Sandia National Laboratories

CHIRP

Cloud Hypervisor Forensics and Incident Response Platform

Vince Urias

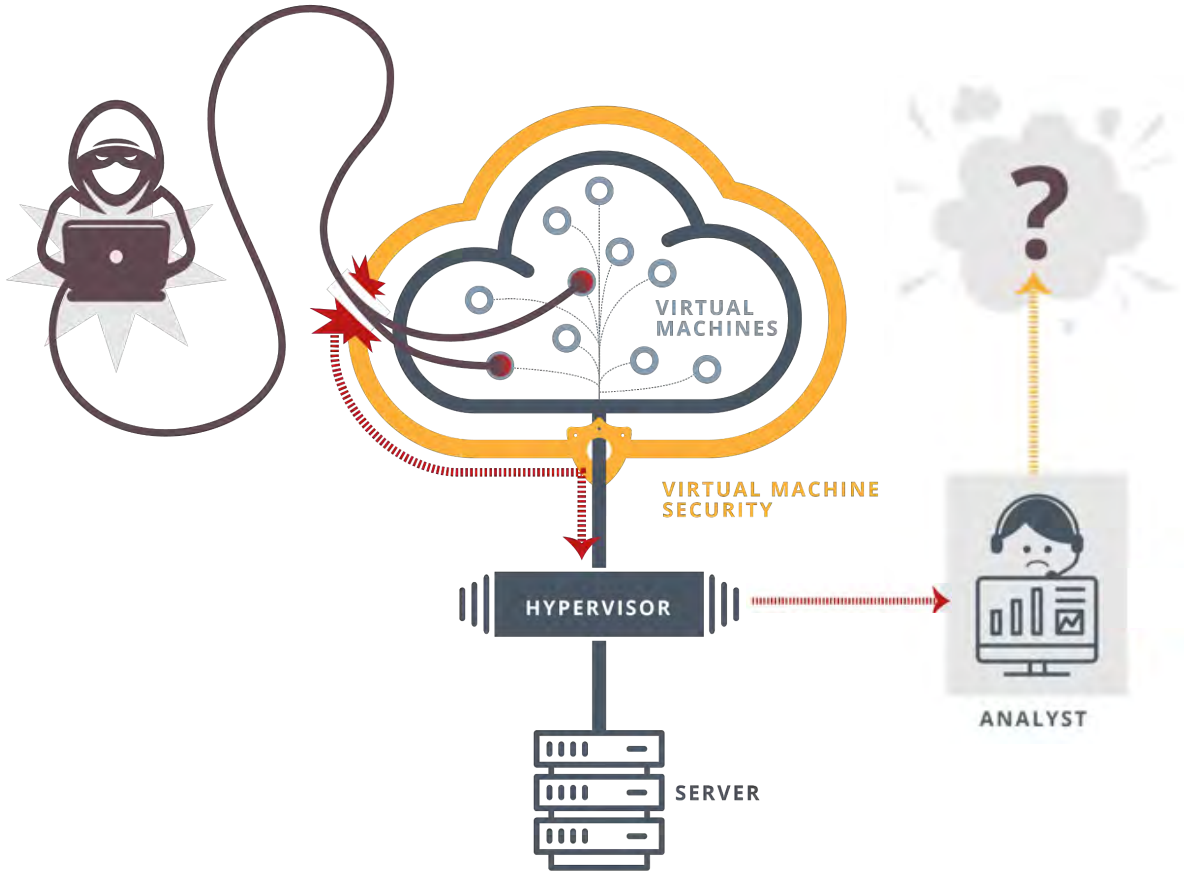
SANDIA NATIONAL LABORATORIES
DOE PACT VIRTUAL SHOWCASE

SAND2020-6433 PE

NNSA U.S. DEPARTMENT OF ENERGY

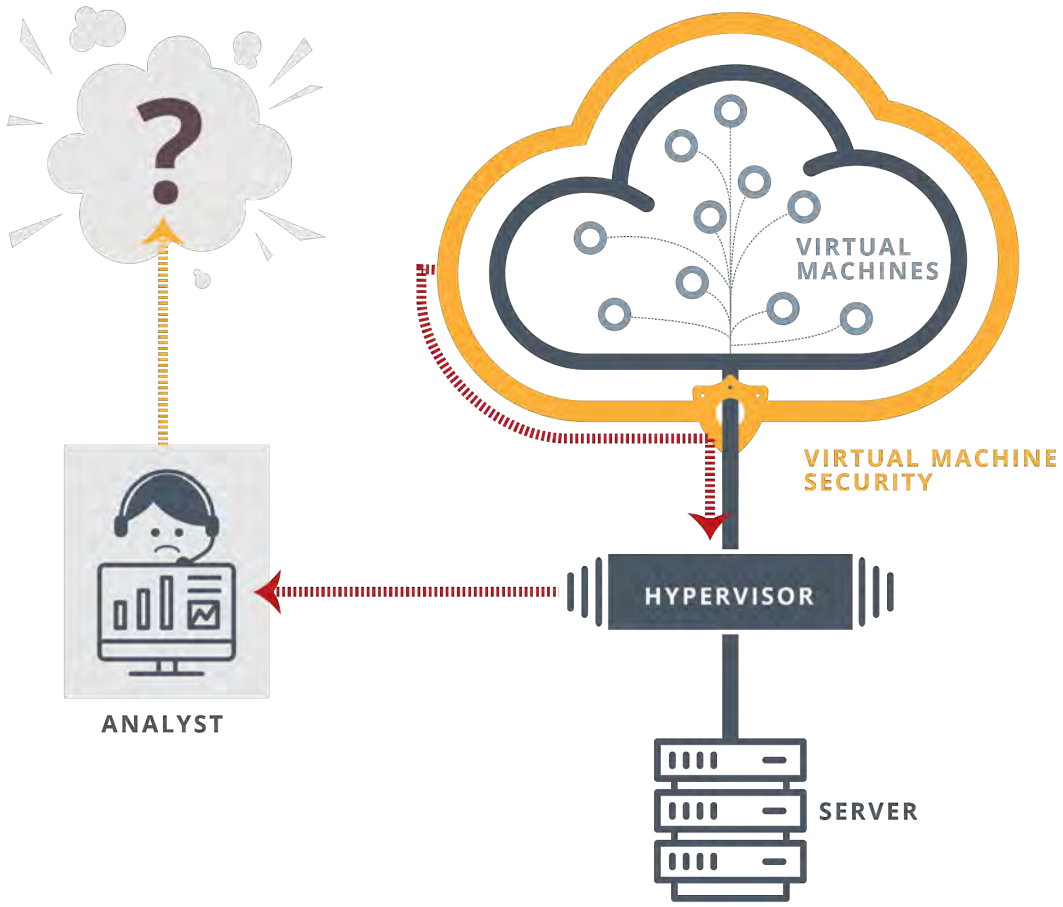
Business Problem

When cyber incidents occur in the cloud, the SOC Analyst has no visibility beyond the hypervisor.



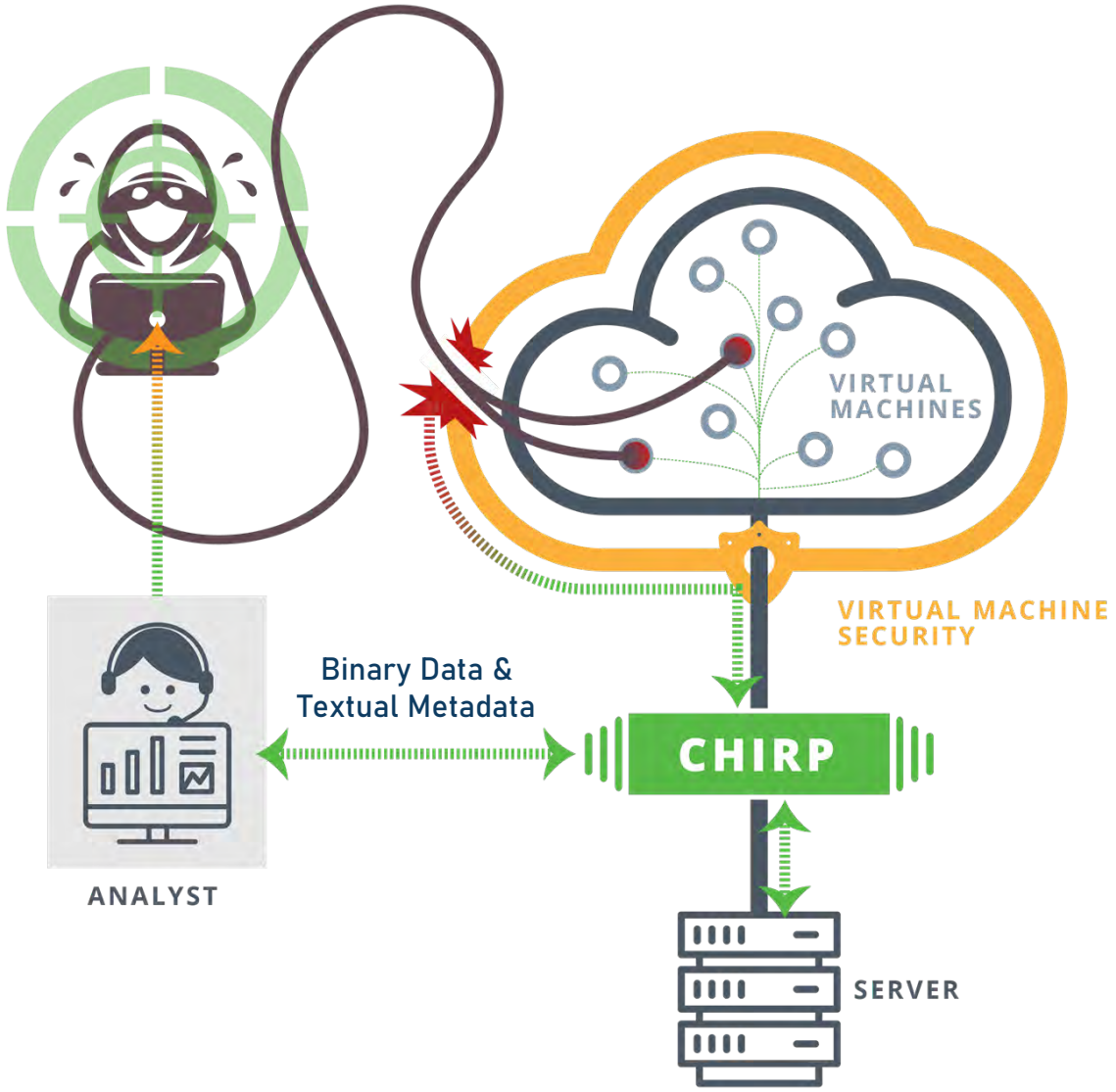
Technical Challenge

- Ephemeral nature of cloud environment
- Hypervisor instrumentation with minimal overhead
- Hypervisor agnostic
- OS agnostic
- Common data model
- Broad data set collection



Solution Architecture

CHIRP



Virtual Machine Introspection (VMI)

- Sandia has developed a custom VMI implementation to address the issue of overhead.
- Platform agnostic
- Text and binary data collection
- Allows correlation with other sources

Development History & Results

CHIRP

Principal Investigator
Vince Urias

- 20 Years of Cyber experience at Sandia Labs
- Numerous national awards

Developmental History

2016 Initial Development starts

2017 First disclosed (Dec)

2019 R&D 100 Winner (May)

Funding History
~$1M LDRD

Research Team
William Stout
Caleb Loverro

Technology Readiness Level (TRL):
TRL 6 as of June 2020

IP Protection

- Patent Application # 16/051,005 filed July 31, 2018
- Copyright approved for commercial licensing – May 2018

Market Validation
2 government deployments

Partner splunk>

Lead Sandia National Laboratories

Technical Requirements & Benefits

CHIRP

TECHNICAL REQUIREMENTS

- Access to the Hypervisor
- Designed for IaaS from the start
- Off prem and on prem cloud-ready

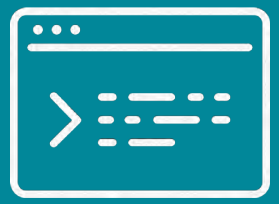
BENEFITS

- Lightweight
- Real-time dynamic response
- Configurable logging → incident response or forensics

Technology Roadmap

CHIRP

Management Dashboard



Expanded Analytics

Improve Deployment

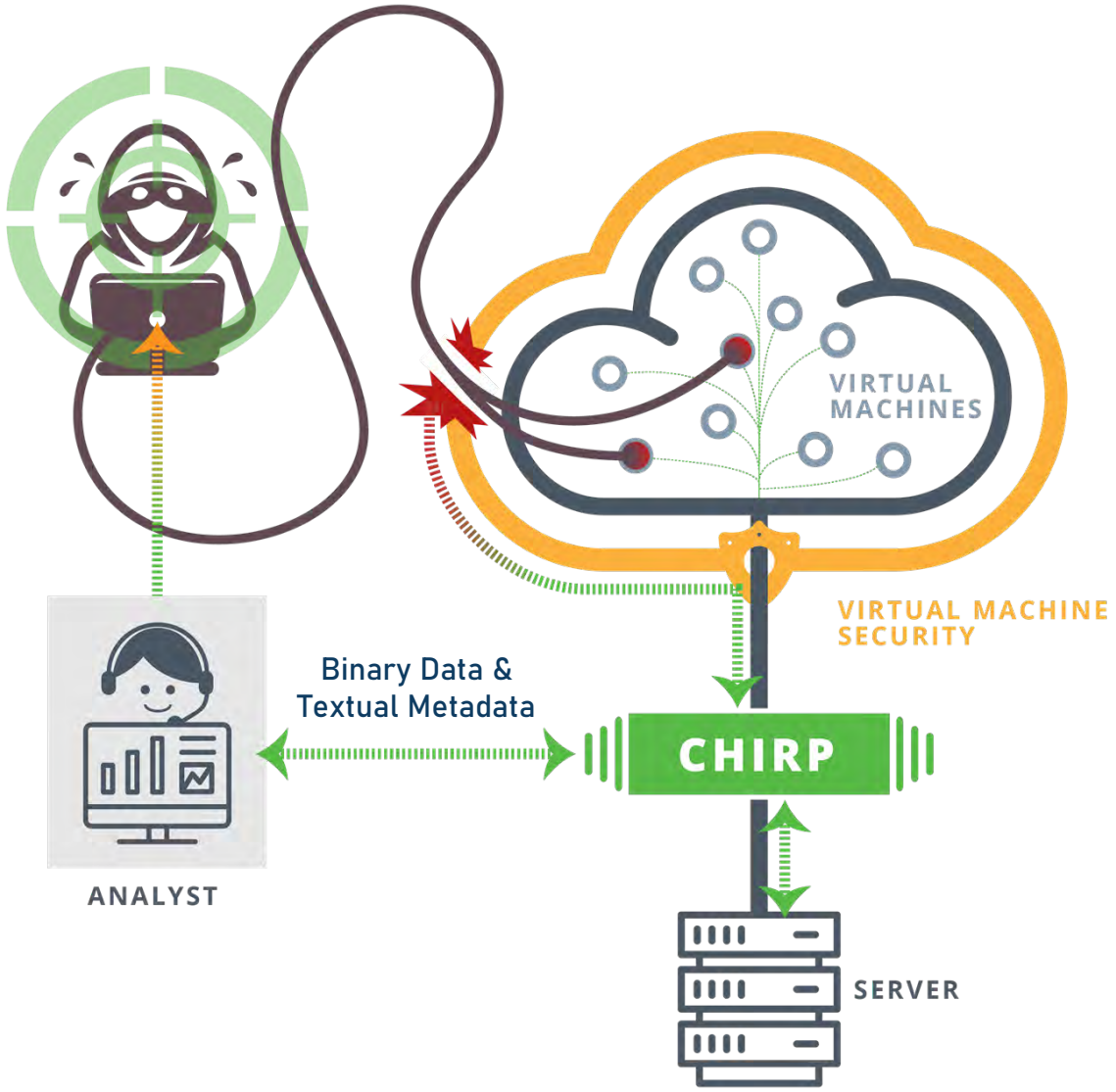
Rigorous Documentation

Private sector pilot for market validation

Bringing CHIRP to Market


Sandia National Laboratories
CHIRP
Cloud Hypervisor Forensics and Incident Response Platform
DOE PACT VIRTUAL SHOWCASE
DEMONSTRATION

Solution Architecture

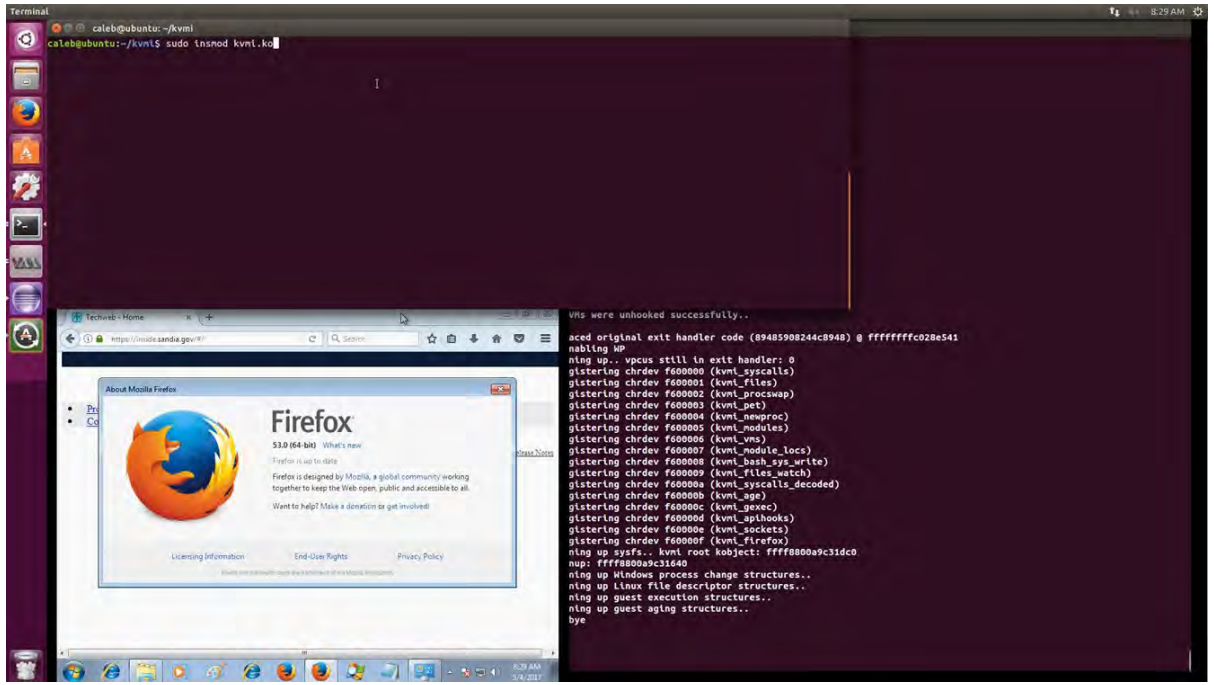


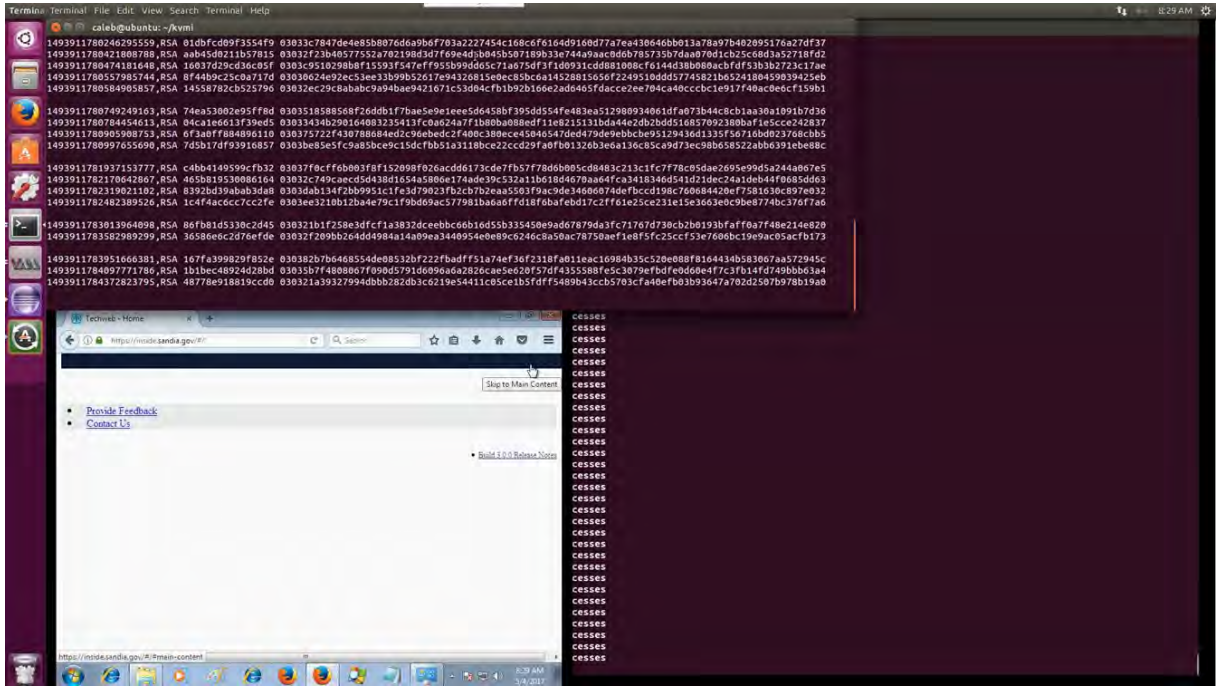
Vignette Demonstrations

- Module Dumping
- Key Extraction
- Shell
- Carving
- System Call Decoding
- Reverse Engineering Pipeline
- Big Data Extraction

Key Extraction

CHIRP
DEMONSTRATION



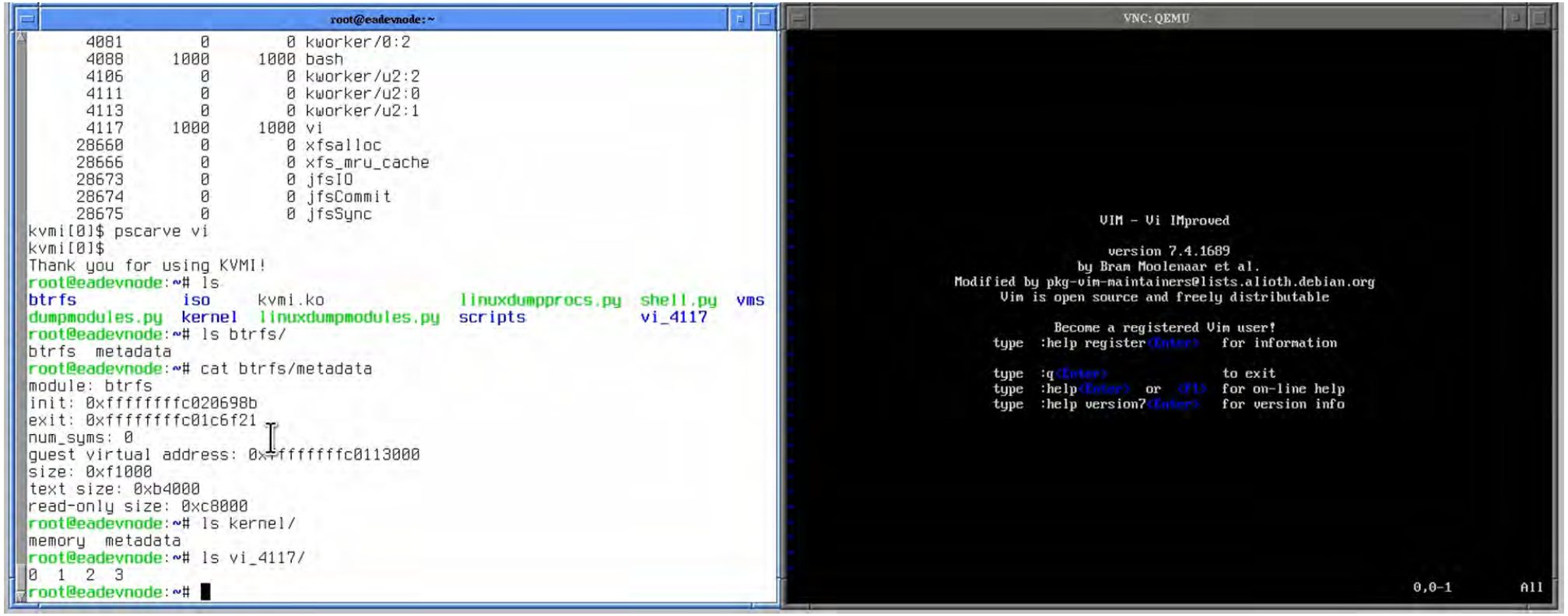


Keys used for encryption are extracted from processes (e.g., Firefox, Windows LSASS, etc.)
Here, Firefox browses to sandia.gov, RSA keys for the session are extracted.

Able to see within the session… eliminating man-in-the-middle attacks.

Carving

CHIRP
DEMONSTRATION

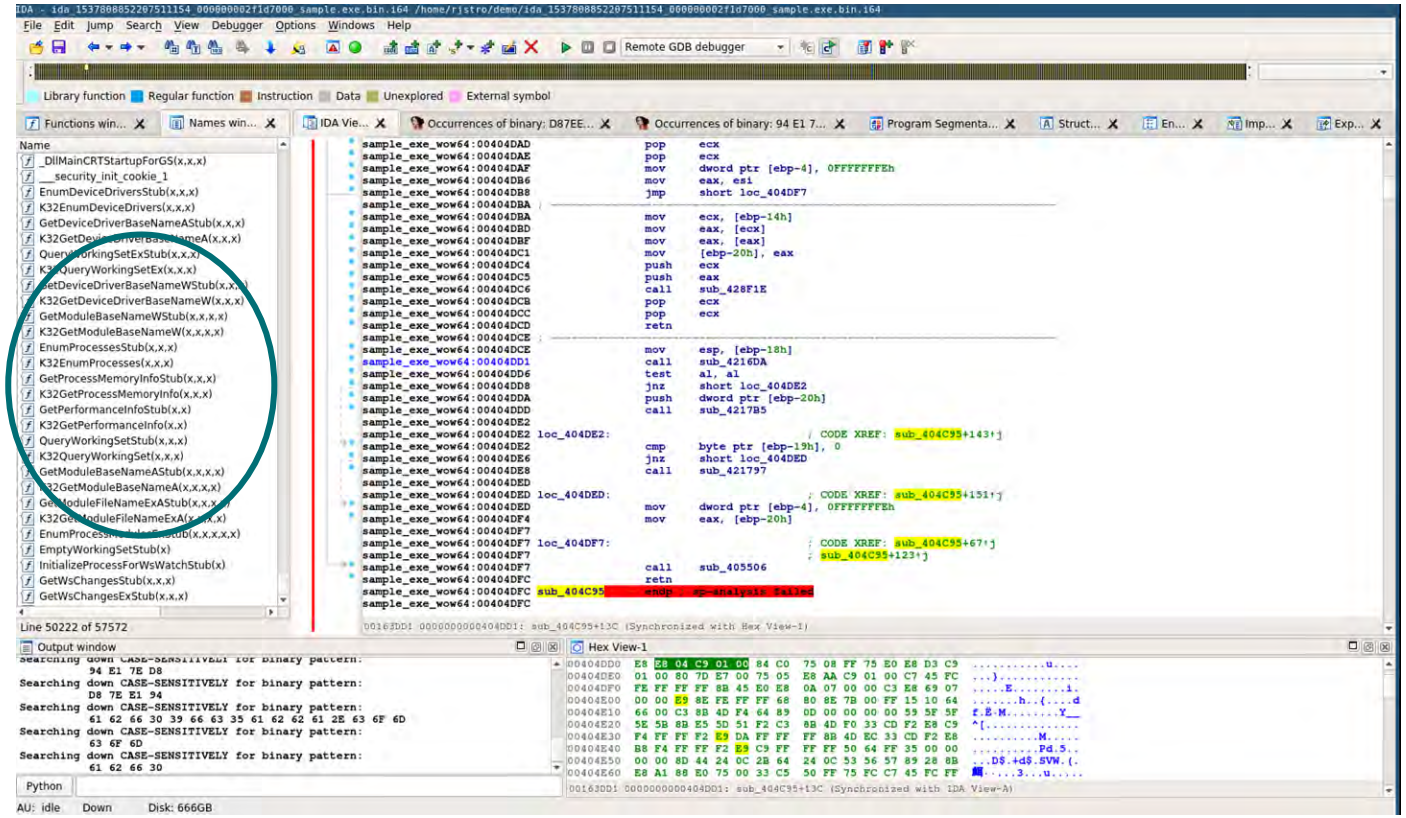


This example shows how the analyst can carve the VI text editor application (process) from the VM

Reverse Engineering Pipeline

- Track all memory for a process, including all libraries and OS code (not just the binary itself).
- Copy of all these mappings as binary that can be loaded into IDA, with multiple copies so program state can be analyzed at different times during execution.
- An idapython script is used to relocate all those mappings to correct locations, providing the reverse engineer deep insight into a given process.

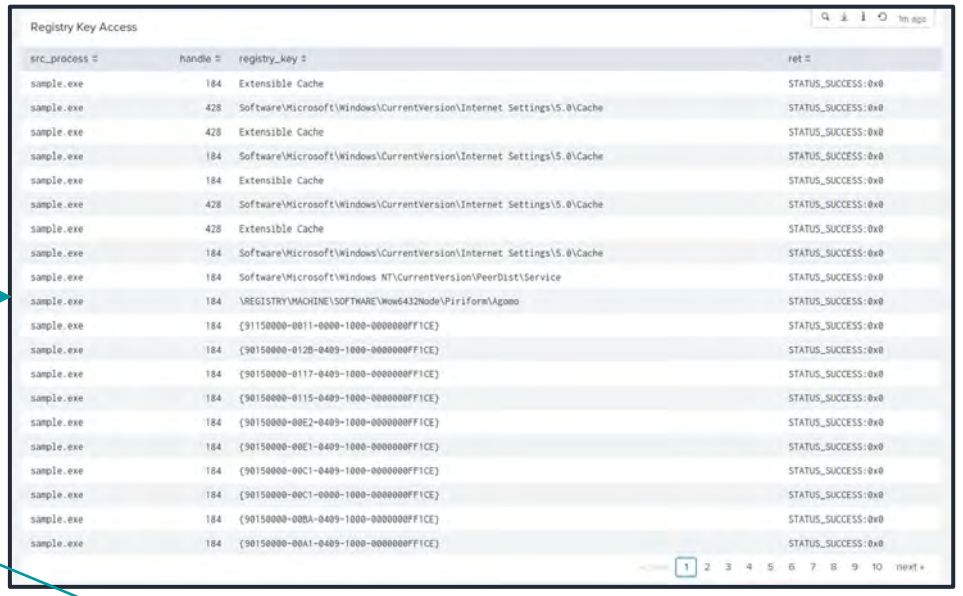
Automatically gather symbols and library information, and correctly relocate (in memory)

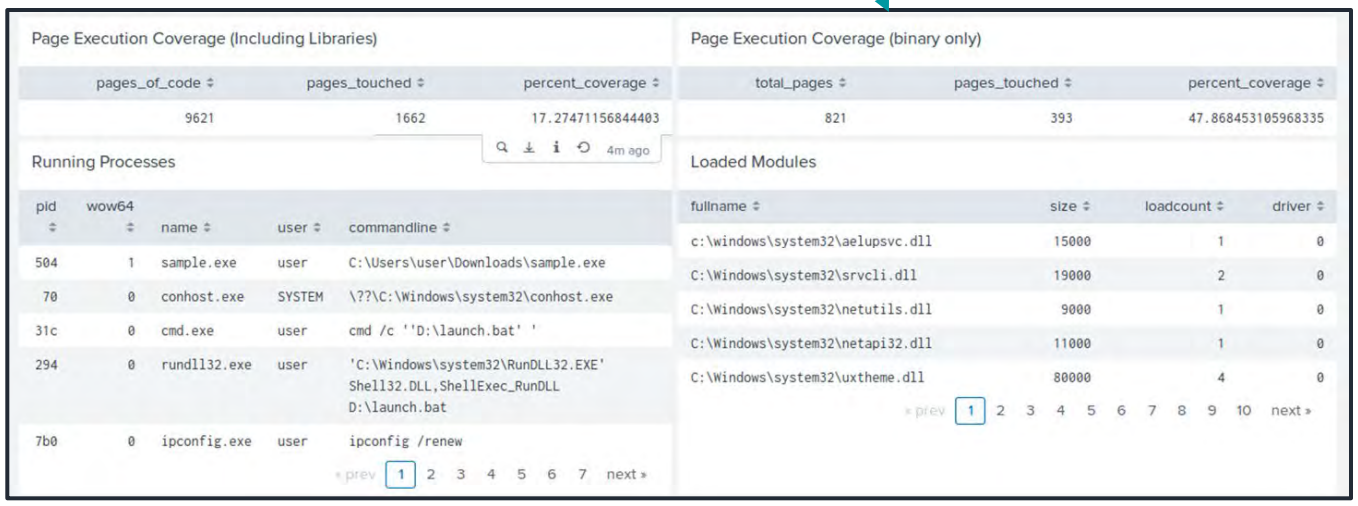


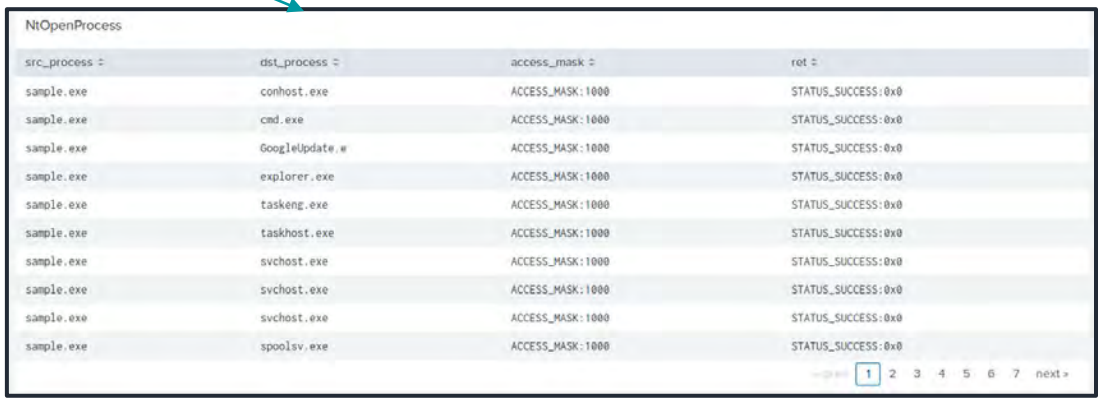
Big Data Extraction

CHIRP
DEMONSTRATION

Feature	Linux	Windows	Mac
Raw syscalls	Y	Y	Y
Decoded syscalls	Y	Y	Y
PID/proc name extraction	Y	Y	Y
Guest execution	Y	Y	N
Kernel carving	Y	Y	N
Module carving	Y	Y	N
Process carving	Y	Y	N
Process tracking (start & exit)	Y	Y	N
File extraction	Y	Y	N
Biometrics	Y	N	N
Socket chardev	N	Y	N





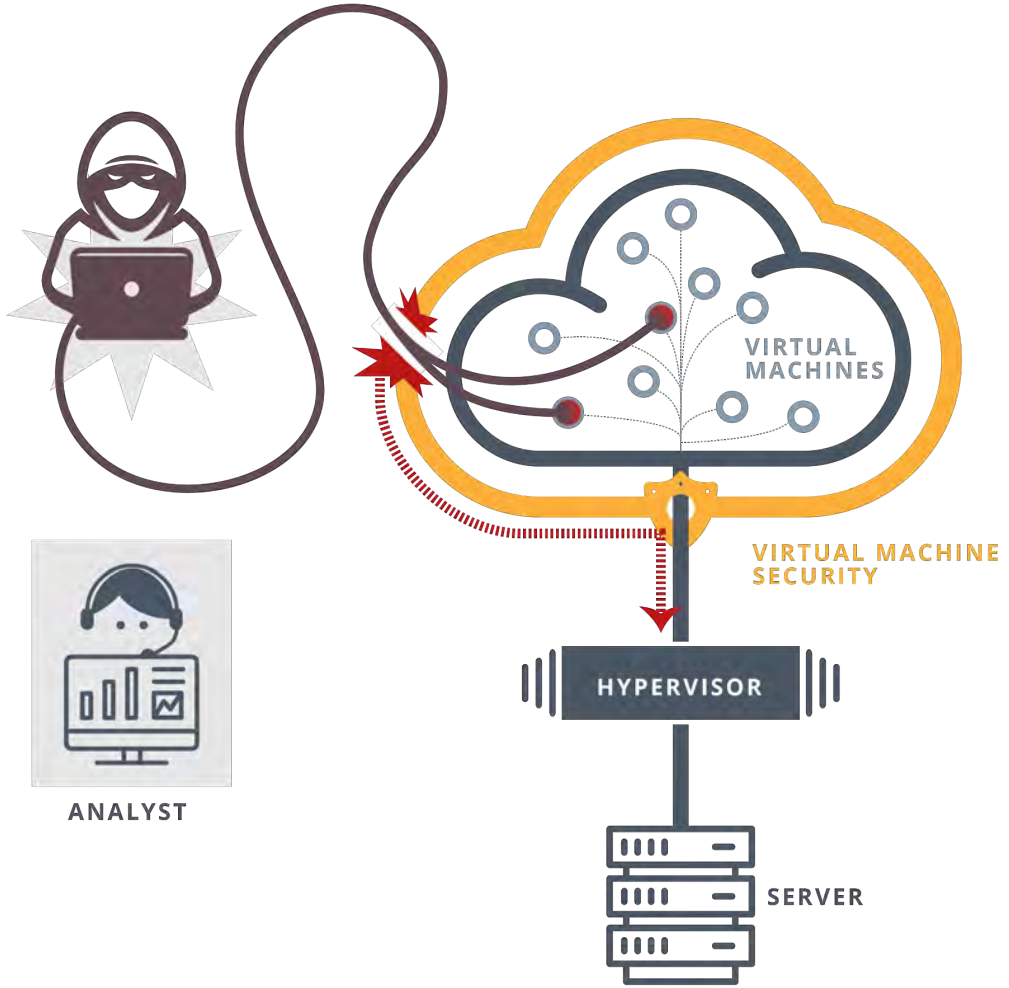


Reserve Slides

Business Problem

CHIRP

Real-time forensics and incident response for Cloud Service Providers providing deeper analytics of security incidents—leading to better protection and improved visibility.



Solution Architecture

CHIRP

Virtual Machine Introspection (VMI)

- Sandia has developed a custom VMI implementation to address the issue of overhead.
- Platform agnostics
- Text and binary data collection
- Allows correlation with other sources

Cloud user accesses the cloud system to instantiate a VM

The CHIRP VMI module is loaded on the Cloud host

The CHIRP intercepts communication between guest and host

Binary data is extracted from the VMI and output to the host

Textual metadata is extracted from the VM and output to the host

Analyst accesses CHIRP to perform duties

CHIRP Process Flow

Cloud user accesses the Cloud system to instantiate a VM

- VM specification is sent to a hypervisor on the host.
- Hypervisor allocates resources to the guest VM and starts it.

The CHIRP VMI module is loaded on the Cloud host

- CHIRP is injected into kernel-space of the host Cloud server
- CHIRP detects hypervisor and places itself in the same administrative domain

The CHIRP intercepts communication between guest and host

- CHIRP hooks VM-exit handler, assuming hypervisor functions
- Dynamically detects all VM Operating Systems and state on the host

Binary data is extracted from the VM and output to the host

- Binary data (memory, files, documents, malware, etc.) stored to the host
- Binary data are analyzed

Textual metadata is extracted from the VM and output to the host

- Textual content (e.g., logs) are ingested by a Security Information and Event Management (SIEM) system
- Textual data is correlated against other data (network, system) in the SIEM

Analyst accesses CHIRP to perform duties

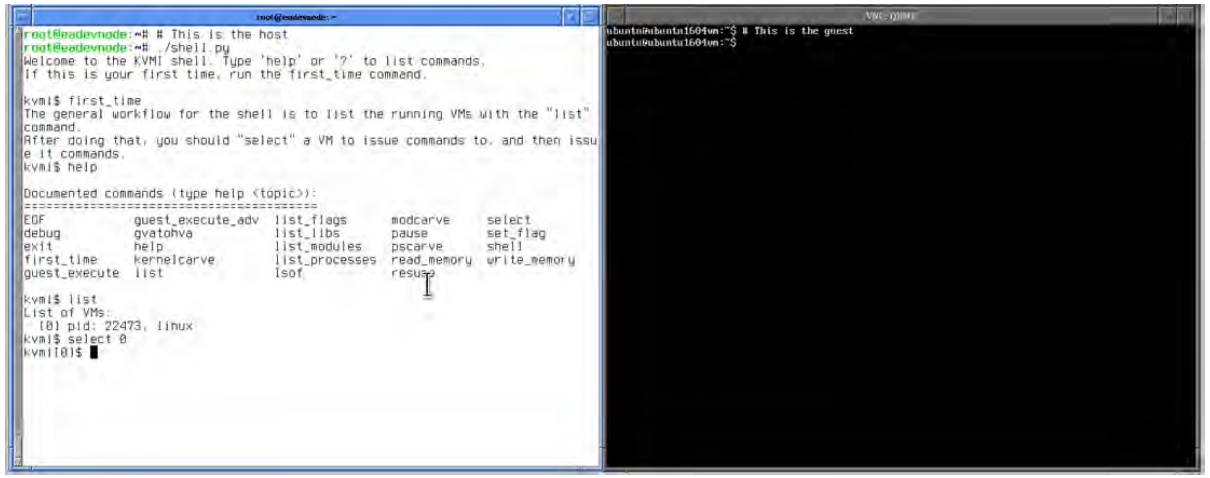
- Inspects binary data (executes files, reverse engineers malware, etc.)
- Inspects textual data to identify indicators of compromise or attack
- Modifies VM state as needed to aid/prolong an investigation

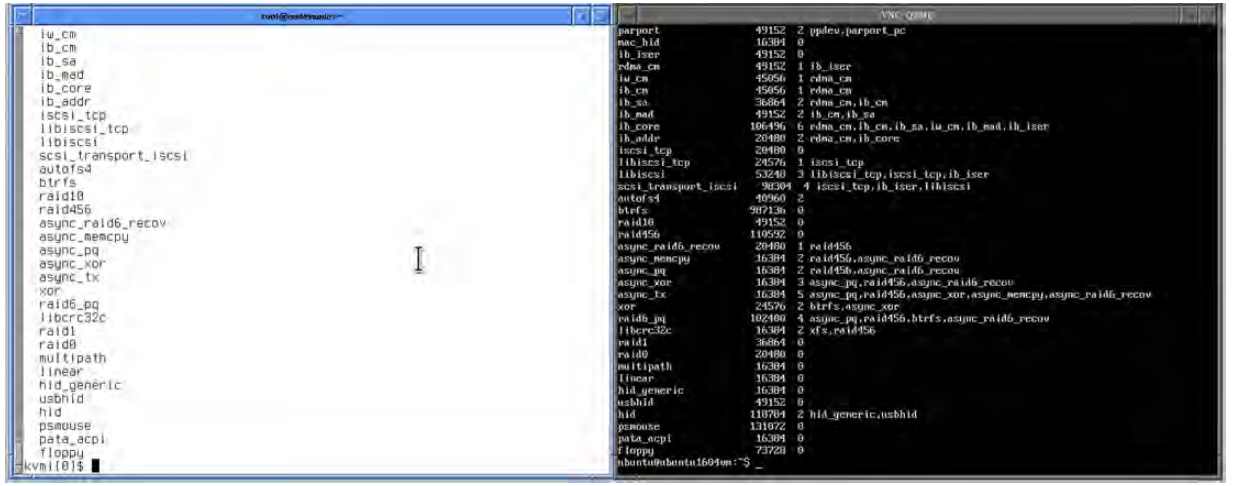
Demonstration Outline

Vignette Demonstrations

- Module Dumping
- Key Extraction
- Shell
- Carving
- System Call Decoding
- Reverse Engineering Pipeline
- Big Data Extraction

Interactive Shell



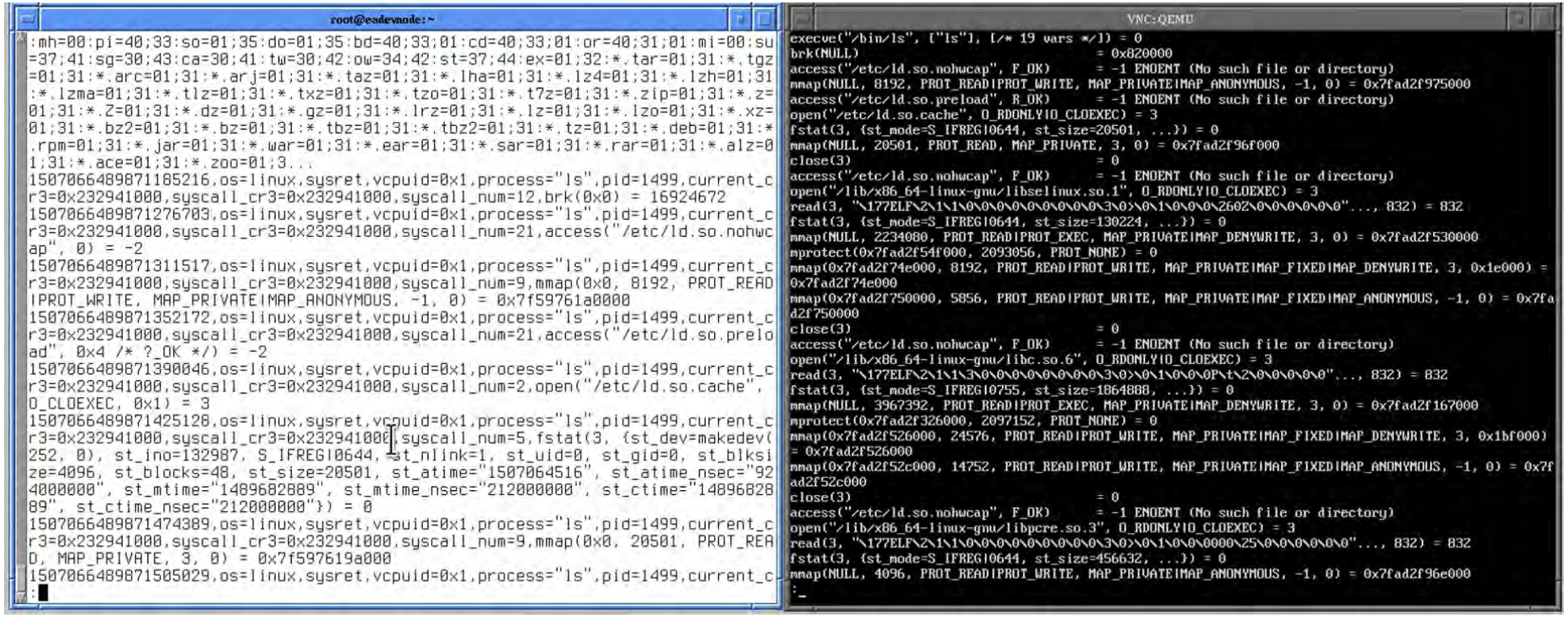


An interactive shell can be used by the analyst to gather information non-obtrusively from the VM in real-time.

- List VMs
- List libraries in use
- List modules executed
- List file descriptors

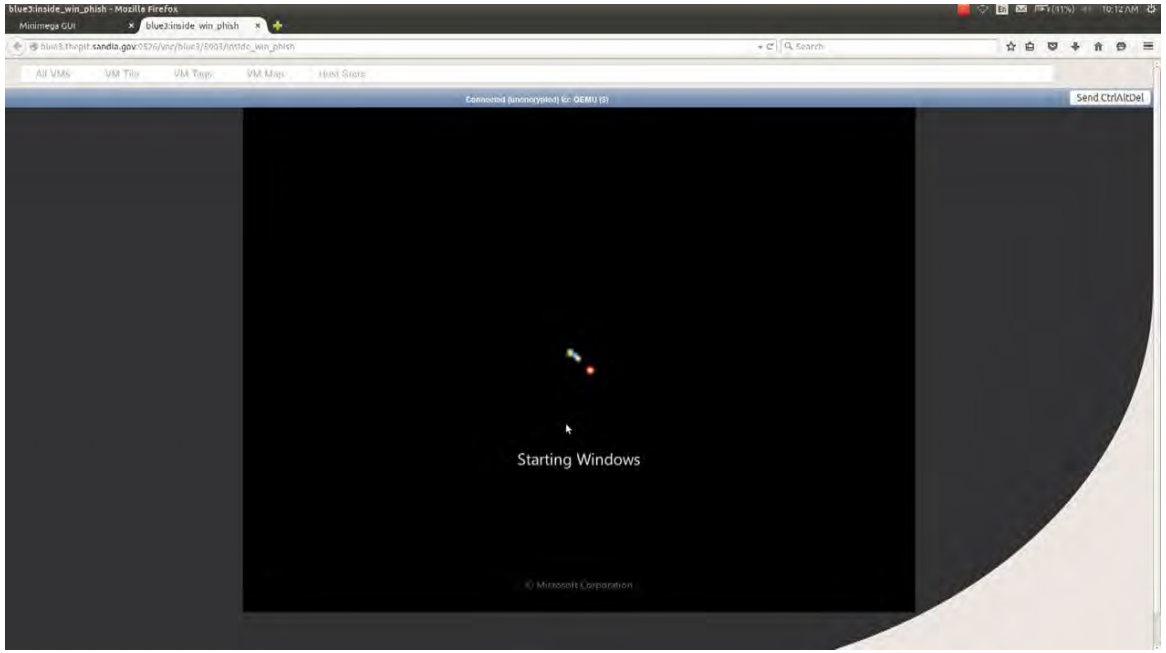
- List running processes
- Write into memory/process
- Start processes on the VM
- Carve objects from memory
  - Files
  - Operating system kernel

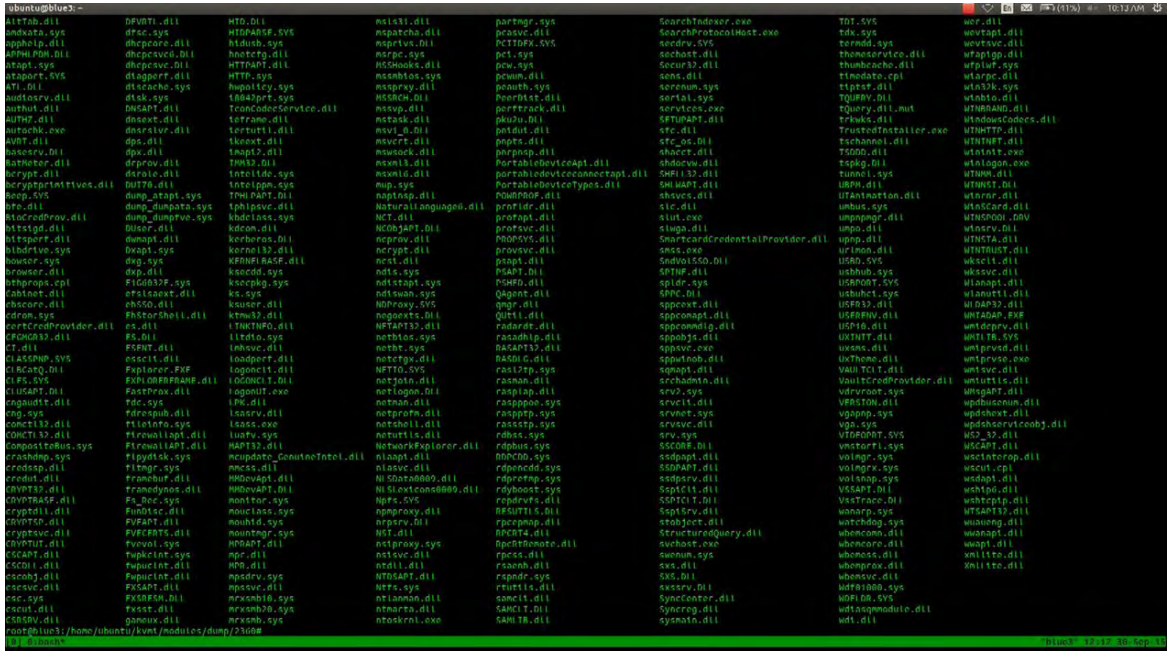
System Call Decoding



This example shows how an analyst can further watch actions on the VM through system call analysis. Decoded system calls with parameters are shown in the shell, with an strace of the ‘ls’ program in the right side (VM).

Module Dumping





Executed binaries are extracted automatically from the guest virtual machine.
Here, on initial boot, 500 modules (dynamic linked libraries (*.dll), executables (.*exe, *.sys)) are extracted.

From boot up onward, CHIRP captures valuable information.