Galahad

A Secure User Computing Environment for the Cloud

IARPA VirtUE

Matt Leinhos
matt.leinhos@starlab.io

Derek Straka
derek.straka@starlab.io

September 6, 2017

Team Members

1221 Connecticut Ave., Suite 4
Washington DC 20036

Jon Tourville
Derek Straka
Kelli Little
Matt Leinhos

Raytheon
BBN Technologies

10 Moulton Street
Cambridge, MA 02138

Alex Jordan
Stephanie Gavin

Agenda

- Goals and Motivation
- Galahad Approach
- Research and Development Objectives
- Galahad Use Case
- Galahad VirtUE
- Galahad Security
- Galahad User Experience
- Galahad Lifecycle Management
- Galahad Sensing and Logging
- Metrics
- Schedule and Milestones

Goals and Motivation

- **Objective: Detection and mitigation of threats attempting to exploit, collect, and/or effect user computing environments (UCE) within public clouds**

- **Cloud service providers have not offered any game changing security solutions**
  - Adversaries can leverage an arsenal of capabilities used to succeed
  - Providers cannot necessarily be trusted

- **Current end-point security solutions and analytical approaches are not tuned for cloud environments**

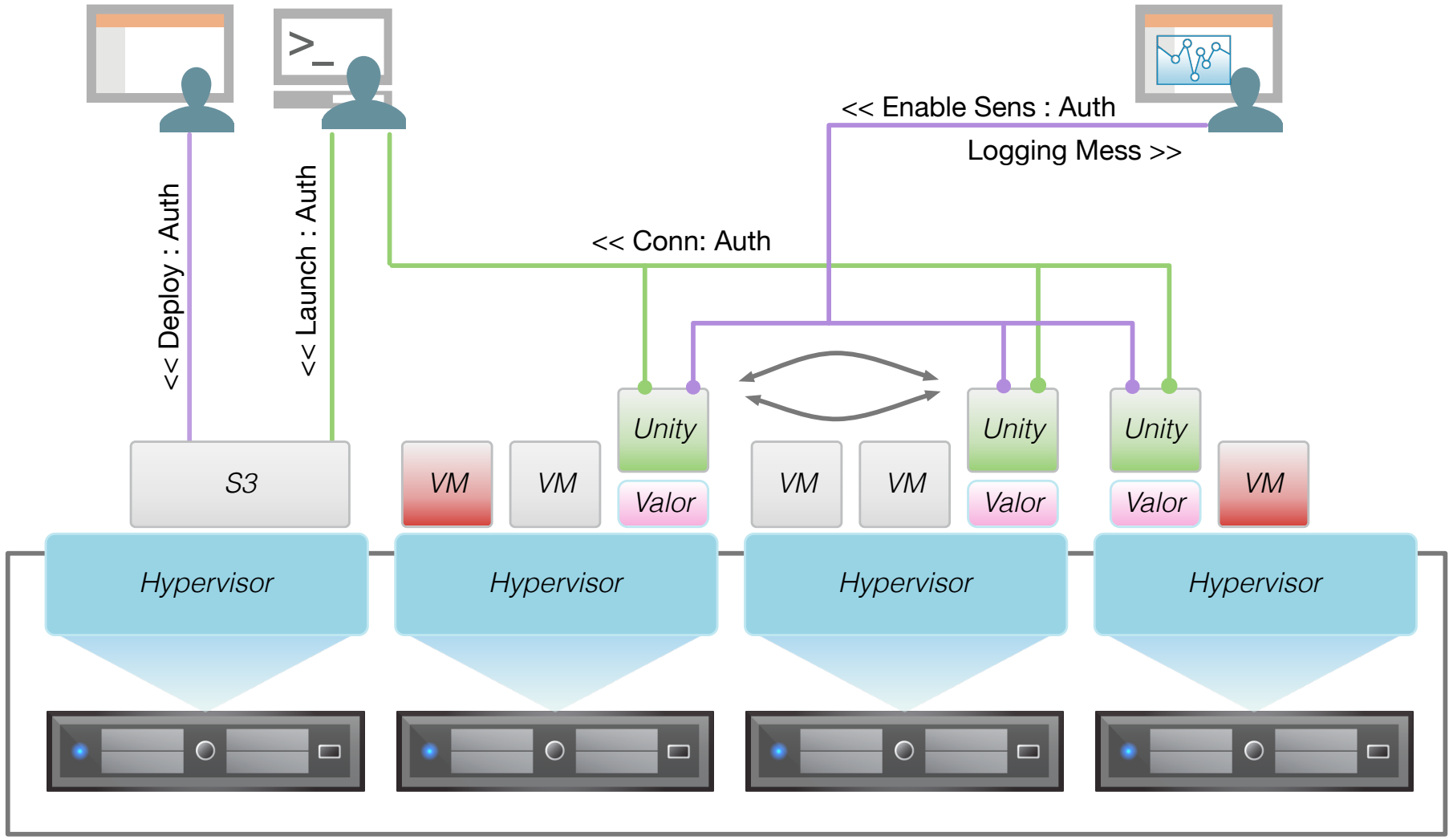
Galahad Approach

- **To combat threats in a public cloud, isolate, protect what is controlled, and maneuver**
  - Do not attempt to establish trust
  - Do not require special cloud services, e.g., dedicated servers
  - Impede the ability of adversaries to operate within AWS by making it more difficult to co-locate
  - Force adversaries to consume more resources thereby increasing the accuracy, rate, and speed with which threats maybe detected
  - Facilitate the creation of role-enabled security models

Research and Development Objectives

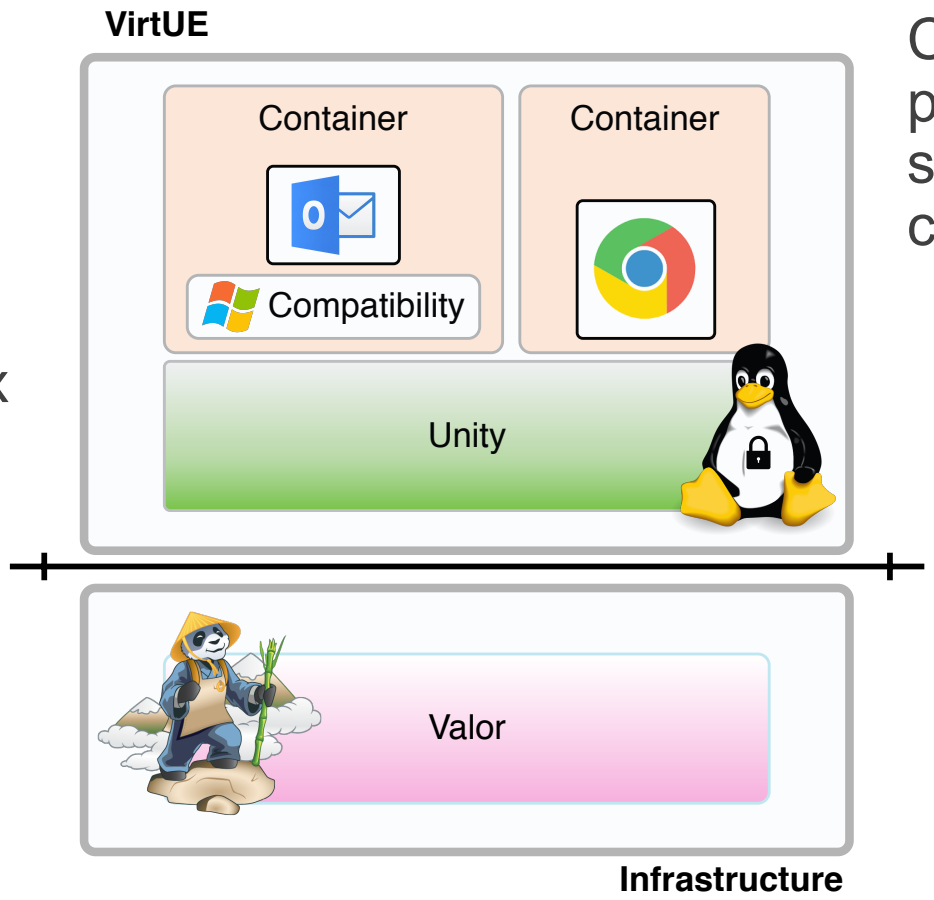
- **Deliver a defensible, role-based UCE capable of operating in AWS**
  - Limit adversary access to target while reducing overall attack surface
  - Support legacy applications
  - Make accessible from thin client, enable virtue-to-virtue sharing, and support single sign-on
  - Facilitate administration and scale to organizations of various sizes
  - Capture logging options across VirtUE software stack
  - Facilitate improved analytics and sensor control (Phase II)

Galahad Use Case



Galahad VirtUE

A small, hardened, de-privileged Linux OS VM



Containers for easy packaging and security configuration

A nested hypervisor to facilitate regular, recurring live migration of Unity VMs inside AWS

Galahad Security

- **Attack surface minimization of Galahad VirtUE**
  - Leverage Xen Kconfig for Valor
  - Build Unity with components favoring security such as granular application packaging, modular kernel components, and syscall filtering
  - Investigate the use of Unikernels
- **Stackable Linux Security Module (LSM)**
  - Deprivilege root and enable introspection for mandatory access control enforcement
- **Intelligent migration inside AWS using XenBlanket**
- **Challenges**
  - Minimizing the impact of migration on the end user experience
  - Controlling migration to enhance security

Galahad User Experience

- **Users interact with VirtUEs through Galahad's Canvas**
- **Approached modeled after Qubes OS**
  - Processes running in Unity and Canvas connected via secure network channel, e.g., leveraging Amazon's VPCs
  - GUI app displays flow through a modified X Windows system
  - Facilitates inter-VirtUE file / clipboard sharing
- **Controls for starting / stopping VirtUEs, i.e., opening / closing apps**
  - Menu allows users to select role and a VirtUE (app) therein
- **Challenges**
  - Single sign-on
  - Access to Microsoft resources

Galahad Lifecycle Management

- **User and administrator acceptance depends on sound lifecycle management**
- **Virtue Assembler to package apps with security-enhanced VirtUE**
  - Bundle configuration with Unity and container(s) before signing
  - Accurately report the contents of a VirtUE and its configuration
- **VirtUE Administrator to create, store, launch, and halt VirtUEs**
  - Uses a Control API to ensure VirtUEs communicate w/proper canvas
  - Verify the integrity of VirtUEs during launch, migration, and during administrator review
  - Retrieve status from executing VirtUEs, e.g., resource utilization
- **Challenges**
  - Maintaining situational awareness of VirtUEs

Galahad Sensing and Logging

- **Sensors throughout the Galahad VirtUE stack**
  - Instrumenting the compatibility layer (Wine)
  - Linux application instrumentation inside Unity
  - Hardware interaction monitoring and Unity introspection from Valor
  - Unity / Valor inter-layer validation
- **Actuators to enable response actions, e.g., closing a connection or terminating a process**
- **VirtUE administrator to configure sensors and provide "standing orders" for response**
- **Challenges**
  - Managing the control, configuration, and attribution of sensing and logging throughout migration
  - Scaling of sensing and logging to large deployments

Metrics

- **Program metrics mapped to requirements**
  - 11 Requirements evaluated via 19 metrics over three areas: functional, security, performance
  - Mapped each metric to a primary Galahad component

Star Lab Component	Requirement Synopsis	Metric Statement	Measure Type	Current Star Lab Progress
Unity	Virtues shall present themselves as atomic, largely immutable entities to other Virtues and external processes. They shall be simpler and more modular than current VDI solutions with a minimized attack surface	The number of exposed system calls in the exposed portion of a Virtue	count <= 200	
		The number of running processes in the exposed portion of a Virtue	count <= 20	
		The number of active/available services in the exposed portion of a Virtue	count <= 45	
		The number of communication paths that traverse a Virtue trust boundary	count <= 3	
		The number of credentials in the exposed portion of a Virtue	count <= 1	
		The success rate and performance cost of a Virtue with default protections in deterring		

Schedule and Milestones

- **Months 1-6**
  - Valor migrating VMs within AWS
  - Initial Unity protections / attack surface minimizations (< 10)
  - Initial VirtUE Administrator and Control API
  - Initial VirtUE Assembler prototype
  - Initial logging options (< 5)
- **Months 7-18**
  - Prototype and refine Canvas (increase # VirtUEs / resources)
  - Increase number of supported apps (Windows and Linux)
  - Increase number of logging options and Unity protections
  - Mature Control API and Assembler
  - Performance evaluation

Questions

Galahad Architecture