Análisis Forense Automatizado para Computación en la Nube

Daniel Rodríguez
@dvirus
Alex Rincón
@nemesis545

Resumen

Este trabajo de investigación se centra en el uso de la herramienta de gestión de configuraciones Puppet y diversas herramientas basadas en software libre para automatizar el proceso de adquisición y preservación evidencia volátil en máquinas virtuales (VM's) o contenedores con sistema operativo Linux. Con el fin de mejorar los tiempos de respuesta ante un incidente de seguridad.

Palabras clave: Forense, Cloud, Virtualización, Contenedores, Investigación

Introducción

El presente documento permite describir el proceso de gestión centralizada y la automatización para la recolección y preservación de evidencia durante un proceso de análisis forense en un entorno basado en sistemas linux. La motivación para desarrollar esta investigación se debe a los retos con los que cuenta la computación en la nube, como el acceso físico, la diversidad de distribuciones, los niveles de privilegios, el almacenamiento distribuido y la cantidad de máquinas virtuales o contenedores que pueden estar involucrados dentro de un mismo tenant (cliente).

Metodología

La metodología a seguir para la implementación del proceso de automatización, se basa en el proceso de análisis forense definido por El Instituto Nacional de Estándares y Tecnología (NIST)[1]

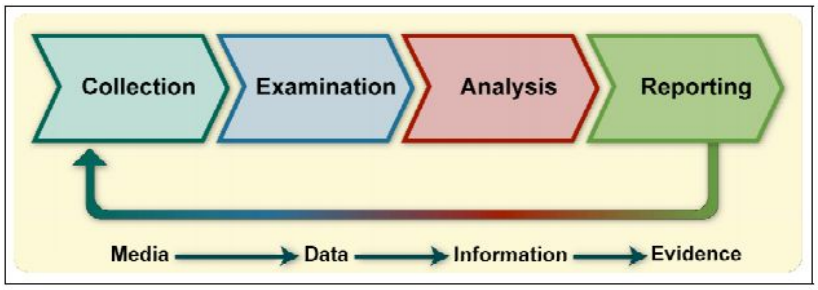

Fig 1: Proceso de Análisis Forense Digital

Acerca de Puppet

Puppet es una herramienta diseñada para administrar la configuración de sistemas similares a Unix y a Microsoft. Es utilizada para el manejo de grandes cantidades de servidores a la vez. Programado en ruby, puppet permite gran flexibilidad y modularidad estableciendo como único límite la imaginación del administrador.

La arquitectura de puppet consiste en la clasificación de los nodos o equipos a administrar, los cuales mantienen una comunicación periódica con el puppet-master (Servidor Central), con el fin de recibir nuevas configuraciones y reportar el estado de los cambios efectuados sobre los nodos (Cliente).

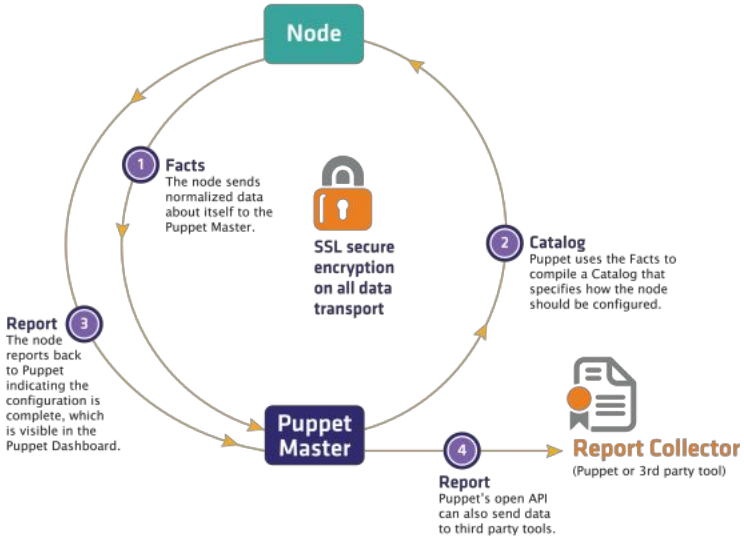

Fig 2. Funcionamiento de Puppet

Colección de información volátil

Con el fin de evitar contaminar el sistema comprometido o afectado se debe recolectar la evidencia con un orden específico según el nivel de volatilidad de cada elemento como se describe en el RFC 3227 Guía para la colección y archivo de evidencia [2] el cuál se resume en la siguiente tabla.

Orden	Elementos
1	Registros, información en caché
2	Tabla de enrutamiento, arp caché, tabla de procesos, estadísticas del kernel, memoria RAM
3	Archivos temporales del sistema, Swap
4	Disco Local
5	Logs remotos y datos de monitoreo relevantes
6	Configuración física, topología de red
7	Datos contenidos en medios de archivo

Las clases de puppet [3] han sido desarrolladas para recopilar la información según el principio de volatilidad.

La primera clase permite adquirir contenido en memoria RAM, Procesos en ejecución, conexiones de red, archivos en uso y sesiones, a través del uso de herramientas basadas en software libre, las cuales podrán estar preinstaladas en el sistema o ser desplegadas mediante puppet.

La información que recopila la clase que genera puppet utiliza los siguientes comandos:
Memoria RAM:
sudo insmod lime-3.X.X-X-generic.ko "path=/media/dvirus/memoryRAM.lime format=raw"

Información del sistema:
hostname, whoami, logname, id, uptime, uname -a, printenv, cat /proc/version

Información de Red:
ifconfig -a, who, route -n, netstat -anp, ss, arp -a

Información de Procesos:
top, pidstat

Colección de información no-volátil

Para la recopilación de información no volátil se generan clases específicas dependiendo el rol del servidor con el fin de recopilar información como:
Archivos de configuración, usuarios y grupos, Archivos de contraseñas, tareas programadas, logs, archivos de aplicaciones, archivos de datos, entre otros.

Preservación de la evidencia

Durante el proceso de recolección de los artefactos que servirán como evidencia se deben generar los hashes de integridad los cuales serán validados mediante una clase en puppet, el mecanismo de verificación a usar en todos los casos será SHA1.

Ejemplo:
fb5edc1c597e51889b02e3e1f55a83bc857f7fa6 memdump.lime

Descripción de la solución

La solución se compone de una máquina virtual que contiene las principales herramientas para el proceso de adquisición de evidencia forense, esta máquina actúa como el nodo principal “master” y tiene la funcionalidad de establecer las tareas que ejecutarán los nodos o máquinas virtuales que se encuentren dentro del ambiente virtual del cliente (tenant)

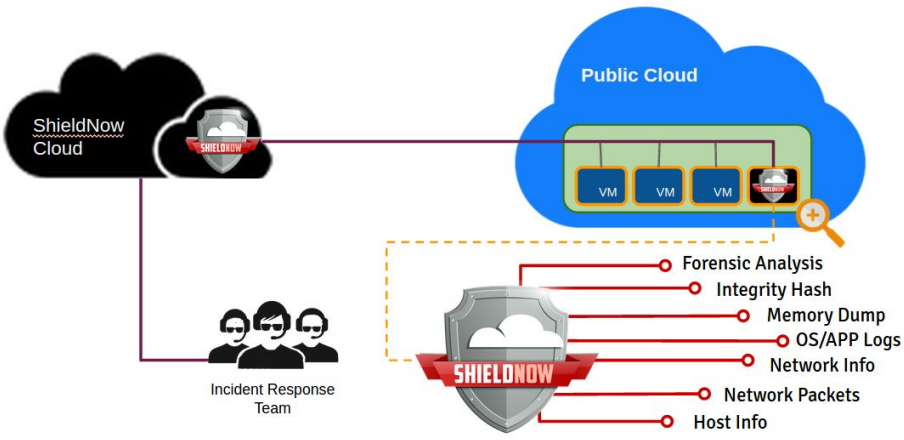

Fig 3: Solución propuesta

Cuando el equipo de Respuesta a incidentes declara el incidente se ejecuta la acción sobre el nodo principal, quien enviará la señal al nodo afectado, este iniciará las tareas indicadas por el master, las cuales están definidas en las clases de puppet.

```
exec { "reportarse":      #main exec all command in one
  command => "insmod /lib/modules/${kernelrelease}/updates/dkms/lime.ko \"path=/tmp/demo2.lime format=lime\" && dd if=$(cat /etc/fstab|gr
  path    => "/usr/local/bin/:/usr/bin/:/bin/:/sbin/:/usr/sbin/",
 }
}
node default {
 file {'/root/.ssh/id_rsa':                                    # resource type file and filename
  ensure  => present,                                         # make sure it exists
  mode    => 0600,                                            # file permissions
  source  => 'puppet:///files/llaves/llave-priv',
  owner   => 'root',
 }
 file {'/root/.ssh/id_rsa.pub':                                     # resource type file and filename
  ensure  => present,                                         # make sure it exists
  mode    => 0644,                                            # file permissions
  source  => 'puppet:///files/llaves/llave-pub',
  owner   => 'root',
 }
```

```
package {["openssh-client","openssh-server","tshark","rsync","lime-forensics-dkms","htop","sudo","linux-headers-amd64"]:
  ensure => installed
 }
```

Fig 4: Clase Puppet en el servidor Master

El nodo recibirá la petición del master y ejecutará una serie de rutinas para recolectar logs del sistema, tabla arp, tabla de enrutamiento, volcado de memoria swap, volcado de memoria RAM, captura de red en formato .pcap y los hash de integridad de los archivos.

Una vez se procesa la información esta es enviada a través de SSH a un CHROOT que prepara el máster para recibir la información del nodo.

```
2cf0de7966fcc238b12f42df621a6beee55dba17b39be620cd147295afebc0c9a6bd58a0270cfb160edd16fbf46e253
793e5eed943d57a2db1881d72e0c9d5c4  /tmp/artefacto/auth.log
d13254e9a6d6d12af1765b099d14e3c8742b731c98993ded94783883946a84a3b7c46f2e587a58ab6965c313125d
77e039ec05e80c51e803a11d5d456005ed8c  /tmp/artefacto/dataw
582939b8399b80852ea83500dd1ac244b2a3332212085777207e8166e0673f19b00623c903edb91f729626491f52
7e575b9f7b6157e07d2473da07a46952bff5  /tmp/artefacto/swap.swap
3f73d4a9591f5ba47c65a739ec8c0bf0c3a8f41bef778479374b4588af7e7009ba12a4405d9937925b7825e6f51974
ac28f628f6bfcfea58bf1994aa04f5d785  /tmp/artefacto/memorydump.lime
06f9deb3034b5cf649c7aa3e07430417a868e64e80930564a29f52d64d1d3e6a9edf651698f97f204c1777bc748e9b
653d39c7fd1c9521592332019c04a67cad  /tmp/artefacto/messages
37ec32ee176380e81a2ce7e25eb30f662b5097d7a2fcd54af0960d29a14e01a03aac63202c906db0192a8c8cd4f6d
f2a033d11c0c3879b8b4d3a8b85512babdc  /tmp/artefacto/netdump.pcap
```

Fig 5: Evidencia Recopilada

Trabajo Futuro

La primera fase requiere la configuración de puppet para recopilar información en ambientes bajo sistema operativo Microsoft Windows y un ajuste de sellado de tiempo. El proyecto continúa con la implementación de una interfaz web para operar los nodos de forma amigable y automatizar el proceso de análisis y reporte.

Conclusiones

El proceso de recolección de artefactos para el análisis forense mediante el uso de herramientas de software libre y gestionados con la herramienta puppet permite actuar de forma rápida ante un incidente sobre una instancia con sistema operativo GNU/Linux, este escenario es ideal para la administración de múltiples nodos, escenario muy comunes en entornos de virtualización.

Las ventajas de usar puppet y mantener una librería de herramientas preinstaladas sobre las máquinas virtuales ayuda a que se evite la contaminación de la evidencia y permite estar preparado para analizar un incidente que afecte cualquiera de los pilares de la seguridad de la información.

Bibliografía

[1] NIST (National Institute of Standards and technology. Guide to Integrating Forensic Techniques into Incident Response, Special Publication 800-86. Disponible en: http://csrc.nist.gov/publications/nistpubs/800-86/SP800-86.pdf

[2] D. Brezinski. RFC3237 - Guidelines for Evidence Collection and Archiving, Order of Volatility Disponible en: https://tools.ietf.org/html/rfc3227#section-2.1

[3] Puppet 4.2 Reference Manual - Language: Classes. Disponible en: https://docs.puppetlabs.com/puppet/latest/reference/lang_classes.html

[4] Michael Hale Ligh, Andrew Case, Jamie Levy, AAron Walters. The Art of Memory Forensics: Detecting Malware and Threats in Windows, Linux, and Mac Memory, Julio 2014

[5] Cameron H Malin. Malware Forensics Field Guide for Linux Systems, 2014