic investigation (Olsen, 2014).

Snapshots can be shared with other AWS accounts, and this is a good practice, particularly if the security team has a dedicated account used for incident handling. The advantage of this approach is that once an EBS Volume is created in the new AWS account, it is protected from anyone who does not have rights to the incident handling account and the forensic analysis can be performed in a presumably uncompromised environment. In the discussion that follows, the dedicated account will be referred to as **incident-response.** However, AWS account aliases must be globally unique, so one solution is to prepend the organization name to this phrase in actual practice. For example, “acme-incident-response.”

Snapshots should never be shared publicly, unless great care is taken. This common mistake is an easy way to leak sensitive data and keys (NVTEH, 2017). There is an option to encrypt an EBS volume and it is important to note that snapshots made of an encrypted EBS volume will likewise be encrypted (Amazon Web Services, n.d.b). Therefore, the keys used to encrypt an EBS volume must be made available to the forensicator to analyze the virtual hard drive.

The ThreatResponse `aws_ir` command line tool has a subcommand, `host_compromise`, that automates the process of making the EBS snapshot and will also call another ThreatResponse tool, `MargaritaShotgun`, to create a forensic image of the EC2 instance memory (Krug, 2016). According to Amazon, the only way to capture the memory of an Amazon Linux EC2 instance is via an SSH session, passing the SSH keys to a memory imaging tool such as `MargaritaShotgun` (personal communication, October 17, 2017). Hence, `aws_ir` has command line options to

Kenneth G. Hartman,
ken@kennethghartman.com

provide the SSH key that has root access to the particular EC2 instance being imaged (AWS_IR quickstart, n.d.). Appendix B provides instruction on how to use `aws_ir` to create a EBS Snapshot and forensic memory image in a single command. To speed up incident response, Ephemeral Systems has created publicly-shared Amazon Machine Images (AMI) with the ThreatResponse tools pre-installed (Ephemeral Systems, n.d.).

5. Provision a Forensic Workstation

To manually provision a SIFT Workstation on an AWS EC2 Instance, perform the following steps:

1. Log into the **incident-response** account
2. Launch an EC2 instance using the console. Use the latest Ubuntu AMI in the AWS Marketplace, for example:
   `Ubuntu Server 16.04 LTS (HVM), SSD Volume Type - ami-996372fd"`
3. Choose a **t2.large** or larger instance type.
4. Click "next" as needed to get to the "Configure Security Group" form.
5. Create a new "Incident Response" security group and configure it such that the only ingress rule is to allow SSH from the external IP address by selecting the "My IP" option as shown in Figure 2, below. The egress rule may be left as the default "allow all" so that the system can install software.
6. Choose an existing SSH key or select the option to that the AWS console create a new one.
7. Once the instance launches, SSH to it with the appropriate key.
8. On the Ubuntu EC2 instance, run the following command:

```
wget https://github.com/sans-dfir/sift-
cli/releases/download/v1.5.1/sift-cli-linux
```

NOTE: The version will change over time, so update the path as necessary to ensure the command pulls down the latest version.

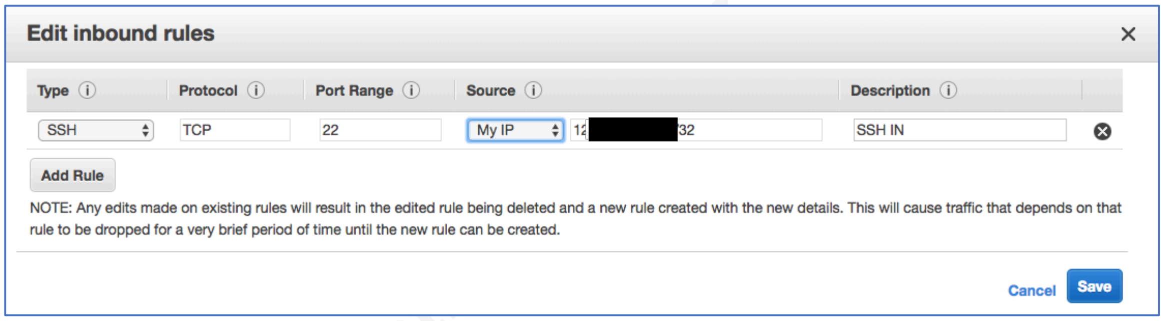
Kenneth G. Hartman,
ken@kennethghartman.com

9. Move and rename the SIFT Install tool:

```
sudo mv sift-cli-linux /usr/local/bin/sift
```

10. Set the proper permissions and run it:

```
sudo chmod 755 /usr/local/bin/sift
sudo sift install
```



Edit inbound rules

Type	Protocol	Port Range	Source		Description
SSH	TCP	22	My IP	12[redacted]32	SSH IN

Add Rule

NOTE: Any edits made on existing rules will result in the edited rule being deleted and a new rule created with the new details. This will cause traffic that depends on that rule to be dropped for a very brief period of time until the new rule can be created.

***Figure 2** – Configure Inbound Rules to Allow SSH from a Single IP Address*

For more information on provisioning an EC2 instance, see *Launching an Instance Using the Launch Instance Wizard* (n.d.) and for complete instructions on installing the SIFT Workstation refer to (SIFT CLI, n.d.). Lastly, a Github repository named SIFTonEC2 contains a script to automatically provision a SIFT workstation on an Elastic Cloud Computing instance (SIFTonEC2, n.d.).

The SIFT installation process may take up to 30 minutes and it is highly recommended that incident handlers provision a new system for each case. One way to streamline this is to make an AMI of the SIFT Workstation before using it (Olsen, 2014). This allows future SIFT instances to be launched from the AMI. However, the instance may still need to be updated. To update a SIFT Workstation, run the following commands:

```
apt-get update
apt-get upgrade
sift update
shutdown -r now  #Reboot
```

6. Attaching Evidence to the SIFT Workstation

Once the SIFT Workstation has been provisioned, the next step is to attach evidence to it and mount the volume for analysis. This process is detailed as follows:

Kenneth G. Hartman,
ken@kennethghartman.com

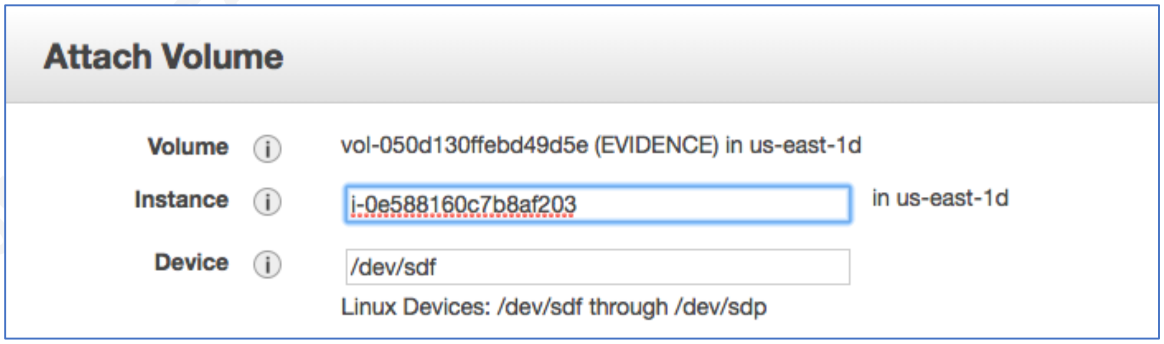
6.1. Create an EBS Volume from a shared snapshot

Find the snapshot in the AWS Console, using the **private snapshot** filter. Next, click on the snapshot and select **Create Volume** from the Actions button. NOTE: Be sure to create the new volume in the same availability zone as the SIFT Workstation.

While creating the volume, add an additional tag with the key set to "Incident" and the value set to the case number for the incident. Set the "Name" tag to "EVIDENCE" to distinguish it from the SIFT Workstation OS Volume or other volumes in the account.

6.2. Attach the new volume to the SIFT Workstation

Select the new volume in the AWS Console. Next, select the **Attach Volume** option from the **Actions** button. The wizard will ask which instance to attach to, so select the instance that was just launched. If the instance is not listed, the wizard may have created the EBS volume in the wrong availability zone. The console will report which device the volume will be attached to, /dev/xvdf, for example (Olsen, 2014).


Attach Volume

Volume vol-050d130ffebd49d5e (EVIDENCE) in us-east-1d

Instance i-0e588160c7b8af203 in us-east-1d

Device /dev/sdf

Linux Devices: /dev/sdf through /dev/sdp

Figure 3 – Attach EVIDENCE volume to SIFT Workstation

6.3. Mount the Evidence Volume as Read Only

Verify the volume was attached as a device using the **lsblk** command as root:

```
root@siftworkstation:/home/ubuntu# lsblk
NAME    MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
xvda    202:0    0   8G  0 disk
└─xvda1 202:1    0   8G  0 part /                ← Root Partition
xvdf    202:80   0   8G  0 disk
└─xvdf1 202:81   0   8G  0 part                  ← Unmounted Partition
```

Kenneth G. Hartman,
ken@kennethghartman.com

The command output above shows that the OS is mounted on partition xvda1 of /dev/xvda. It also shows that a single partition on the new volume that is not mounted (xvdf1). This indicates that the volume is attached, just not mounted yet.

6.4. Determine the format of the partition and mount

Use the **file** command to determine the format of the partition as shown below:

```
root@siftworkstation:/home/ubuntu# file -s /dev/xvdf1
/dev/xvdf1: Linux rev 1.0 ext4 filesystem data, UUID=4967eba2-7f57-4688-8f0b-
069ca22b0b3e (extents) (large files) (huge files
```

Make a directory and mount the evidentiary Linux file system as read-only:

```
mkdir /mnt/linux_mount # Need to make any Linux mount points
mount -o ro /dev/xvdf1 /mnt/linux_mount/
```

If the file system was Microsoft Windows, the following command would be used (Lee, 2009a):

```
 mount –o loop,ro,show_sys_files,streams_interface=windows /dev/xvdf1
/mnt/windows_mount
```

Note that the `windows_mount` directory already exists on the SIFT Workstation, but the `linux_mount` will need to be created.

Verify the partition is mounted using either the **mount** or the **ls** commands:

```
root@siftworkstation:/home/ubuntu# mount | grep "/mnt"
/dev/xvdf1 on /mnt/linux_mount type ext4 (ro,relatime,data=ordered)

root@siftworkstation:/home/ubuntu# ls -als /mnt/linux_mount/
total 128
 4 dr-xr-xr-x  25 root root  4096 Jan 23  2017 .
 4 drwxr-xr-x  17 root root  4096 Sep 25 22:22 ..
 0 -rw-r--r--   1 root root     0 Jan 13  2017 .autorelabel
 4 dr-xr-xr-x   2 root root  4096 Jan 13  2017 bin
 4 dr-xr-xr-x   4 root root  4096 Jan 13  2017 boot
 4 drwxr-xr-x  11 root root  4096 Jan 13  2017 cgroup
 4 drwxr-xr-x   2 root root  4096 Dec 20  2016 dev
 4 drwxr-xr-x  77 root root  4096 Jan 13  2017 etc
 4 drwxr-xr-x   3 root root  4096 Jan 13  2017 home
 4 dr-xr-xr-x   7 root root  4096 Jan 13  2017 lib
(continued...)
```

Kenneth G. Hartman,
ken@kennethghartman.com

6.5. Prepare the EC2 Memory Image and Profile

Assuming that the `aws_ir` script created an image of the EC2 instance memory, the first task is to copy it to the SIFT Workstation from the S3 bucket created when `aws_ir` executed. The memory image will have a “lime” file extension and the name of the bucket will start with “cloud-response.” Copy the image file to `/cases`.

Volatility and Rekall are the leading open-source solutions for memory image analysis, however both require a profile that is specific to the kernel of the system that was imaged. Both projects have repositories of profiles for CentOS, Ubuntu, and other distributions, but do not have any Amazon Linux profiles at the time of this writing. Since there are so many kernel permutations, the documentation for both Volatility and Rekall provide instructions on how to create a custom profile, assuming the analyst has the source code for the kernel headers (Linux profiles, n.d.; Linux support in Rekall, n.d.).

Creating custom profiles is beyond the scope of this paper, however it should be stressed that profiles for all kernels used in production should be prepared in advance. Remember that kernels may be updated as part of the patching process, so maintaining a current inventory is a critical aspect of incident response preparation (Linux profiles, n.d.). Memory analysis will be discussed further in Section 10.8.

7. Perform Forensic Analysis

SANS has published a Digital Forensics & Incident Response Poster that describes a process for finding malware on Windows that is titled “Finding Malware – Step by Step” (SANS DFIR, 2013). The following sections follow that general process, adapting it for Amazon Linux.

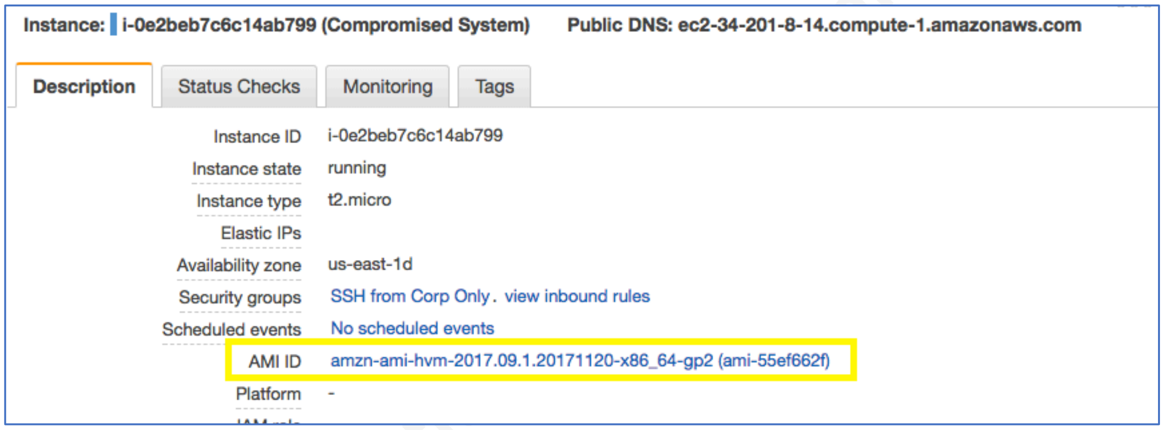
7.1. Evidence Preparation and Data Reduction

Now that the evidence is attached to the SIFT Workstation, a first step is to carve data from the unallocated space and then separate out the files that are known to be good. While it may be tempting to skip the step of creating the hash database, this powerful tactic is easy to do.

Kenneth G. Hartman,
ken@kennethghartman.com

7.1.1. Generate Hash List from AMI

The known files hash list will be generated on the SIFT Workstation from the same AMI that was used to launch the instance as determined by looking at the original metadata associated with the compromised instance. See Figure 4.


Instance: i-0e2beb7c6c14ab799 (Compromised System)   Public DNS: ec2-34-201-8-14.compute-1.amazonaws.com

Instance ID	i-0e2beb7c6c14ab799
Instance state	running
Instance type	t2.micro
Elastic IPs	
Availability zone	us-east-1d
Security groups	SSH from Corp Only. view inbound rules
Scheduled events	No scheduled events
AMI ID	amzn-ami-hvm-2017.09.1.20171120-x86_64-gp2 (ami-55ef662f)
Platform	-

***Figure 4** – Identifying the AMI to use for the Hash Database*

Launch an EC2 Instance based on the AMI and make a snapshot as soon as the status checks are complete but before logging onto the system. Next, make a volume from the snapshot in the same availability zone as the SIFT Workstation. Use a name tag such as "HASH-BASELINE" for both the snapshot and the volume to differentiate these objects from those related to the evidence and the SIFT Workstation itself.

Using the same steps described in Sections 6.3 and 6.4, attach and mount the HASH-BASELINE volume as the third volume on the SIFT Workstation using a unique mount point, such as `/mnt/linux_base`. For example:

```
# mkdir /mnt/linux_base
# mount -o ro /dev/xvdg1 /mnt/linux_base/
# lsblk
NAME    MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
xvda    202:0    0   8G  0 disk
└─xvda1 202:1    0   8G  0 part /
xvdf    202:80   0   8G  0 disk
└─xvdf1 202:81   0   8G  0 part /mnt/linux_mount
xvdg    202:96   0   8G  0 disk
└─xvdg1 202:97   0   8G  0 part /mnt/linux_base
```

It is very useful to have the base AMI mounted as a read-only volume that can be referred to during the investigation as the "known good" state. This is particularly true if the AMI has been hardened and contains custom files that are intended to be there. To

Kenneth G. Hartman,
ken@kennethghartman.com

help with identifying the differences between the volume under investigation and the reference volume, a hash database of all files on the reference volume can be created using `hfind` (Carrier, 2003a) as follows:

```
# find /mnt/linux_base/ -exec /usr/bin/md5sum {} \; > known_files.md5
# hfind -i md5sum known_files.md5
```

7.1.2. Identify files which are new or modified

Next, create a hash list of files for the volume under investigation ("`investigate_files.md5`") and use that list to create an additional list of the files that are new or have changed ("`changed_files.txt`"). Scanning `changed_files.txt` may reveal interesting patterns for further investigation.

```
# find /mnt/linux_mount/ -exec /usr/bin/md5sum {} \; >
investigate_files.md5
# hfind -i md5sum investigate_files.md5
# awk '{print $1}' investigate_files.md5 | hfind known_files.md5 | grep
"Hash Not Found" | awk '{print $1}' > changed.md5
# hfind -f changed.md5 investigate_files.md5 > changed_files.txt
```

7.1.3. Recover the unallocated space and sort files

The `tsk_recover` utility can be used to recover files from unallocated space by passing in the device of the attached volume and the location to store the recovered files as parameters to the command (Carrier, tsk_recover man page, n.d.b). The syntax for this is simply:

```
tsk_recover /dev/xvdf1 /cases/recovered
```

The `tsk_recover` tool adds the sector where the file was located to the end of the file name and uses a semicolon as a delimiter. For example:

```
/mnt/data/etc/yum.repos.d/epel.repo;5a135316
```

Visually scanning through the directory listing may identify files that deserve additional investigation. Use this command:

```
# ls -lR /cases/recovered/ | less
```

The Sleuth Kit contains another tool, `sorter` (Carrier, sorter man page, n.d.c), for recovering unallocated space and will sort both allocated and unallocated unknown

Kenneth G. Hartman,
ken@kennethghartman.com

files per rules appropriate to the file system. The command to do this in the background is:

```
sorter –s -f ext4 -d /mnt/data -x known_files.md5 /dev/xvdf1 &
```

This command uses the `known_files` hash list that was indexed previously and stores the output onto a new data volume that is at least the size of the evidence volume. The AWS documentation explains how to create and attach an empty EBS volume as well as format it (Amazon Web Services, n.d.c). Note that the above command assumes that the empty EBS volume was mounted to `/mnt/data`. The command to do that is:

```
# mount /dev/xvdh /mnt/data/
```

The -s switch tells `sorter` to save a copy of the sorted files into a directory for each category (Carrier, sorter man page, n.d.c). A typical directory listing of the data volume after running this command is shown below.

```
# ls -l
total 9156
drwxr-xr-x 2 root root    4096 Dec  4 21:30 archive
-rw-r--r-- 1 root root    2058 Dec  4 21:30 archive.txt
drwxr-xr-x 2 root root    4096 Dec  4 21:30 data
-rw-r--r-- 1 root root    1386 Dec  4 21:30 data.txt
drwxr-xr-x 2 root root    4096 Dec  4 21:30 documents
-rw-r--r-- 1 root root    1057 Dec  4 21:30 documents.txt
-rw-r--r-- 1 root root 6768155 Dec  4 21:30 exclude.txt
drwxr-xr-x 2 root root    4096 Dec  4 21:29 exec
-rw-r--r-- 1 root root   48378 Dec  4 21:29 exec.txt
drwxr-xr-x 2 root root    4096 Dec  4 21:30 images
-rw-r--r-- 1 root root   28857 Dec  4 21:30 images.txt
-rw-r--r-- 1 root root 2124974 Dec  4 21:30 mismatch_exclude.txt
-rw-r--r-- 1 root root   82004 Dec  4 21:30 mismatch.txt
-rw-r--r-- 1 root root     478 Dec  4 21:30 sorter.sum
drwxr-xr-x 2 root root   36864 Dec  4 21:30 text
-rw-r--r-- 1 root root  213600 Dec  4 21:30 text.txt
-rw-r--r-- 1 root root    7148 Dec  4 21:30 unknown.txt
```

The executable files that are new or modified relative to the AMI are in the `\mnt\data\exec folder` and are listed in `exec.txt`. The output from `sorter` can be further refined using a hash list from the National Software Reference Library (Carrier, 2003b). The National Software Reference Library (n.d.) contains hashes of all files from the software applications in their expansive “Reference Data Set.”

Kenneth G. Hartman,
ken@kennethghartman.com

7.1.4. Carving files from unallocated space

`Foremost` is another tool for recovering files from unallocated space. While `sorter` uses the output of the file command (Carrier, File system forensic analysis, 2005), `Foremost` uses signatures of files that consist of distinct byte patterns within the header and footer of each file type it can extract. This technique is called *data carving* (Foremost, n.d.). The results depend on the presence of accurate signatures for the file system to be carved. The current `foremost.conf` file that is distributed with the SIFT workstation is focused on media files and Microsoft Windows (Kristensen, 2014). The command to run `foremost` on the evidence prepared, as discussed above, is:

```
# foremost -dv -o /cases/forermost -c /etc/foremost.conf /dev/xvdf1
```

7.1.5. Determine if keys are present on compromised system

While sorting and classifying the files on the compromised volume, analysts should look in the SSH and AWS hidden directories for the presence of private keys. Unprotected private keys on an EC2 instance are a poor security practice and should be in violation of the organization's security policy. If SSH keys are needed on an EC2 instance, they should be protected with a long passphrase.

Amazon best practices recommend assigning an Identity and Access Management (IAM) role to EC2 instances to avoid storing AWS keys on them (Amazon Web Services, n.d.e). Discussion of the Amazon IAM service is outside of the scope of this paper. If any AWS keys or private SSH keys are identified, they should be assumed to be compromised. Run the following commands to identify private keys:

```
# ls /mnt/linux_mount/home/ec2-user/.ssh/
# ls /mnt/linux_mount/home/ec2-user/.aws/
# egrep -r AKIA[A-Z0-9]{16}  /mnt/linux_mount/
# egrep -r "PRIVATE KEY-----"  /mnt/linux_mount/
```

It is also recommended to list the public keys that are used to SSH to the instance. In conjunction with the syslog, this information helps to determine who is accessing system. Use:

```
# ls /mnt/linux_mount/home/*/.ssh/authorized_keys
# cat /mnt/linux_mount/home/ec2-user/.ssh/authorized_keys
```

7.2. Perform Anti-Virus Checks

The second step in the SANS "Finding Malware – Step by Step" process for Microsoft Windows advises to scan for malware. However, it should first be determined if any security software is installed. Sometimes anti-virus software may be installed but the alerts are ignored. In other cases, the discovery of agents may indicate additional courses of investigation to be explored.

7.2.1. Determine if any security software is installed

AWS System Manager – The AWS System Manager provides the ability to patch, configure, and audit EC2 instances to a baseline. It also allows administrators to inventory the software on EC2 instances (Amazon Web Services, n.d.f). If the AWS Systems Manager is found, there will be additional information available via the AWS EC2 Console that may support the forensic investigation of the EC2 instance. To check if the AWS Systems Manager is present, look in the default location for the executable:

```
# ls /mnt/linux_mount/usr/bin/amazon-ssm-agent
```

or its log:

```
# ls /mnt/linux_mount/var/log/amazon/ssm/amazon-ssm-agent.log
```

AWS Inspector – AWS Inspector is Amazon's vulnerability scanner (Amazon Web Services, n.d.g). Finding this tool on an EC2 instance indicates that there may be vulnerability scans available via the Inspector console. To learn more about Inspector, see the user guide (Amazon Web Services, n.d.g). Any information regarding known vulnerabilities on the EC2 instance can help focus the investigation. Look for the executable in the default location:

```
# ls /mnt/linux_mount/opt/aws/awsagent/bin/awsagent
```

Splunk Forwarder – The Splunk Universal Forwarder is used by many organizations to collect and archive system logs (Splunk, n.d.). By default, the executable is found at

Kenneth G. Hartman,
ken@kennethghartman.com

`/mnt/linux_mount/opt/splunkforwarder/bin/splunkd` and the logs it collects can be located using the command:

```
# cat /mnt/linux_mount/opt/splunkforwarder/etc/apps/*/local/inputs.conf
```

7.2.2. Scan with ClamAV

ClamAV is an open-source anti-malware scanner (ClamAV, n.d.) that comes preinstalled on the SIFT Workstation. Ensure that SIFT workstation is updated to keep the signatures current. To scan the mounted evidence volume with ClamAV, use:

```
# clamscan -i -r --log=/cases/clam.log /mnt/linux_mount/
```

Likewise, to scan the files recovered from the unallocated space, use:

```
# clamscan -i -r --log=/cases/clam.log /cases/recovered
```

7.3. Search for Known Indicators of Compromise

Loki is an IOC scanner written in python (Roth, n.d.) that runs without issue on the SIFT Workstation, but does have to be installed separately. To install it and have it scan the attached evidence, use the following commands:

```
cd /tmp
wget https://github.com/Neo23x0/Loki/archive/v0.24.3.tar.gz
tar -xzvf v0.24.3.tar.gz
cd Loki-0.24.3/
pip install pylzma
pip install netaddr
python loki.py -p /mnt/linux_mount/
```

The following snippet is output from Loki when it discovers indicators of compromise:

```
[ALERT]
FILE: /mnt/linux_mount/bin/.libgcc SCORE: 100 TYPE: ELF SIZE: 2979640
FIRST_BYTES: 7f454c460201010300000000000000000002003e00 / ELF>
MD5: 4fa4269b7ce44bfce5ef574e6a37c38f
SHA1: e88c46b6f775cf5079857e45ac7bf3faeb9d8e11
SHA256: 63210b24f42c05b2c5f8fd62e98dba6de45c7d751a2e55700d22983772886017
CREATED: Fri Sep  1 06:28:41 2017 MODIFIED: Wed Aug 30 09:49:31 2017 ACCESSED:
Fri Sep  1 06:27:57 2017
REASON_1: Malware Hash TYPE: MD5 HASH: 4fa4269b7ce44bfce5ef574e6a37c38f
SUBSCORE: 100 DESC: Linux.Lady http://vms.drweb.com/virus/?_is=1&amp
[WARNING]
FILE: /mnt/linux_mount/usr/lib/python2.7/dist-
packages/pip/_vendor/distlib/w32.exe SCORE: 60 TYPE: EXE SIZE: 88576
FIRST_BYTES: 4d5a90000300000004000000ffff0000b8000000 / MZ
MD5: 8a629fa901ed50f3dd7bba92b4535f03
```

Kenneth G. Hartman,
ken@kennethghartman.com

```
SHA1: cb73af97341c347a20d1246cce906250c17eadf7
SHA256: 78445153afe05dc55a435336b87011dc14d08ec3c8d126e2986c8c2c60c7ce9b
CREATED: Mon Jun 19 16:04:21 2017 MODIFIED: Wed Aug 10 21:51:27 2016 ACCESSED:
Wed Aug 10 21:51:27 2016
REASON_1: File Name IOC matched PATTERN:
/(q32|q64|wceaux|w86|q86|quarkpwd[^/]*|m64|m32|hash32|hash64|64|32|w32|w64|wce3
2|wce64|w32|w64|wce|p32|p64|ps32|ps64|mimikatz|mimilove|mm32|mm64|pw32|pw64|g32
|g64|gs32|gs64|hash|hashdump|dumpsvc)\.exe SUBSCORE: 60 DESC: Cred Dumping
```

7.4. Identify Evidence of Persistence

To survive a reboot, malware must use a persistence mechanism. The two most common methods are cron jobs and start-up scripts.

7.4.1. Look for unusual cron jobs

The SANS Linux Intrusion Discovery Cheat Sheet (SANS Institute, n.d.b) provides the following two suggestions for looking at system-wide cron jobs:

```
cat /mnt/linux_mount/etc/crontab
ls /mnt/linux_mount/etc/cron.*
```

It is also important to check the cron jobs for all users. To list the users that have cron jobs scheduled on the attached Amazon Linux volume, run:

```
ls –l /mnt/linux_mount/var/spool/cron/*
```

To quickly peruse the cron jobs when several user accounts have jobs scheduled, use:

```
find /mnt/linux_mount/var/spool/cron/* -exec cat {} \;
```

7.4.2. Look for unusual start-up scripts

Some malware will make use of the start-up scripts that Linux runs at boot time when entering a specific run level. On some Linux distributions, these are found in `/etc/init.d`, but on Amazon Linux and Red Hat variants, the scripts will be in `/etc/rc*.d`. Use the following command to list the scripts, in reverse chronological order:

```
ls -als -t /mnt/linux_mount/etc/rc*.d/
```

Cron jobs and the start-up scripts may have innocuous names and may call other scripts, so sometimes an investigator may have to dig into the details to determine its

Kenneth G. Hartman,
ken@kennethghartman.com

nature. The "Bill Gates Botnet" (Akamai, 2016) for example, installs the following scripts:

```
/etc/init.d/DbSecuritySpt
/etc/rc1.d/S97DbSecuritySpt
/etc/rc2.d/S97DbSecuritySpt
/etc/rc3.d/S97DbSecuritySpt
/etc/rc4.d/S97DbSecuritySpt
/etc/rc5.d/S97DbSecuritySpt
```

7.5. Check for Suspicious Files

7.5.1. Look in the tmp directory

The `/tmp` directory is frequently used to store uploaded files. Use the following commands to examine the `/tmp` directory of the mounted volume as well as the recovered files:

```
ls -als /mnt/linux_mount/tmp
ls -als /cases/recovered/tmp
```

A look in the `/tmp` directory on one compromised instance produced the following three suspicious files:

- `gates.lod`
- `moni.lod`
- `tmplog`

The first two files are IOC's for the Bill Gates Botnet (Akamai, 2016) as determined by an internet search. The contents of the tmplog file showed the configuration for a Bitcoin miner:

```
CMD: /bin/wipefs -B -o stratum+tcp://mine.ppxxmr.com:7777 -u {REDACTED}
-p x -k --max-cpu-usage=100
```

7.5.2. Look for unusual files

The SANS Linux Intrusion Discovery Cheat Sheet (SANS Institute, n.d.b) advises to look for unusual SUID files and assumes the user has knowledge of which SUID files are normal. See *What is SUID and how to set SUID in Linux/Unix* (Anne, 2011) for an

Kenneth G. Hartman,
ken@kennethghartman.com

explanation of SUID files. The following commands perform the comparison to the mounted baseline volume:

```
# find /mnt/linux_mount/ -uid 0 -perm -4000 -print > suid_evidence
# find /mnt/linux_base/ -uid 0 -perm -4000 -print > suid_base
# cut suid_base -d"/" -f4-  > suid_base_relative
# cut suid_base -d"/" -f4-  > suid_evidence_relative
# diff suid_base_relative suid_evidence_relative
```

The same technique can be used to identify large files that are greater than a certain size and compare them to the base volume. The following commands look for files greater than ten megabytes:

```
find /mnt/linux_mount/ -size +10000k
find /cases/recovered/ -size +10000k
```

7.5.3. Look for files with high entropy

DensityScout is a tool on the SIFT workstation that detects packing, compression, and encrypted files that exceed a “density” threshold (Wojner, 2012). Because compressed files are common on Linux, comparison to the baseline volume reduces the output significantly.

```
# densityscout -r -p 0.1 -l 0.1 -o high_density_evidence.txt /mnt/linux_mount/
# densityscout -r -p 0.1 -l 0.1 -o high_density_base.txt /mnt/linux_base/
# cut high_density_evidence.txt -d"/" -f4- > high_density_evidence_relative.txt
# cut high_density_base.txt -d"/" -f4-  > high_density_base_relative.txt
# diff high_density_base_relative.txt high_density_evidence_relative.txt
```

The files recovered from unallocated space should be examined as well.

```
# densityscout -r -p 0.1 -l 0.1 -o high_density_recovered.txt /cases/recovered/
```

7.5.4. Examine suspicious files for IOCs

When suspicious files are identified, the `strings` command can provide a quick indication regarding the nature of the file and can identify potential indicators of compromise. The output of the strings command may include file names, IP addresses, configuration details, menu options, and help screens. IP addresses should be investigated by a network security analyst. Example commands include:

```
strings  mymalwarefile                                        # Extract Strings
grep -aoE "\b([0-9]{1,3}\.){3}[0-9]{1,3}\b"  mymalwarefile  # Extract IPs
```

Kenneth G. Hartman,
ken@kennethghartman.com

7.6. Review system logs and configuration

Ideally, the critical logs were offloaded to a central repository such as Splunk or a S3 bucket. Regardless, an examination of the logs on the EBS volume is an important activity because system logs are a valuable source of information regarding the state of the system before and after the attack and may provide answers to how and when the system was compromised.

7.6.1. Review the bash history

Although the bash history is not a robust audit log, it still provides information that may be of interest to the examiner. For example, it can reveal skill level and stylistic proclivities unique to a certain hacker in addition its intended purpose of capturing the recent commands entered by a user. A user can modify the bash history associated with his or her account, but rarely will do so unless trying to cover the tracks. Because there are multiple ways to avoid bash history logging (Skoudis, 2012) it should not be considered a security control. Indications that bash history has been altered or evaded is a noteworthy TTP. Examine the default ec2-user account, root, and any other accounts on the system.

```
cat /mnt/linux_mount/home/ec2-user/.bash_history
cat /mnt/linux_mount/root/.bash_history
```

7.6.2. Examine local user accounts and groups

The SANS Linux Intrusion Discovery Cheat Sheet (SANS Institute, n.d.b) suggests that an investigator look for unusual accounts and multiple accounts with a user id (UID) set to zero. Also, note any new groups or services that have created an account as well.

```
# diff /mnt/linux_base/etc/passwd /mnt/linux_mount/etc/passwd
# diff /mnt/linux_base/etc/group /mnt/linux_mount/etc/group
```

Look for accounts with passwords set, assuming this is a policy violation:

```
cat /mnt/linux_mount/etc/shadow | grep -F "$"
```

Kenneth G. Hartman,
ken@kennethghartman.com

7.6.3. Identify boot history

Understanding when a system has been booted is important information for constructing a timeline of events in the life of the EC2 instance. Use the following commands to glean this information:

```
# ls -als  /mnt/linux_mount/var/log/dmesg*
# grep Cloud-init  /mnt/linux_mount/var/log/cloud-init.log
# cat  /mnt/linux_mount/var/log/boot.log
```

7.6.4. Identify past IP addresses

Snapshots and AMIs may be shared across different AWS accounts and knowledge of this can help enrich the timeline. In addition, there may be other circumstances where the IP address will change. This can be identified in the `cloud-init-output.log` as follows:

```
# grep -A4 -B1 "Net device info" /mnt/linux_mount/var/log/cloud-init-
output.log
```

7.6.5. Look at the yum log

The yum.log contains a list of the packages that have been installed on the EC2 instance. Understanding this information may provide insight into how the system was intended to function or if any vulnerable software was added. It may be helpful to compare with the baseline to determine which packages have been added or removed since the system was launched from the AMI.

```
# cat /mnt/linux_mount/var/log/yum.log
# diff /mnt/linux_base/var/log/yum.log /mnt/linux_mount/var/log/yum.log
```

7.6.6. Look in /var/log/ for relevant logs

The default location for all logs is `/var/log` and all logs should be perused to identify information that may be relevant to the specific investigation. The `var/log/secure` file contains details about user account changes, ssh connections, sudo activity, and su sessions. By default, Amazon Linux has the `auditd` service running and the considerable amount of valuable information that it has logged will be found in `/var/log/audit/*`

Kenneth G. Hartman,
ken@kennethghartman.com

```
ls -als /mnt/linux_mount/var/log/
cat /mnt/linux_mount2/var/log/secure* | less
cat /mnt/linux_mount2/var/log/audit/* | less
```

The audit logs will take some practice to read and since the timestamp is in epoch time, the manual review is only to look for anomalies. The SuperTimeline discussed later in this paper will parse both the `secure` and `audit` logs and convert the timestamp to be human readable.

7.6.7. Look at webserver logs

The EC2 instance may be running a vulnerable web server or web service. The typical configuration will be to have some logging enabled by default, but the logs may be in unusual places on the volume. Any web server logs found on the system are bound to be a treasure trove of information with a variety of attempted attacks. The following command will find most web logs:

```
# egrep -lr "GET / HTTP/1.1" /mnt/linux_mount | less
```

The best way to analyze these logs is to pull them into Splunk. Splunk has a free license that permits ingestion of 500 megabytes of data per day. Incident response teams should be prepared to launch a dedicated Splunk AMI as needed for ad hoc analysis. If Elastic Load Balancers (ELB) are used to proxy incoming HTTP requests, the source IP address in the web logs may be the internal address of the ELB and not the attacker. During IR preparations, this setting should be changed and a process to archive the web server logs should be established.

7.7. Perform a timeline analysis

A timeline is an indispensable tool that can unify the investigation effort. The process of creating it can help identify gaps and missing information. A summarized timeline can help to effectively communicate the sequence of events to management (Liston, 2012).

Kenneth G. Hartman,
ken@kennethghartman.com

7.7.1. Make a Filesystem Timeline

A filesystem timeline is quick to create, as it just sorts the file system metadata into chronological order. By default, the timeline will include both allocated as well as unallocated files (Lee, 2009b). The output of the two-step process is a CSV file that can be analyzed with `grep` or converted into a spreadsheet.

First, make the body file (Timelines, 2008):

```
# fls -r -m /  /dev/xvdf1 > /cases/body.txt
```

Then, make a CSV file using `mactime` (Mactime, 2010):

```
# mactime -b /cases/body.txt -d  > /cases/timeline.csv
```

The `fls` command gathers the metadata for each file into a body file which contains one row per file. This includes the modify, access, create, and change/birth (MACB) timestamps. The `mactime` command rearranges this metadata in temporal order, creating multiple records when the MACB timestamps on a given file are different.

7.7.2. Make a SuperTimeline

A SuperTimeline will take much longer to create but includes many of the system logs, in addition to file system activity (Lee, 2009b). First, make a plaso dump file in the background with `nohup`, so that the terminal can be disconnected if necessary:

```
# nohup log2timeline.py /cases/plaso.dump  /dev/xvdf1 &
```

Plaso is an engine for storing timestamp data and the dump file is essentially the database. The format of the dump file is designed to be efficient from a performance and storage perspective (Metz & Gudjonsson, n.d.). Hence, other tools are necessary to access the database after it is created with the `log2timeline.py` command.

The log2timeline.py tool supports a command line switch to adjust the timezone, however EC2 instances are set to UTC by default. Hence, all logs are generated without time correction and, therefore, this switch is not needed when creating a SuperTimeline. When the processing is complete, pinfo.py can be used to examine the plaso file (Metz & Gudjonsson, n.d.):

```
# pinfo.py -v /cases/plaso.dump | less
```

The psort.py tool is used to create a CSV from the plaso file, however an unfiltered CSV file can be unwieldly and is often too large to analyze with Microsoft Excel. Therefore, the standard process is to create a CSV between two dates that encompass the events of interest to the investigation (Nides, 2011).

```
# psort.py /cases/plaso.dump "date > 'YYYY-MM-DD HH:MM:SS' AND date <
'YYYY-MM-DD HH:MM:SS'" > /cases/supertimeline.csv
```

7.7.3. Fix timestamp in syslog records

There is a potential issue with creating timelines on EC2 instances launched from the typical Amazon Linux images in the marketplace. These systems use a default setting for syslog that does not include the year in the date stamp. To address this, plaso assumes that the year on the syslog files is the maximum year in the plaso database. Thus, if any other logs have invalid time stamps, all syslog records could show up a year or more into the future and could be missed in a CSV file that was created with date filters as shown in the previous section. As of this writing, the plaso developers have an open ticket to determine a better way to handle this situation (Metz, Is the behavior of maximum year limit in syslog parser what is desired?, 2017).

Ideally, a hardened image would be used which properly configures the syslog timestamp to include the year, but this may not be the case, so the investigator should be prepared to address the default settings. The options are to alter the time stamps to include the year before the various syslog files are processed by plaso or to alter the timestamps in the resultant CSV file. Since the CSV is derived and the syslog files are intact evidence, the simplest and best choice is to modify any future dates in the CSV and re-sort it. The steps are shown below, assuming the EC2 instance has only existed in the year 2017:

```
# psort.py /cases/plaso.dump > /cases/supertimeline-all.csv
# sed -r 's|^[[:digit:]]{4}(.*,syslog,.*)$|2017\1|g'
/cases/supertimeline-all.csv > /cases/supertimeline-all-fixed.csv
```

Next, sort the output and extract the dates of interest, keeping the header row:

```
# head -n 1 supertimeline-all.csv > supertimeline-all-extract.csv
# sort supertimeline-all-fixed.csv > supertimeline-all-sorted.csv
$ grep "YYYY-MM-DD" supertimeline-all-sorted.csv >> supertimeline-all-
extract.csv
```

Kenneth G. Hartman,
ken@kennethghartman.com

The file can now be loaded into a spreadsheet for analysis and color-coding. For an example of a colorized timeline for Windows, see the *Digital forensic SIFTing: Colorized super timeline template for log2timeline output files* blog post (Lee, 2012).

7.8. Perform Memory Analysis

To perform memory analysis with Rekall, one must have a profile that matches the operating system kernel (Michael, 2014). Once the memory dump is obtained and copied to the SIFT Workstation as discussed in Section 6.5 it can be analyzed with Rekall using the following command, substituting LINUX-PROFILE with the path to the profile:

```
$ rekall --profile LINUX-PROFILE -f /cases/memcapture.lime
```

This puts Rekall in an interactive mode allowing plugins to be run one after the other without having to repeat the information that was just entered onto the command line. The steps below show this process:

7.8.1. Identify rogue processes

To capture the list of processes in memory, use `pslist` and to display the processes in a hierarchical tree us `pstree`. To save the data to a file use the `output` parameter as shown.

```
[1] memcapture.lime 02:45:23> pslist output="/cases/pslist.txt"
----------------------------> pslist(output="/cases/pslist.txt")
Out<02:45:24> Plugin: pslist (LinuxPsList)
[1] memcapture.lime 02:46:38> pstree output="/cases/pstree.txt"
----------------------------> pstree(output="/cases/pstree.txt")
Out<02:46:39> Plugin: pstree (LinPSTree)
```

7.8.2. Dump suspicious processes

Dump the kernel drivers to a directory:

```
[1] memcapture.lime 03:37:30> moddump dump_dir="/cases/moddump"
----------------------------> moddump(dump_dir="/cases/moddump")
```

Kenneth G. Hartman,
ken@kennethghartman.com

Dump all memory into files that can be analyzed with tools like `grep` and `strings` or a scanned with a virus scanner:

```
[1] memcapture.lime 03:40:55> memdump dump_dir="/cases/memdump"
----------------------------> memdump(dump_dir="/cases/memdump")
```

7.8.1. Review network artifacts

The netstat plugin lists the connections and the related process name.

```
[1] memcapture.lime 03:22:47> netstat output="/cases/netstat.txt"
----------------------------> netstat(output="/cases/netstat.txt")
Out<03:22:48> Plugin: netstat (Netstat)
```

The steps described above are only a small portion of what can be done with Rekall and memory analysis. Analysis of the processes and network artifacts collected in the previous steps may provide additional IOCs or TTPs.

7.9. Perform Third-party hash lookups

At this point in the investigation, the investigator may have identified one or more files that are suspected to be malware and may wonder if the samples have been seen in the wild. VirusTotal is a free online service that detects malicious content in files and URLs. Suspicious files can be uploaded and will be scanned with multiple anti-virus scanning engines. In addition, VirusTotal can be searched by hash to see if it is in their database already. VirusTotal also warns that any files that are uploaded are made available to its premium users (VirusTotal, n.d.). Therefore, the service should be used with discretion, especially if uploading custom malware will tip off an attacker or if the file contains sensitive data or source code.

The SIFT Workstation contains two tools to facilitate VirusTotal usage from the command line. These tools are both written by Didier Stevens and require a VirusTotal API key (Stevens, n.d.), which can be obtained by creating a free account on VirusTotal.

The first tool, `virustotal-search.py`, takes a list of hashes and searches the VirusTotal database. The following snippet illustrates its usage:

```
$ export VIRUSTOTAL_API2_KEY={YOUR_API_KEY HERE}
$ cat hashlist4virustotal.md5
4fa4269b7ce44bfce5ef574e6a37c38f
```

Kenneth G. Hartman,
ken@kennethghartman.com

```
8a629fa901ed50f3dd7bba92b4535f03
3bd016b34cd5bd0a562b8f56f0cb971b
9b587ceee0e1236fff1333bd8752c64e
$ virustotal-search.py hashlist4virustotal.md5 > virustotal-results.txt
```

The results of the query are saved as `virustotal-results.txt` and are displayed in Appendix B, due to the length. The first two MD5 hashed queries were from the Loki scan listed in Section 7.3 and the last two hashes were selected at random from files in evidence.

The second tool is `virustotal-submit.py` and it requires that a python module called `poster` be installed before it can be used on the current version of the SIFT workstation. The command to install poster and submit a file to VirusTotal is shown below:

```
$ sudo pip install poster
$ virustotal-submit.py /mnt/linux_mount/lib64/libnfsidmap.so
/mnt/linux_mount/lib64/libnfsidmap.so;1;Scan request successfully
queued, come back later for the
report;3bd016b34cd5bd0a562b8f56f0cb971b;42732907e5cf2ccbba25944ee5bee44
8545465e33363fb2b27200611586c4d31;42732907e5cf2ccbba25944ee5bee44854546
5e33363fb2b27200611586c4d31-
1512590781;https://www.virustotal.com/file/42732907e5cf2ccbba25944ee5be
e448545465e33363fb2b27200611586c4d31/analysis/1512590781/
```

When the `virustotal-submit.py` command submits the file, it gets queued for processing and a URL is returned. The investigator can visit the URL with a browser, or VirusTotal can be queried using `virustotal-search.py` and providing the hash as before.

7.10. File Timestamp Anomalies

Attackers have been known to use timestomping to mimic the times of other files in the folder in which the malware is hiding. Malware does this by altering the MACB timestamps (MITRE, 2017). On Windows systems, timestomping can be detected by identifying discrepancies between the $FILE_NAME time stamps and the $STANDARD_INFO time stamps that are stored in the Master File Table (Hull, 2010), however that data is not available in Linux file systems.

Timestomping is accomplished using two techniques. One method is to copy the metadata from one file to another. The other method uses a program to set the metadata

Kenneth G. Hartman,
ken@kennethghartman.com

directly (Offensive Security, n.d.). When the MACB times are set directly, the nanoseconds are left as zeros. Therefore, seeing zeros in the nanoseconds position may indicate timestomping. This is not foolproof as some programs or operating system functions may zero the nanoseconds as well. The `full-time` switch can be used with `ls` to see the nanoseconds on a file listing and `stat` can be used to view the full details:

```
sudo ls -als --full-time /mnt/linux_mount/home/ec2-user/
total 32
4 drwx------  4  500  500 4096 2017-12-03 22:38:13.936111000 +0000 .
4 drwxr-xr-x  3 root root 4096 2017-12-03 22:26:42.342253249 +0000 ..
4 -rw-------  1  500  500   13 2017-12-03 22:38:13.936111000 +0000 .bash_history
4 -rw-r--r--  1  500  500   18 2017-08-30 19:00:41.000000000 +0000 .bash_logout
4 -rw-r--r--  1  500  500  193 2017-08-30 19:00:41.000000000 +0000 .bash_profile
4 -rw-r--r--  1  500  500  124 2017-08-30 19:00:41.000000000 +0000 .bashrc
4 drwxr-xr-x 10 root root 4096 2017-12-03 22:34:23.364111000 +0000 joplin
4 drwx------  2  500  500 4096 2017-12-03 22:26:42.702253249 +0000 .ssh
```

Notice that the operating system left the nanoseconds on the timestamps as zeros on the .bash* files and then when the bash_history was updated due to normal usage the nanoseconds changes to the current time.

To focus the search for timestamp anomalies on the files that have changed, perform a visual scan using the file listing that was created in Section 7.1.2, as shown below:

```
# cat changed_files.txt | cut -d" " -f3 | xargs ls -als --full-time |
grep ".000000000" | less
# cat changed_files.txt | cut -d" " -f3 | xargs stat | less
```

7.11. Reverse Engineer Malware

At this stage, the SANS "Finding Malware – Step by Step" process recommends passing the malware samples to an analyst skilled with reverse engineering malware, if the situation warrants it. Typically, this decision will be based on the sensitivity of the compromise and the nature of the attack.

8. Conclusion

The tendency of many cloud users is to immediately terminate a virtual machine that is compromised, destroying the evidence in the process. Instead, the tools from ThreatResponse make it easy to preserve the evidence. However, training, practice, and documented incident response procedures are still necessary to ensure the incident

Kenneth G. Hartman,
ken@kennethghartman.com

handlers have the access necessary to respond. Once the evidence is collected, the SIFT Workstation can be used to analyze the evidence, find indicators of compromise to determine the scope of the incident, determine timelines, and perform a root cause analysis. While there is certainly much more that can be addressed on the topic of Amazon Linux EC2 Forensics, this paper provides step by step guidance for those on the front lines.

Kenneth G. Hartman,
ken@kennethghartman.com

References

Akamai. (2016, April 4). *Threat advisory: "BillGates" botnet.* Retrieved from Akamai: https://www.akamai.com/kr/ko/multimedia/documents/state-of-the-internet/bill-gates-botnet-threat-advisory.pdf

Amazon Web Services. (n.d.a). *AWS CloudTrail*. Retrieved from AWS: https://aws.amazon.com/cloudtrail/

Amazon Web Services. (n.d.b). *Amazon elastic block store*. Retrieved from AWS: https://aws.amazon.com/ebs/

Amazon Web Services. (n.d.c). *Amazon EBS volumes*. Retrieved from AWS: http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSVolumes.html

Amazon Web Services. (n.d.d). *Launching an instance using the Launch Instance Wizard*. Retrieved from AWS: http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/launching-instance.html

Amazon Web Services. (n.d.d). *Using an IAM role to grant permissions to applications running on Amazon EC2 instances*. Retrieved from AWS: http://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_use_switch-role-ec2.html

Amazon Web Services. (n.d.e). *AWS System Manager*. Retrieved from AWS: https://aws.amazon.com/systems-manager/

Amazon Web Services. (n.d.f). *What is Amazon Inspector?* Retrieved from AWS: https://docs.aws.amazon.com/inspector/latest/userguide/inspector_introduction.html

Anne, S. (2011, December 29). *What is SUID and how to set SUID in Linux/Unix?* Retrieved from The Linux Juggernaut: https://www.linuxnix.com/suid-set-suid-linuxunix/

Kenneth G. Hartman,
ken@kennethghartman.com

Arnold, T. (2016, April 19). *Incident response in Amazon EC2: First responders guide to security incidents in the cloud.* Retrieved from SANS Reading Room: https://www.sans.org/reading-room/whitepapers/cloud/incident-response-amazon-ec2-first-responders-guide-security-incidents-cloud-36902

AWS_IR quickstart. (n.d.). Retrieved from Read The Docs: http://aws-ir.readthedocs.io/en/latest/quickstart.html

Carrier, B. (2003a, July 15). *The sleuth kit informer, issue #6.* Retrieved from Slueth Kit: http://www.sleuthkit.org/informer/sleuthkit-informer-6.html

Carrier, B. (2003b, April 15). *The sluethkit informer, Issue #3*. Retrieved from Slueth Kit: http://www.sleuthkit.org/informer/sleuthkit-informer-3.html

Carrier, B. (2005). *File system forensic analysis.* Upper Saddle River, NJ: Addison-Wesley Professional.

Carrier, B. (n.d.a). *hfind man page*. Retrieved from The Slueth Kit: http://www.sleuthkit.org/sleuthkit/man/hfind.html

Carrier, B. (n.d.b). *tsk_recover man page.* Retrieved from The Slueth Kit: http://www.sleuthkit.org/sleuthkit/man/tsk_recover.html

Carrier, B. (n.d.c). *sorter man page.* Retrieved from The Slueth Kit: http://www.sleuthkit.org/sleuthkit/man/sorter.html

ClamAV. (n.d.). *ClamAV*. Retrieved from https://www.clamav.net/

Clarke, T. (2010, July 9). *Is there best practice for a server to system administrator ratio?* Retrieved from Computerworld Australia: https://www.computerworld.com.au/article/352635/there_best_practice_server_system_administrator_ratio_/

Kenneth G. Hartman,
ken@kennethghartman.com

Coles, C. (n.d.). *AWS vs Azure vs Google cloud market share 2017*. Retrieved from Skyhigh: https://www.skyhighnetworks.com/cloud-security-blog/microsoft-azure-closes-iaas-adoption-gap-with-amazon-aws/

Dykstra, J. A. (2013). *Digital forensics for infrastructure-as-a-service cloud computing.* Catonsville, MD, USA: University of Maryland at Baltimore County.

Ephemeral Systems. (n.d.). *ThreatResponse examiner workstation by Ephemeral Systems*. Retrieved from Github: https://github.com/EphemeralSystems/threatresponse-ws

Foremost. (n.d.). Retrieved from Sourceforge.net: http://foremost.sourceforge.net/

Goutam, R. K. (2015, December). The Problem of Attribution in Cyber Security. *International Journal of Computer Applications, 131*(7). Retrieved from http://www.ijcaonline.org/research/volume131/number7/goutam-2015-ijca-907386.pdf

Hamilton, J. (2007). On designing and deploying Internet-scale services. *Proceedings of the 21st Large Installation System Administration Conference* (pp. 231-242). Dallas, TX: USENIX Association. Retrieved from http://static.usenix.org/event/lisa07/tech/full_papers/hamilton/hamilton_html/

Hull, D. (2010, November 2). *Digital forensics: Detecting time stamp manipulation*. Retrieved from SANS DFIR: https://digital-forensics.sans.org/blog/2010/11/02/digital-forensics-time-stamp-manipulation

Kauflin, J. (2917, March 16). *The fast-growing job with a huge skills gap: Cyber security*. Retrieved from Forbes: https://www.forbes.com/sites/jeffkauflin/2017/03/16/the-fast-growing-job-with-a-huge-skills-gap-cyber-security/#7faece2b5163

Kenneth G. Hartman,
ken@kennethghartman.com

Kral, P. (2011, December 12). *The incident handler's handbook.* Retrieved from SANS Reading Room: https://www.sans.org/reading-room/whitepapers/incident/incident-handlers-handbook-33901

Kristensen, E. (2014, March 12). *Foremost configuration file*. Retrieved from Github: https://github.com/sans-dfir/sift-files/blob/master/foremost/foremost.conf

Krug, A. (2016). *Hardening AWS environments and automating incident response for AWS compromises.* Retrieved from Black Hat: https://www.blackhat.com/docs/us-16/materials/us-16-Krug-Hardening-AWS-Environments-And-Automating-Incident-Response-For-AWS-Compromises.pdf

Krug, A., McCormack, A., Ferrier, J., & Parr, J. (n.d.). *Open source incident response toolkit*. Retrieved from ThreatResponse: https://threatresponse.cloud/

Lee, R. (2009a, February 19). *Digital forensic SIFTing: How to perform a read-only mount of filesystem evidence*. Retrieved from SANS Digital Forensics and Incident Response Blog: https://digital-forensics.sans.org/blog/2009/02/19/digital-forensic-sifting-how-to-perform-a-read-only-mount-of-evidence/

Lee, R. (2009b, February 24). *Digital forensic SIFT'ing: Registry and filesystem timeline creation*. Retrieved from SANS DFIR: https://digital-forensics.sans.org/blog/2009/02/24/digital-forensic-sifting-registry-and-filesystem-timeline-creation/

Lee, R. (2012, January 25). *Digital forensic SIFTing: Colorized super timeline template for log2timeline output files*. Retrieved from SANS DFIR: https://digital-forensics.sans.org/blog/2012/01/25/digital-forensic-sifting-colorized-super-timeline-template-for-log2timeline-output-files

Linux profiles. (n.d.). Retrieved from Github: https://github.com/volatilityfoundation/volatility/wiki/Linux

Kenneth G. Hartman,
ken@kennethghartman.com

Linux support in Rekall. (n.d.). Retrieved from Github: https://github.com/google/rekall/tree/master/tools/linux

Liston, K. (2012, June 22). *Investigator's tool-kit: Timeline*. Retrieved from Internet Storm Center: https://isc.sans.edu/forums/diary/Investigators+Toolkit+Timeline/13537/

Mactime. (2010, August 13). Retrieved from Sluethkit Wiki: https://wiki.sleuthkit.org/index.php?title=Mactime

McGeehan, R. (2016, November 28). *Investigating CloudTrail logs*. Retrieved from Starting Up Security: https://medium.com/starting-up-security/investigating-cloudtrail-logs-c2ecdf578911

Metz, J. (2017, January 3). *Is the behavior of maximum year limit in syslog parser what is desired?* Retrieved from Github: https://github.com/log2timeline/plaso/issues/1171

Metz, J., & Gudjonsson, K. (n.d.). *log2timeline/plaso wiki*. Retrieved from Github.

Michael. (2014, February 14). *Rekall profiles*. Retrieved from Rekall Forensics blog: http://blog.rekall-forensic.com/2014/02/rekall-profiles.html

MITRE. (2017, July 11). *Timestomp*. Retrieved from Mitre partnership network; Adversarial tactics, techniques and common knowledge: https://attack.mitre.org/wiki/Technique/T1099

National software reference library. (n.d.). Retrieved from NIST, Information Technology Laboratory, Software and Systems Division: https://www.nist.gov/software-quality-group/national-software-reference-library-nsrl

Nides, D. (2011, December 16). *General forensic analysis checklist V.1.1 (sic).* Retrieved from SANS DFIR: https://digital-forensics.sans.org/media/log2timeline_cheatsheet.pdf

Kenneth G. Hartman,
ken@kennethghartman.com

NIST special publication 800-61 revision 2, computer security incident handling guide. (2012, August). Retrieved from National Institute of Standards and Technology: http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-61r2.pdf

NVTEH. (2017, May 31). *An elegant way to ruin your company's day - Introduction to public AWS EBS snapshots*. Retrieved from NVTEH: https://www.nvteh.com/news/problems-with-public-ebs-snapshots

Offensive Security. (n.d.). *Timestomp*. Retrieved from Offensive Security: https://www.offensive-security.com/metasploit-unleashed/timestomp/

Olsen, P. (2014, October 12). *Forensics in the Amazon cloud - EC2*. Retrieved from System Forensics: http://sysforensics.org/2014/10/forensics-in-the-amazon-cloud-ec2/

Oltsik, J. (2017a, November). *The life and times of cybersecurity professionals.* Retrieved from ESG: http://www.esg-global.com/esg-issa-research-report-2017

Oltsik, J. (2017b, January 5). *2017: The year of cybersecurity scale*. Retrieved from CSO: https://www.csoonline.com/article/3154813/security/2017-the-year-of-cybersecurity-scale.html

Roth, F. (n.d.). *Loki - simple IOC scanner*. Retrieved from Github: https://github.com/Neo23x0/Loki

SANS DFIR. (2013). *Digital forensics and incident response poster, 22nd edition .* Retrieved from SANS DFIR: https://digital-forensics.sans.org/media/poster_fall_2013_forensics_final.pdf

SANS DFIR. (n.d.). *SIFT workstation*. Retrieved from SANS DFIR: https://digital-forensics.sans.org/community/downloads

SANS Institute. (n.d.a). *FOR508: Advanced digital forensics, incident response, and threat hunting*. Retrieved from SANS: https://www.sans.org/course/advanced-incident-response-threat-hunting-training

Kenneth G. Hartman,
ken@kennethghartman.com

SANS Institute. (n.d.b). *Intrusion discovery cheat sheet v2.0.* Retrieved from SANS: https://www.sans.org/media/score/checklists/ID-Linux.pdf

SIFT CLI. (n.d.). Retrieved from Github: https://github.com/sans-dfir/sift-cli

SIFTonEC2. (n.d.). Retrieved from Github: https://github.com/Resistor52/SIFTonEC2

Skoudis, E. (2012, June 6). *Escaping restricted Linux shells*. Retrieved from SANS Penetration Testing: https://pen-testing.sans.org/blog/2012/06/06/escaping-restricted-linux-shells

Splunk. (n.d.). *The universal forwarder*. Retrieved from Splunk: https://docs.splunk.com/Documentation/Forwarder/7.0.1/Forwarder/Abouttheuniversalforwarder

Stevens, D. (n.d.). *VirusTotal tools*. Retrieved from Didier Stevens: https://blog.didierstevens.com/programs/virustotal-tools/

Stilgherrian. (2015, June 3). *Telstra CISO blasts cyber 'attribution distraction'*. Retrieved from ZDNet: http://www.zdnet.com/article/telstra-ciso-blasts-cyber-attribution-distraction/

ThreatResponse workstation and Rekall. (2017). Retrieved from ThreatResponse: https://threatresponse.cloud/blog/2017/fun_with_rekall.html

Timelines. (2008, October 26). Retrieved from Sluethkit Wiki: https://wiki.sleuthkit.org/index.php?title=Timelines

VirusTotal. (n.d.). *About VirusTotal*. Retrieved from VirusTotal: https://www.virustotal.com/en/about/

Wojner, C. (2012, April 26). *Finding (unknown) malware with DensityScout*. Retrieved from SANS DFIR: https://digital-forensics.sans.org/blog/2012/04/26/finding-unknown-malware-with-densityscout

Kenneth G. Hartman,
ken@kennethghartman.com

9. Appendix A - Using AWS_IR

The complete install instructions are available at the ThreatResponse Github site https://github.com/ThreatResponse/aws_ir

```
sudo pip install pluginbase
sudo pip install aws_ir
```

To verify, display the help with the following command:

```
aws_ir -h
```

Sample aws_ir command and output:

```
$ aws_ir --examiner-cidr-range '54.86.103.230/32' instance-compromise --target
54.152.212.162  --user ec2-user --ssh-key ~/.ssh/your-key-name.pem

2017-11-30T00:59:33 - aws_ir.cli - INFO - Initialization successful proceeding to
incident plan.
2017-11-30T00:59:33 - aws_ir.libs.case - INFO - Initial connection to AmazonWebServices
made.
2017-11-30T00:59:41 - aws_ir.libs.case - INFO - Inventory AWS Regions Complete 14 found.
2017-11-30T00:59:41 - aws_ir.libs.case - INFO - Inventory Availability Zones Complete 37
found.
2017-11-30T00:59:41 - aws_ir.libs.case - INFO - Beginning inventory of resources world
wide.  This might take a minute...
2017-11-30T00:59:41 - aws_ir.libs.inventory - INFO - Searching ap-south-1 for instance.
2017-11-30T00:59:42 - aws_ir.libs.inventory - INFO - Searching eu-west-2 for instance.
2017-11-30T00:59:42 - aws_ir.libs.inventory - INFO - Searching eu-west-1 for instance.
2017-11-30T00:59:43 - aws_ir.libs.inventory - INFO - Searching ap-northeast-2 for
instance.
2017-11-30T00:59:44 - aws_ir.libs.inventory - INFO - Searching ap-northeast-1 for
instance.
2017-11-30T00:59:44 - aws_ir.libs.inventory - INFO - Searching sa-east-1 for instance.
2017-11-30T00:59:45 - aws_ir.libs.inventory - INFO - Searching ca-central-1 for instance.
2017-11-30T00:59:45 - aws_ir.libs.inventory - INFO - Searching ap-southeast-1 for
instance.
2017-11-30T00:59:46 - aws_ir.libs.inventory - INFO - Searching ap-southeast-2 for
instance.
2017-11-30T00:59:47 - aws_ir.libs.inventory - INFO - Searching eu-central-1 for instance.
2017-11-30T00:59:47 - aws_ir.libs.inventory - INFO - Searching us-east-1 for instance.
2017-11-30T00:59:47 - aws_ir.libs.inventory - INFO - Searching us-east-2 for instance.
2017-11-30T00:59:48 - aws_ir.libs.inventory - INFO - Searching us-west-1 for instance.
2017-11-30T00:59:48 - aws_ir.libs.inventory - INFO - Searching us-west-2 for instance.
2017-11-30T00:59:48 - aws_ir.libs.case - INFO - Inventory complete.  Proceeding to
resource identification.
2017-11-30T00:59:48 - aws_ir.plans.host - INFO - Proceeding with incident plan steps
included are ['gather_host', 'isolate_host', 'tag_host', 'snapshotdisks_host',
'examineracl_host', 'get_memory', 'stop_host']
2017-11-30T00:59:48 - aws_ir.plans.host - INFO - Executing step gather_host.
2017-11-30T00:59:49 - aws_ir.plans.host - INFO - Executing step isolate_host.
2017-11-30T00:59:50 - aws_ir.plans.host - INFO - Executing step tag_host.
2017-11-30T00:59:50 - aws_ir.plans.host - INFO - Executing step snapshotdisks_host.
True
2017-11-30T00:59:50 - aws_ir.plans.host - INFO - Executing step examineracl_host.
2017-11-30T00:59:52 - aws_ir.plans.host - INFO - Executing step get_memory.
2017-11-30T00:59:52 - aws_ir.plans.host - INFO - attempting memory run
2017-11-30T00:59:52 - aws_ir.plans.host - INFO - Attempting run margarita shotgun for
ec2-user on 54.152.212.162 with /home/ec2-user/.ssh/your-key-name.pem
{
    "uids": ["Lime Signing Key (Threat Response Official Lime Signing Key)
<security@threatresponse.cloud>"],
    "fingerprint": "80DA92CB09161F241C8F9BC918BA980367172B17"
}
```

Kenneth G. Hartman,
ken@kennethghartman.com

```
2017-11-30T00:59:52 - aws_ir.libs.volatile - CRITICAL - GPG key not in trust chain
attempting interactive installation.
{u'uids': [u'Lime Signing Key (Threat Response Official Lime Signing Key)
<security@threatresponse.cloud>'], u'fingerprint':
u'80DA92CB09161F241C8F9BC918BA980367172B17'}
warning: Repository key untrusted
Importing GPG key 0x80DA92CB09161F241C8F9BC918BA980367172B17:
  Userid: "Lime Signing Key (Threat Response Official Lime Signing Key)
<security@threatresponse.cloud>"
  From  : https://threatresponse-lime-modules.s3.amazonaws.com/REPO_SIGNING_KEY.asc
Is this ok: [y/N] y
2017-11-30T01:00:05 - margaritashotgun.repository - INFO - importing repository signing
key 80DA92CB09161F241C8F9BC918BA980367172B17 Lime Signing Key (Threat Response Official
Lime Signing Key) <security@threatresponse.cloud>
{
    "uids": ["Lime Signing Key (Threat Response Official Lime Signing Key)
<security@threatresponse.cloud>"],
    "fingerprint": "80DA92CB09161F241C8F9BC918BA980367172B17"
}

{
    "uids": ["Lime Signing Key (Threat Response Official Lime Signing Key)
<security@threatresponse.cloud>"],
    "fingerprint": "80DA92CB09161F241C8F9BC918BA980367172B17"
}

2017-11-30T01:00:08 - margaritashotgun.repository - INFO - downloading
https://threatresponse-lime-modules.s3.amazonaws.com/modules/lime-4.9.62-
21.56.amzn1.x86_64.ko as lime-2017-11-30T01:00:08-4.9.62-21.56.amzn1.x86_64.ko
2017-11-30T01:00:12 - margaritashotgun.memory - INFO - 54.152.212.162: dumping memory to
s3://cloud-response-6fa6ff4f27d0408f93232dd9adff2618/54.152.212.162-2017-11-30T01:00:06-
mem.lime
2017-11-30T01:00:38 - margaritashotgun.memory - INFO - 54.152.212.162: capture 10%
complete
2017-11-30T01:01:04 - margaritashotgun.memory - INFO - 54.152.212.162: capture 20%
complete
2017-11-30T01:01:30 - margaritashotgun.memory - INFO - 54.152.212.162: capture 30%
complete
2017-11-30T01:01:58 - margaritashotgun.memory - INFO - 54.152.212.162: capture 40%
complete
2017-11-30T01:02:28 - margaritashotgun.memory - INFO - 54.152.212.162: capture 50%
complete
2017-11-30T01:02:56 - margaritashotgun.memory - INFO - 54.152.212.162: capture 60%
complete
2017-11-30T01:03:24 - margaritashotgun.memory - INFO - 54.152.212.162: capture 70%
complete
2017-11-30T01:03:52 - margaritashotgun.memory - INFO - 54.152.212.162: capture 80%
complete
2017-11-30T01:04:21 - margaritashotgun.memory - INFO - 54.152.212.162: capture 90%
complete
2017-11-30T01:04:52 - margaritashotgun.memory - INFO - 54.152.212.162: capture complete:
s3://cloud-response-6fa6ff4f27d0408f93232dd9adff2618/54.152.212.162-2017-11-30T01:00:06-
mem.lime
{'failed': [], 'completed': ['54.152.212.162'], 'total': 1}
2017-11-30T01:04:53 - aws_ir.plans.host - INFO - memory capture completed for:
['54.152.212.162'], failed for: []
2017-11-30T01:04:53 - aws_ir.plans.host - INFO - Executing step stop_host.
```

Kenneth G. Hartman,
ken@kennethghartman.com

9.1. Appendix B - Contents of the virustotal-results.txt file

The following is an example of the results of performing a VirusTotal search using the virustotal-search.py command:

```
$ virustotal-search.py hashlist4virustotal.md5 > virustotal-results.txt
$ cat virustotal-results.txt
4fa4269b7ce44bfce5ef574e6a37c38f;1;1;2017-11-30
19:38:52;31;59;https://www.virustotal.com/file/63210b24f42c05b2c5f8fd62
e98dba6de45c7d751a2e55700d22983772886017/analysis/1512070732/;AVG#ELF:B
itCoinMiner-AK [PUP]#20171130#17.8.3705.0,Ad-
Aware#Application.Miner.R#20171130#3.0.3.1010,AhnLab-
V3#ELF/Gafgyt.2979640#20171130#3.11.1.19380,Antiy-
AVL#RiskWare[RiskTool]/Linux.BitCoinMiner.a#20171130#3.0.0.1,Arcabit#Ap
plication.Miner.R#20171130#1.0.0.827,Avast#ELF:BitCoinMiner-AK
[PUP]#20171130#17.8.3705.0,Avast-Mobile#ELF:BitCoinMiner-P
[PUP]#20171130#171130-
02,Avira#LINUX/BitCoinMiner.muef#20171130#8.3.3.6,BitDefender#Applicati
on.Miner.R#20171130#7.2,CAT-
QuickHeal#ELF.Miner.A.GC#20171130#14.00,ClamAV#Unix.Malware.Agent-
1843238#20171130#0.99.2.0,Cyren#ELF/Application.YEXW#20171130#5.4.30.7,
DrWeb#Tool.Linux.BtcMine.70#20171130#7.0.28.2020,ESET-NOD32#a variant
of Linux/BitCoinMiner.L potentially
unsafe#20171130#16497,Emsisoft#Application.Miner.R
(B)#20171130#4.0.1.883,F-
Secure#Application.Miner.R#20171130#11.0.19100.45,GData#Application.Min
er.R#20171130#A:25.14977B:25.11014,Ikarus#PUA.Linux.Miner#20171130#0.1.
5.2,Jiangmin#RiskTool.Linux.ag#20171130#16.0.100,Kaspersky#not-a-
virus:HEUR:RiskTool.Linux.BitCoinMiner.a#20171130#15.0.1.13,MAX#malware
(ai score=100)#20171130#2017.11.15.1,MicroWorld-
eScan#Application.Miner.R#20171130#14.0.297.0,NANO-
Antivirus#Riskware.BitCoinMiner.ekqhah#20171130#1.0.100.20407,Qihoo-
360#virus.elf.bitcointool.a#20171130#1.0.0.1120,Sophos#Linux/BitCoin-
B#20171130#4.98.0,Symantec#Trojan.Gen.NPE#20171130#1.5.0.0,TrendMicro#H
KTL_COINMINE.GE#20171130#9.862.0.1074,TrendMicro-
HouseCall#HKTL_COINMINE.GE#20171130#9.950.0.1006,VBA32#RiskTool.Linux.B
itCoinMiner.a#20171130#3.12.26.4,Zillya#Tool.BitCoinMiner.Linux.1#20171
129#2.0.0.3438,ZoneAlarm#not-a-
virus:HEUR:RiskTool.Linux.BitCoinMiner.a#20171130#1.0;
8a629fa901ed50f3dd7bba92b4535f03;1;1;2017-09-29
17:14:43;0;64;https://www.virustotal.com/file/78445153afe05dc55a435336b
87011dc14d08ec3c8d126e2986c8c2c60c7ce9b/analysis/1506705283/;;
3bd016b34cd5bd0a562b8f56f0cb971b;1;0;The requested resource is not
among the finished, queued or pending scans
9b587ceee0e1236fff1333bd8752c64e;1;0;The requested resource is not
among the finished, queued or pending scans
```

Kenneth G. Hartman,
ken@kennethghartman.com