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

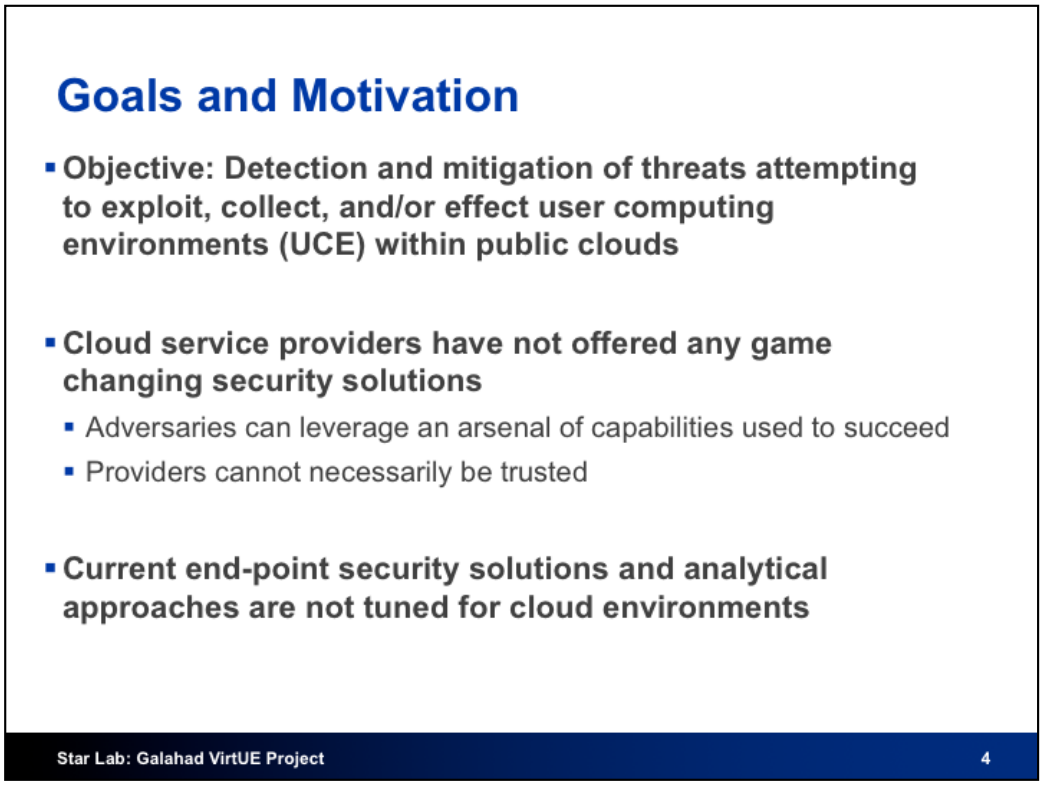
Raytheon
BBN Technologies

10 Moulton Street
Cambridge, MA 02138

Alex Jordan
Stephanie Gavin

Agenda

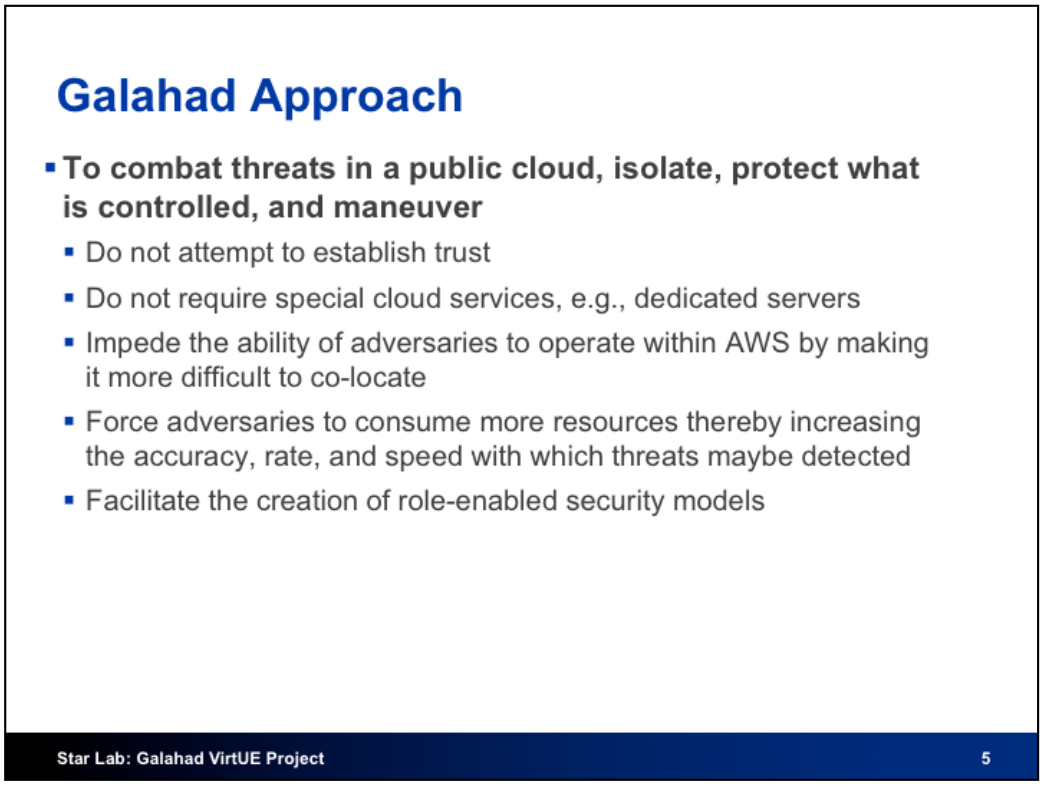
- Goals and Motivation
- Galahad Approach
- Research and Development Objectives
- Galahad Use Case
- Galahad VirtUE
- Galahad Security
- Galahad User Experience
- Galahad Lifecycle Management
- Galahad Sensing and Logging
- Metrics
- Schedule and Milestones



1. Leverage a public cloud while detecting and frustrating bad actors in the cloud with us. In view: 2&3.

2. Cloud providers simply provide VMs that resemble desktop, i.e., general purpose operating systems. They'll protect their resources but not their users'

3. End-point solutions designed for enterprise workstations that fulfill multiple roles. Unclear whether they can even be tuned given the current all-inclusive VM construct. Role-based VMs open the door to improved models of expected use.

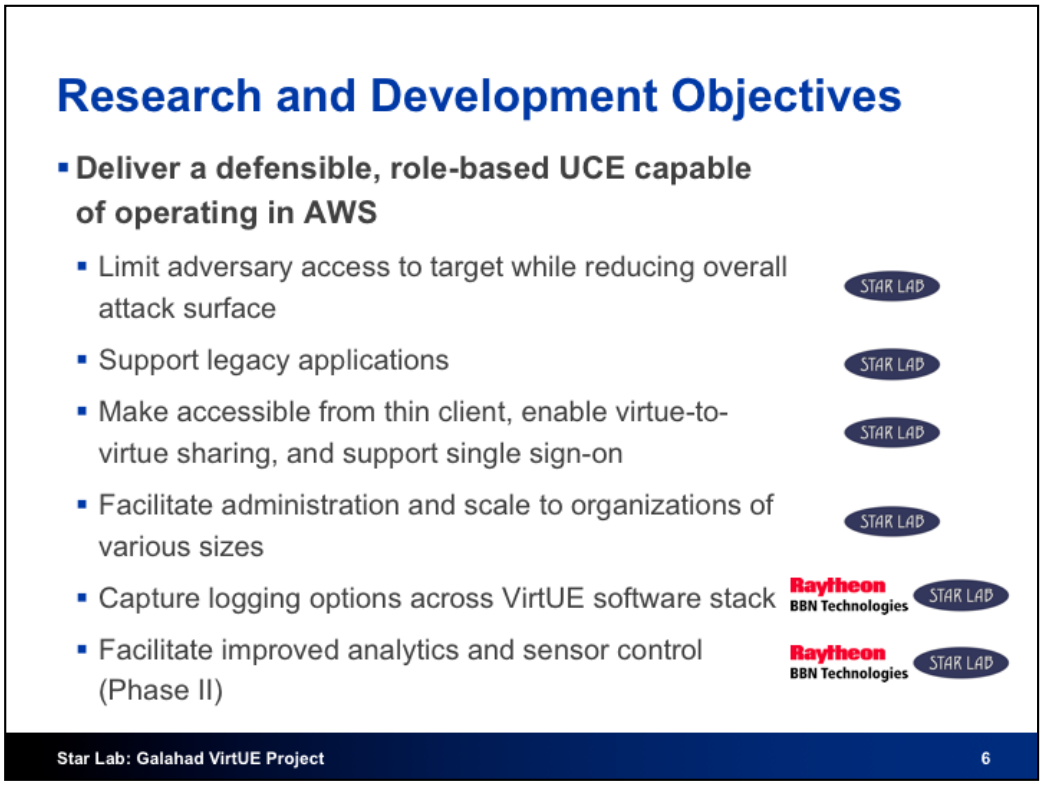


No positive control of traditional trust-establishing components

Don’t rely on “dedicated” (exclusively for our use) server

Make it more costly for adversary to find us in the cloud and more difficult to attack us

Role-based isolation, attack surface minimization practices, operating system (OS) and application hardening techniques, real-time sensing, and maneuver / deception



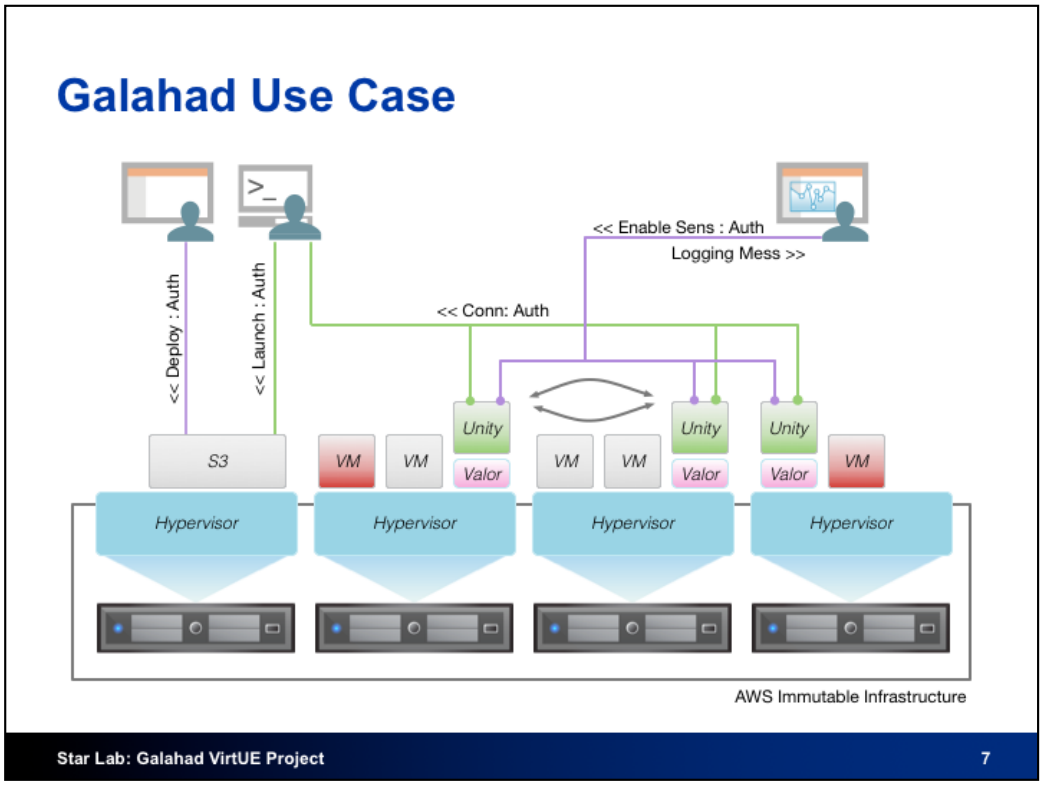
2. Your team's Phase 1 objectives, please also clearly indicate which members of your team are working on specific tasks.

Essentially we have security, usability, control, sensing, and analysis

Defensible -> isolation, hardening, minimization, and maneuver

Administration -> create VirtUEs, launch VirtUEs, destroy VirtUEs, halt VirtUEs, situational awareness of executing VirtUEs

BBN is focused on analytics, sensor and logging



Galahad Canvas

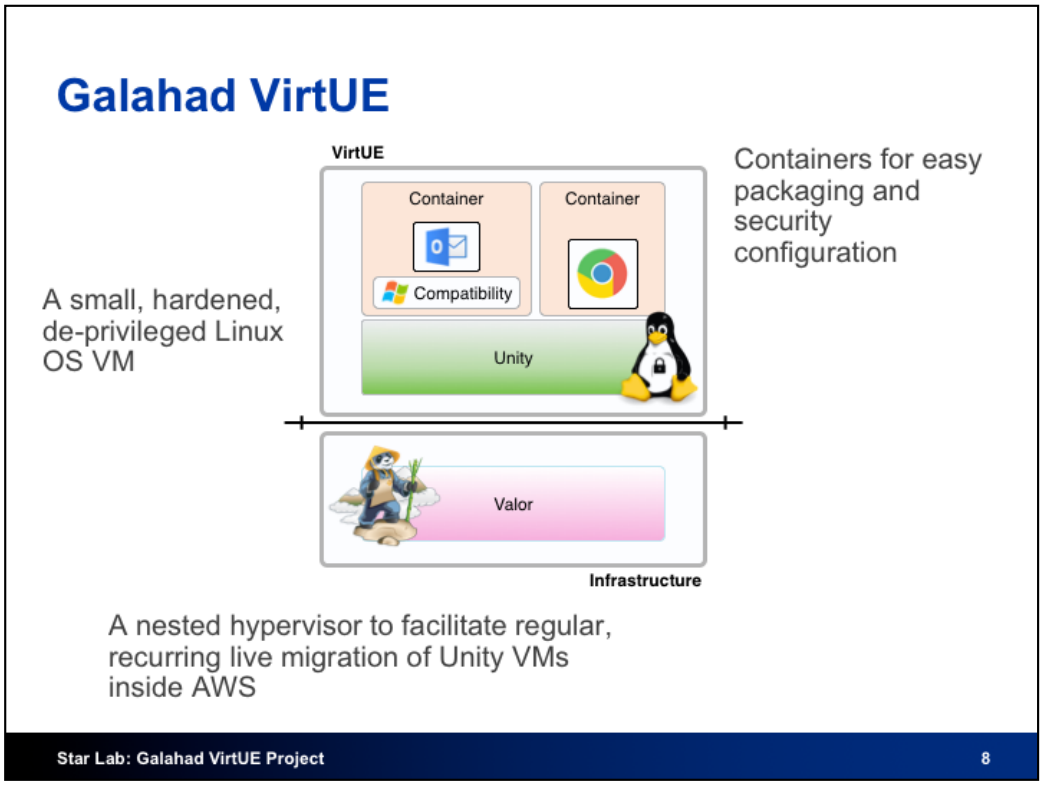
- User interface
- Resource sharing
- Single sign-on

Galahad Lifecycle Control

- VirtUE Assembler
- VirtUE Control API

Galahad Sensors and Loggers

- Configurable to allow for performance / logging tradeoffs
- Spans entire Galahad VirtUE software stack



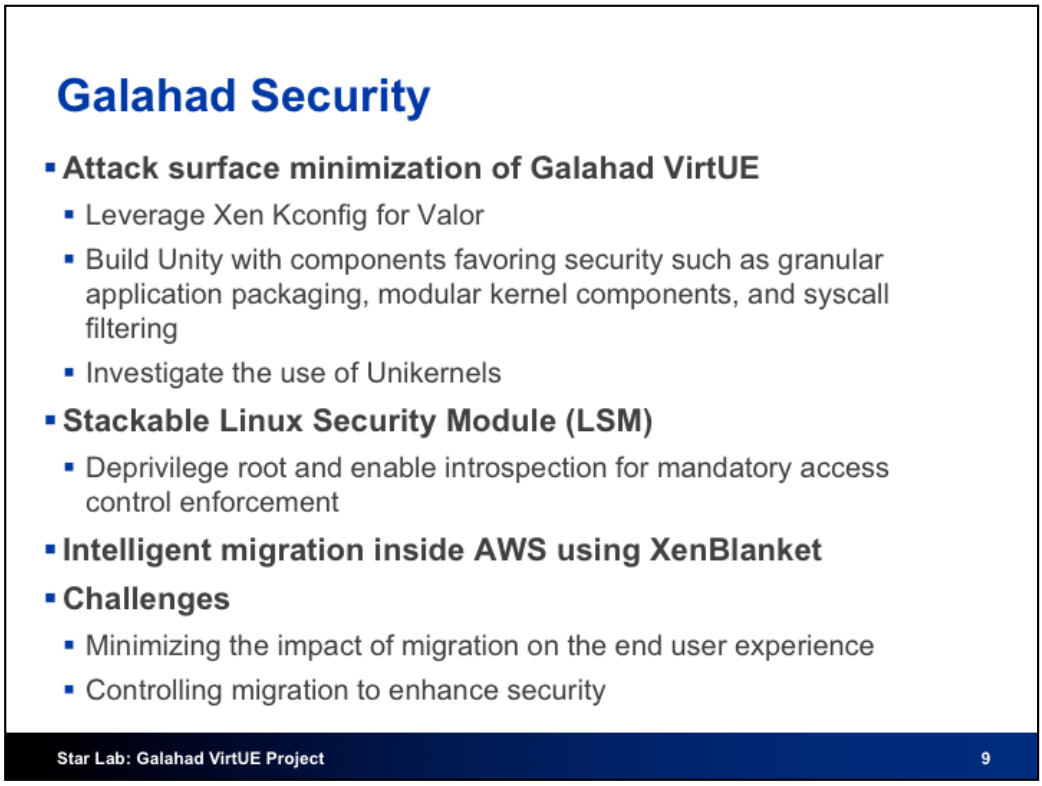
Valor: running directly on our EC2 instance; responsible for supporting migration (VirtUE load/unload) and some sensing

Unity: should it host multiple containers, or only one? We'll look into this.

Is Valor part of a VirtUE or just infrastructure. More discussion is needed. What does it mean if Valor is part of the infrastructure versus part of the VirtUE? Our architecture indicates we might have just Valors available in AWS (for faster migration). Will the eval team assess lone Valors?

Challenges:

- Once we migrate a Unity with its containers, how does it plug back into Valor for security monitoring purposes?
- Will Valor take care of wiping memory for a Unity once it migrates or shuts down?

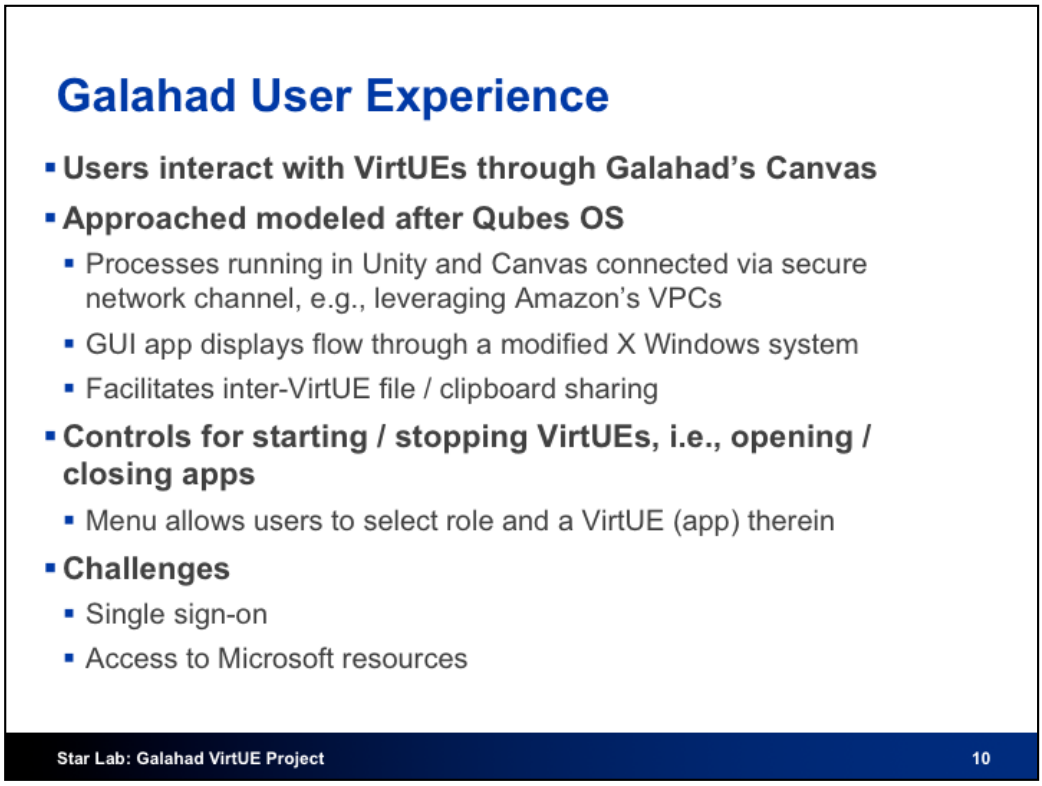


Star Lab has extensive experience with Unikernels. While it isn't likely administrators will want build unikernels, it might be possible to build single app VirtUEs (for a very specific role) and place them in the VirtUE repository. Thus, administrators never have to build them.

There was an interesting study in 2015 where this kind of forced migration was used with spot servers in order to minimize cloud costs. Paper was called "Cutting the cost of hosting online services using cloud spot markets"

Smartly migrating means monitoring for inactivity or when we detect co-location, i.e., when we detect another VirtUE.

How do we know we're really migrating to another piece of hardware?



how you will execute your technical approach and

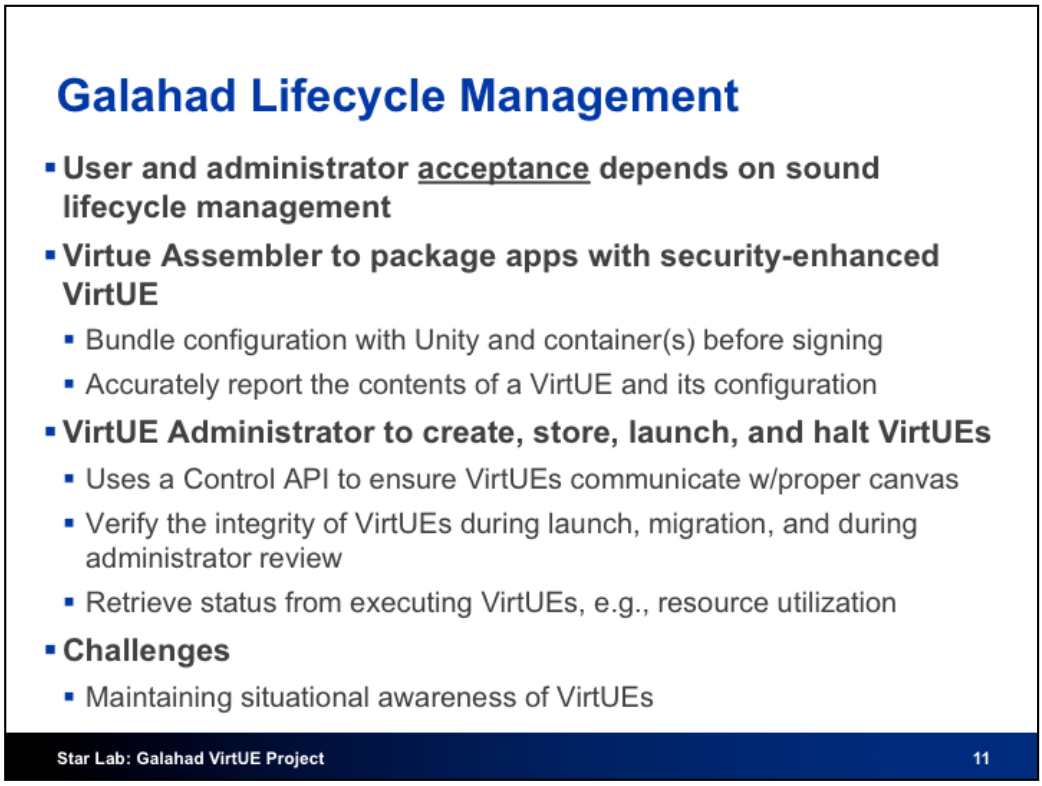
How you envision creating the user interface for eventual VirtUE users.

What technologies will you leverage and how will you ensure users will experience convenience while not violating the basic security tenants of the VirtUE program.

Q: will need to interact with Microsoft DS for authentication and to get access to network resources. How can your Virtue solution work with Microsoft DS while minimizing well-known credential stealing vulnerabilities like pass the ticket and hash that have plagued Windows solutions forever.

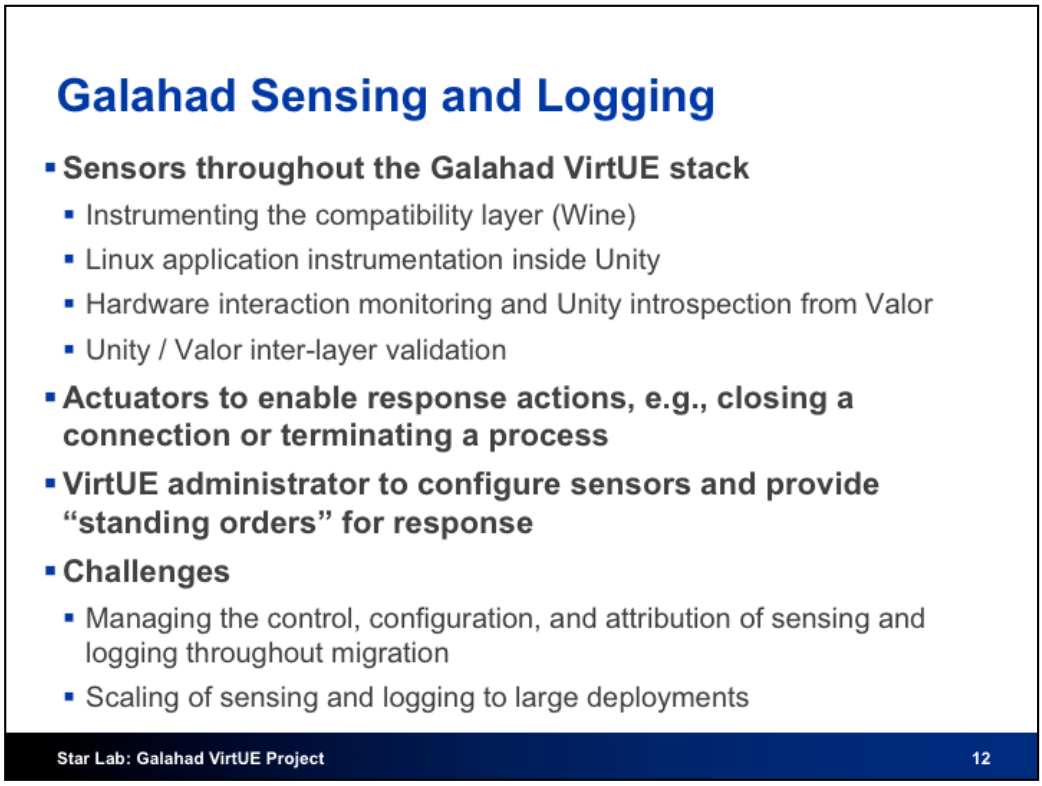
A: single sign-on capabilities: Use existing, open-source SSO solution such as FreeIPA, WSO2, etc. Microsoft LSASS server in cloud for MS resource access? Windows 10 Enterprise or Server 2016 or higher, where Windows Defender Credential Guard was introduced to mitigate against these

attacks. We won’t support this MS resource as a VirtUE, so where should it reside?



Proper canvas is facilitated via a shared key unique to a VirtUE's Unity

Status information will be used to educate users / administrators on cloud untilization (potentially help control costs)



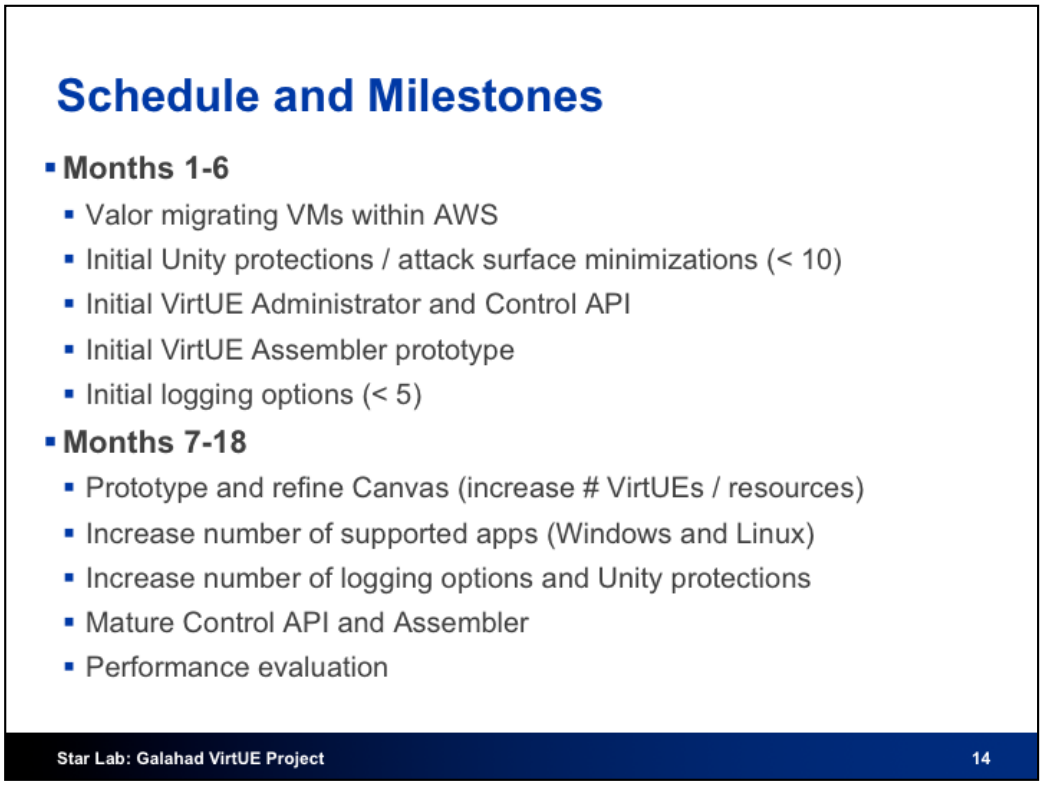
Having Unity be just Linux, i.e., not supporting multiple OSes, makes it easier to create and maintain a sensor and logging capability.

In phase II, the control API can be enhanced for dynamic reconfiguration by Phase II automated analytical systems

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		



What you hope to accomplish and when.

Valor is high-risk, foundational item; must start early

Unity and assembler low-risk (we've done this kind of thing before)

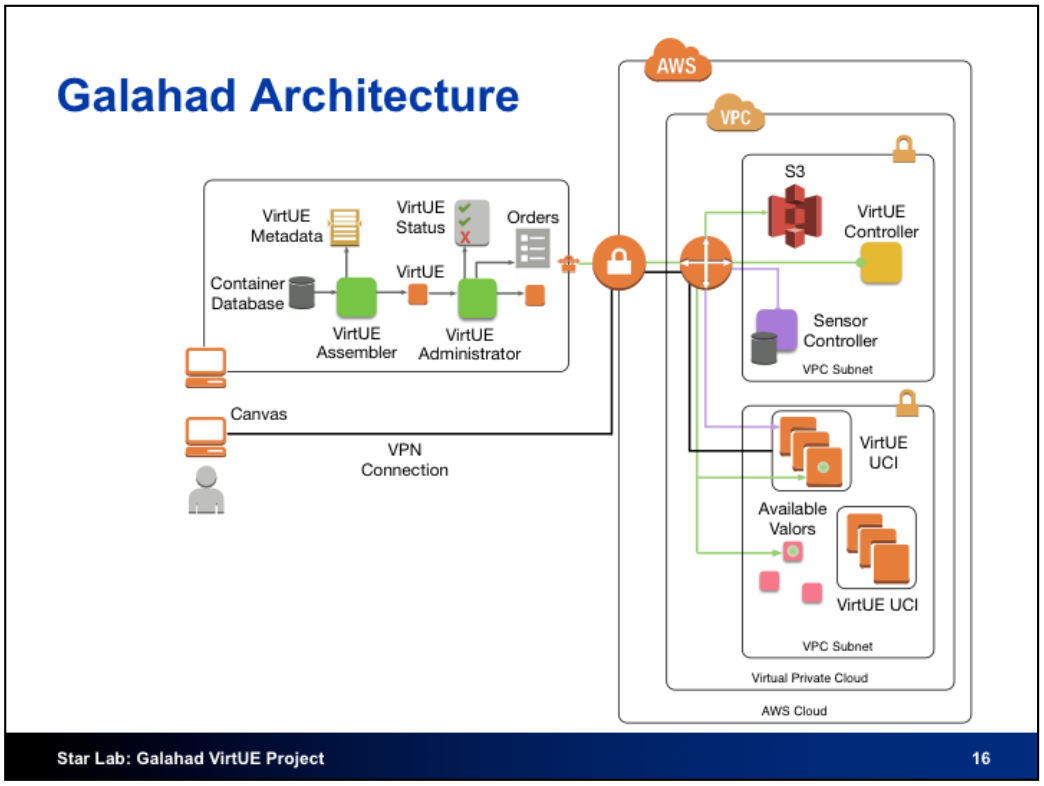
Contol API requires a good understanding of AWS; must start early

Saving Canvas development for a bit later in the program (got to build the foundational elements first)

Numbers in parentheses are quantities specified in the requirements

A more detailed schedule available in GitHub.

Questions



Administrators use top left-hand box. They create VirtUEs using a **container database** and the **assembler** (the assembler will be where the logic exists to achieve attack surface minimization, security hardening, etc.). The assembler includes other packaging features such as security keys, manifests, etc. and can interrogate a local VirtUE image. Once created, the VirtUE Administrator pushes the VirtUE to AWS storage (S3). The VirtUE Administrator can also:

- View the status of individual or groups of VirtUEs
- Push orders to VirtUEs through the VirtUE Controller, e.g., force a launch / halt, migrate a VirtUE, or change the configuration of migration behavior.
- Push **active response [mitigation]** orders through the Sensor Controller, e.g., force some sensors to halt, reconfigure how sensors store data, etc.

The user interacts with their UCI via Canvas running in a thin client. Using the same VirtUE control library found in VirtUE Administrator, users will select roles and start / stop VirtUEs. We need to discuss **SOO**: we should use **SAML**, but where do customer resources like authentication server (e.g. **AD**) live? Which network(s)/VPC(s)? Our design should cooperate with expected deployment.

Within AWS are two subnets, maybe more if research suggests better isolation can be achieved. In one subnet are UCI infrastructure pieces to include storage for VirtUEs (S3), VirtUE controller, Sensor Controller, and Microsoft Infrastructure for authentication to Microsoft resources. The VirtUE Controller will serve as a C2 node for all VirtUEs. The Sensor Controller will focus just the sensing and logging taking place within VirtUEs.