BITS Pilani

Pilani Campus

Cloud, IoT and Enterprise Security

Nishit Narang
WILPD-CSIS
(nishit.narang@pilani.bits-pilani.ac.in)

BITS Pilani

Pilani Campus

<SSCSZG570 , Cloud, IoT and Enterprise Security>

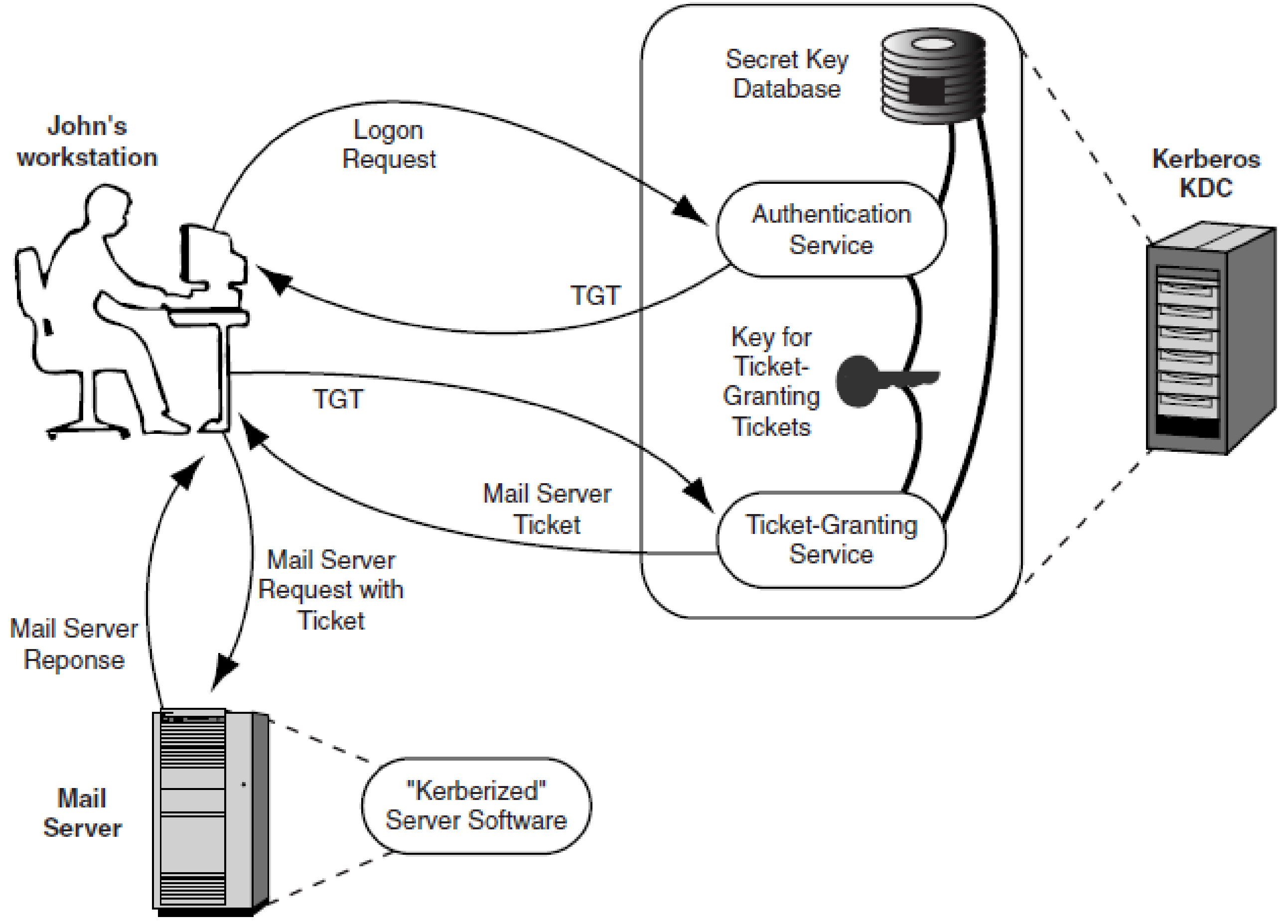
Lecture No. 14: Cloud Security

Identity and Access Management (IAM) - Continued

- **Source Disclaimer**: Content for some of the slides is from the course Textbook:
  - *Ronald L. Krutz, Russell Dean Vines, Cloud Security: A Comprehensive Guide to Secure Cloud Computing, John Wiley & Sons, 2010*
- Some of the slides are taken from Microsoft Educator Learn Material (Microsoft Azure Security Technologies)
- Material for some of the other slides is from following book:
  - *Authentication: From Passwords to Public Keys, by Richard E. Smith*

RECAP: Kerberos

- Kerberos KDC with 2-step ticket granting process



innovate achieve lead

BITS Pilani

Pilani Campus

IAM Protocols and Standards for Cloud Services

IAM for Cloud Services

- Lot of Enterprises are embracing and adopting Cloud Services
- Multiple IAM Standards and Protocols help organizations implement efficient User Access Management practices in the cloud
  - Offer a SSO experience and avoid duplication of identity, attributes and/or credentials → SAML
  - Support automatic (De-)Provisioning of User Accounts → SPML
  - Enforce privilege and entitlement-based Access Control → XACML
  - Integrate applications / cloud services without sharing credentials → OAuth 2.0
    - E.g. Authorize cloud service X to access my data in cloud service Y without disclosing my credentials to X

SAML

- **Security Assertion Markup Language** (**SAML,** pronounced *sam-el*)
  - An XML-based, open-standard data format for exchanging authentication and authorization data between parties
  - In particular, used between an **identity provider** (IdP) and a **service provider** (SP)
- SAML is a product of the OASIS* Security Services Technical Committee

***Organization for the Advancement of Structured Information Standards** (**OASIS**) is a global nonprofit consortium that works on the development, convergence, and adoption of standards for security, Internet of Things, energy, content technologies, emergency management, and other areas.

SAML Principles

- SAML Roles: the specification defines three roles:
  - the principal (typically a user),
  - the Identity provider (IdP), and
  - the service provider (SP)
- SAML Use Case:
  1. Principal requests a service from the service provider
  2. Service provider requests and obtains an identity assertion from the identity provider
  3. On the basis of this assertion, the service provider can make an access control decision
     - i.e. it can decide whether to perform some service for the connected principal
  4. Before delivering the identity assertion to the SP, the IdP may request some information from the principal – such as a user name and password – in order to authenticate the principal

SAML does not specify the method of authentication at the identity provider; it may use a username and password, or other form of authentication, including multi-factor authentication

One identity provider may provide SAML assertions to many service providers. Similarly, one SP may rely on and trust assertions from many independent IdPs.

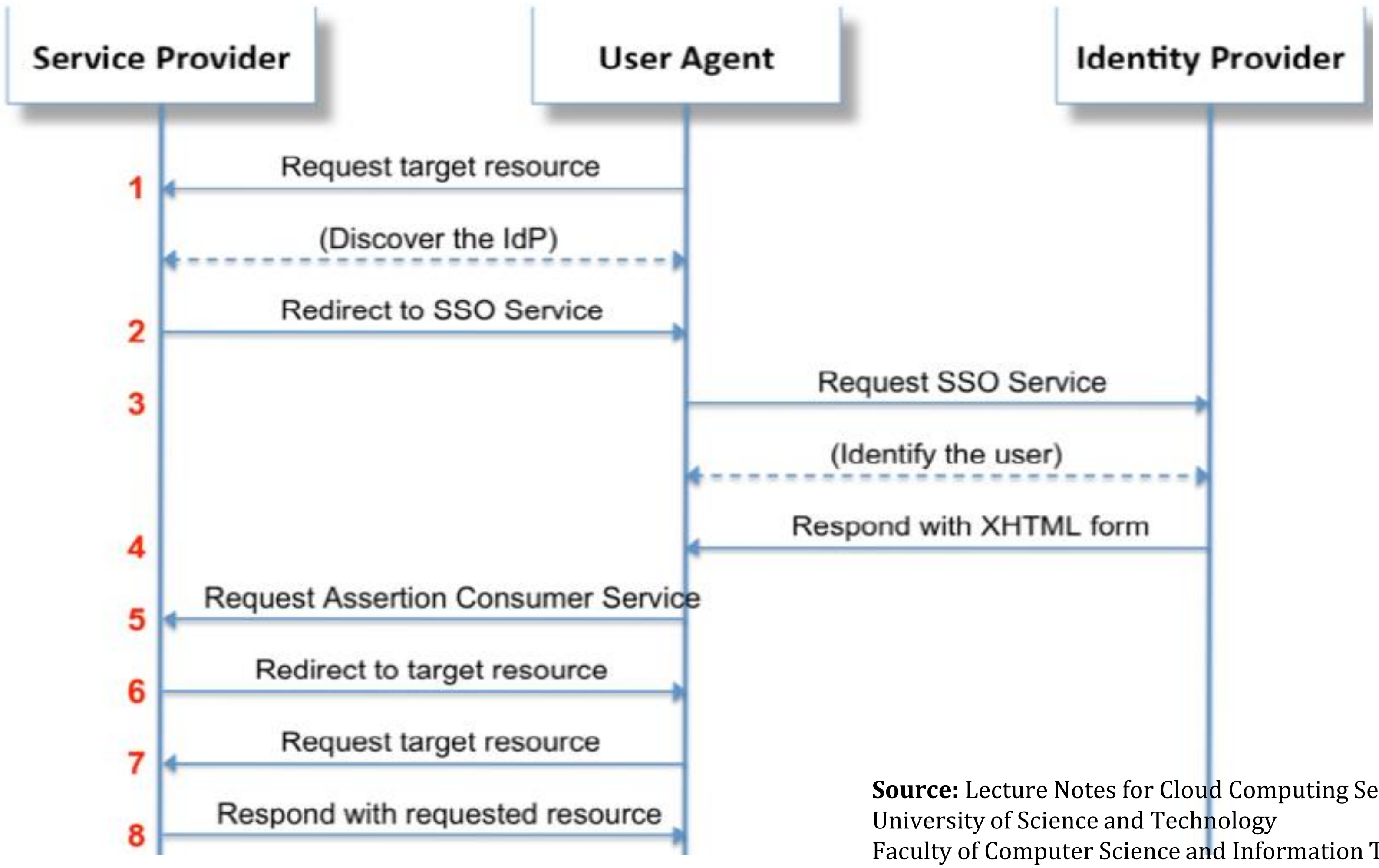
Web Browser SSO using SAML

- The primary SAML use case is the Web Browser Single Sign-On (SSO), where a user using a user agent (usually a web browser) requests a web resource protected by a SAML service provider

- What is SSO?
  - Single sign-on (SSO) is a property of access control of multiple related, yet independent, software systems.
  - With this property, a user logs in with a single ID and password to gain access to a connected system or systems without using different usernames or passwords, or in some configurations seamlessly sign on at each system.
  - This is typically accomplished using the Lightweight Directory Access Protocol (LDAP) and stored LDAP databases on (directory) servers.

Source: Lecture Notes for Cloud Computing Security
University of Science and Technology
Faculty of Computer Science and Information Technology

Message Flow



Source: Lecture Notes for Cloud Computing Security
University of Science and Technology
Faculty of Computer Science and Information Technology

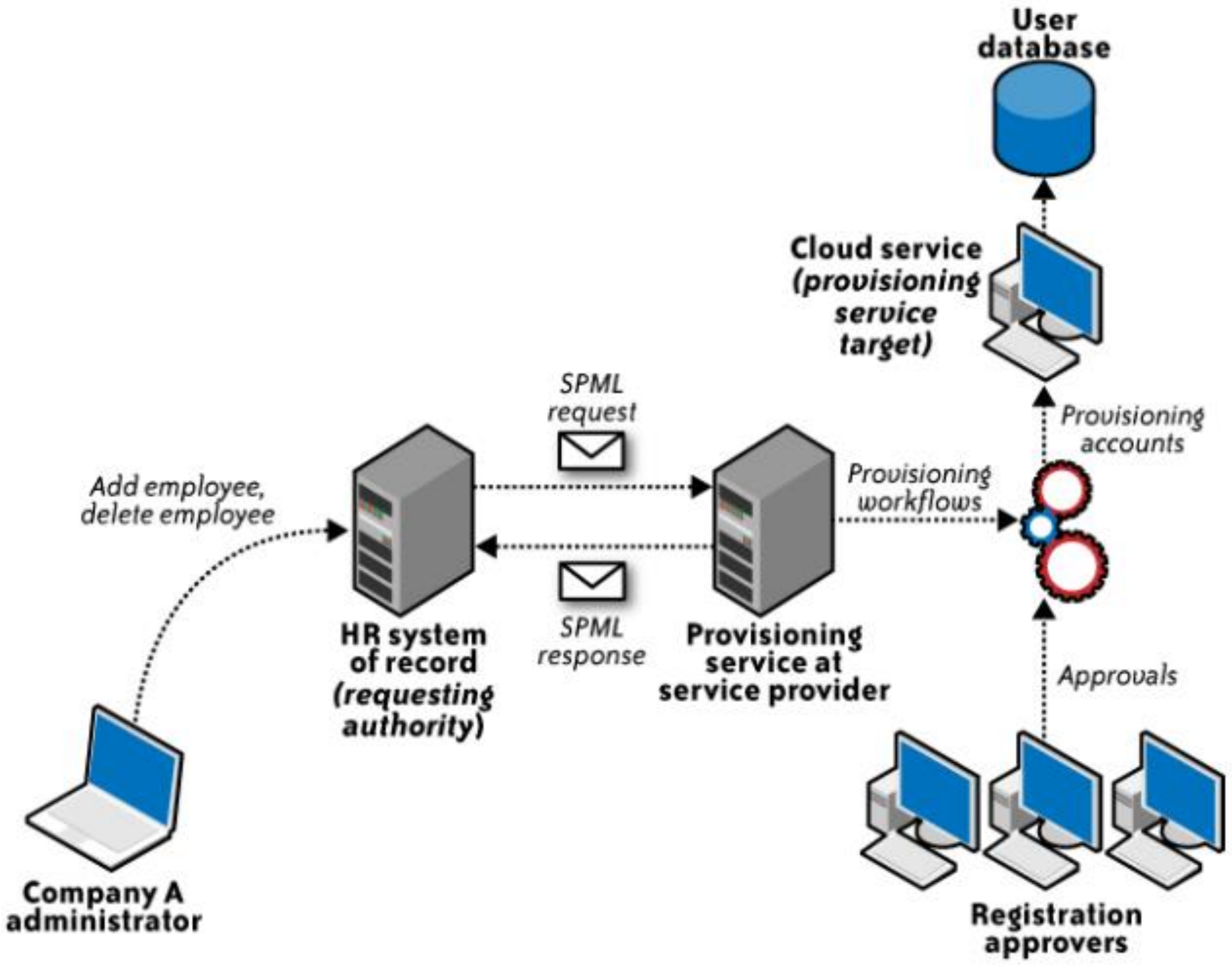
SPML

- Service Provisioning Markup Language
  - XML-based framework developed by OASIS
  - Used to provision user accounts and profiles with the cloud service
  - Enables "just-in-time provisioning" to create accounts for new users in real time (instead of pre-registering user accounts)
- SPML helps achieve the below twin objectives:
  - Automate IT tasks for user provisioning
  - Enables interoperability between different provisioning systems using standard SPML interfaces

SPML Principles

- SPML Roles:
  - Requesting Authority (RA)
    - The client in SPML
  - Provisioning Service Point (PSP)
    - listens to the request from the RA, processes it, and returns a response to the RA
  - Provisioning Service Target (PST)
    - the actual resource on which the action is taken
    - E.g. an LDAP directory that stores an organization's user accounts, or a ticketing system used to issue access tickets

SPML Message Flow


User database
Cloud service (provisioning service target)
SPML request
Provisioning accounts
Add employee, delete employee
Provisioning workflows
SPML response
HR system of record (requesting authority)
Provisioning service at service provider
Approvals
Company A administrator
Registration approvers

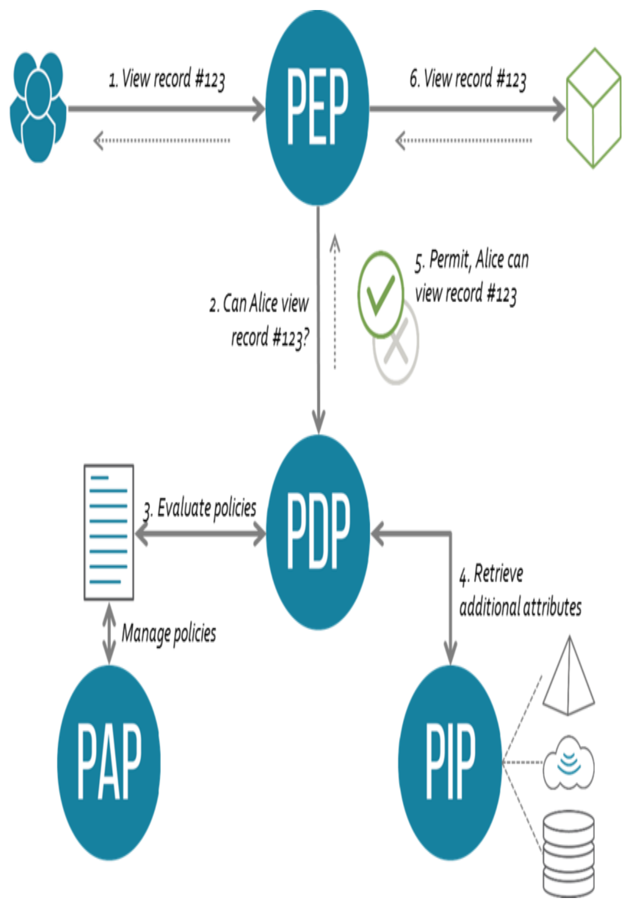
Source: Cloud Security and Privacy, by Tim Mather, Subra and Latif

XACML

- eXensible Access Control Markup Language
  - an OASIS, general-purpose, XML-based standard
  - defines a declarative fine-grained, attribute-based access control policy language and architecture
  - Also defines a processing model describing how to evaluate access requests according to the rules defined in policies
- XACML is primarily an attribute-based access control system (ABAC), also known as a policy-based access control (PBAC) system
  - attributes associated with a user or resource are inputs into the decision of whether a given user may access a given resource in a particular way
  - Role-based access control (RBAC) can also be implemented in XACML as a specialization of ABAC

Source: XACML - Wikipedia

XACML Architecture


1. View record #123
PEP
6. View record #123
2. Can Alice view record #123?
5. Permit, Alice can view record #123
3. Evaluate policies
PDP
4. Retrieve additional attributes
Manage policies
PAP
PIP

Source: XACML - Wikipedia

Abbr.	Term	Description
PAP	Policy Administration Point	Point which manages access authorization policies
PDP	Policy Decision Point	Point which evaluates access requests against authorization policies before issuing access decisions
PEP	Policy Enforcement Point	Point which intercepts user's access request to a resource, makes a decision request to the PDP to obtain the access decision (i.e. access to the resource is approved or rejected), and acts on the received decision
PIP	Policy Information Point	The system entity that acts as a source of attribute values (i.e. a resource, subject, environment)
PRP	Policy Retrieval Point	Point where the XACML access authorization policies are stored, typically a database or the filesystem.

OAuth (2.0)

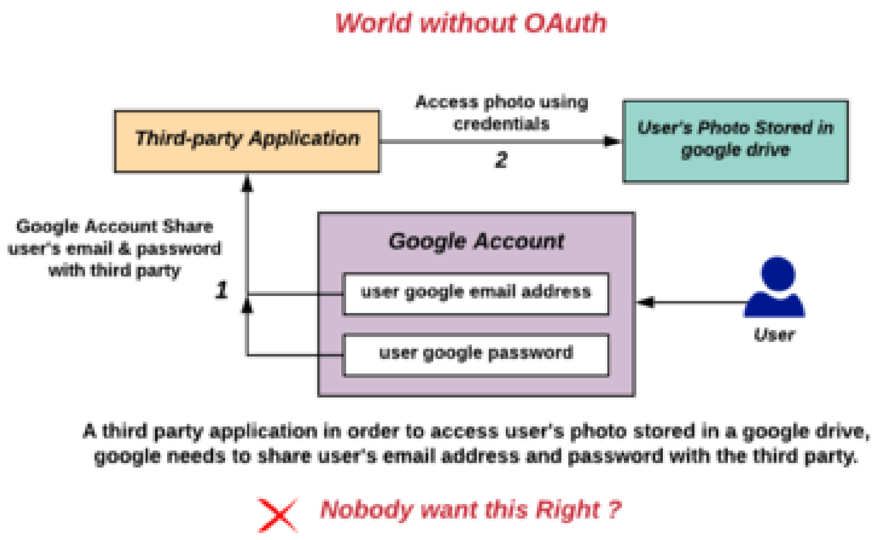
Auth in OAuth could imply ***Authentication***, but means ***Authorization***!!

- **OAuth** (***O****pen* ***Auth****orization*) is an open standard for access delegation
  - commonly used as a way for Internet users to grant websites or applications access to their information on other websites but without giving them the passwords
  - This mechanism is used by companies such as Amazon, Google, Facebook, Microsoft, and Twitter to permit the users to share information about their accounts with third-party applications or websites
- OAuth enables following use cases
  - delegated access control:
    - I, the user, delegate another user or service access to the resource I own. For instance via OAuth, I grant Twitter (the service) the ability to post on my Facebook wall (the resource).
  - handling the password anti-pattern:
    - Whenever you want to integrate 2 services together, in a traditional, legacy model you have to provide service B with your user credentials on service A so that service B can pretend to be you with Service A. This has many risks of course. Using OAuth eliminates the issues with these patterns and lets the user control what service B can do on behalf of the user with service A.

Source: XACML - Wikipedia

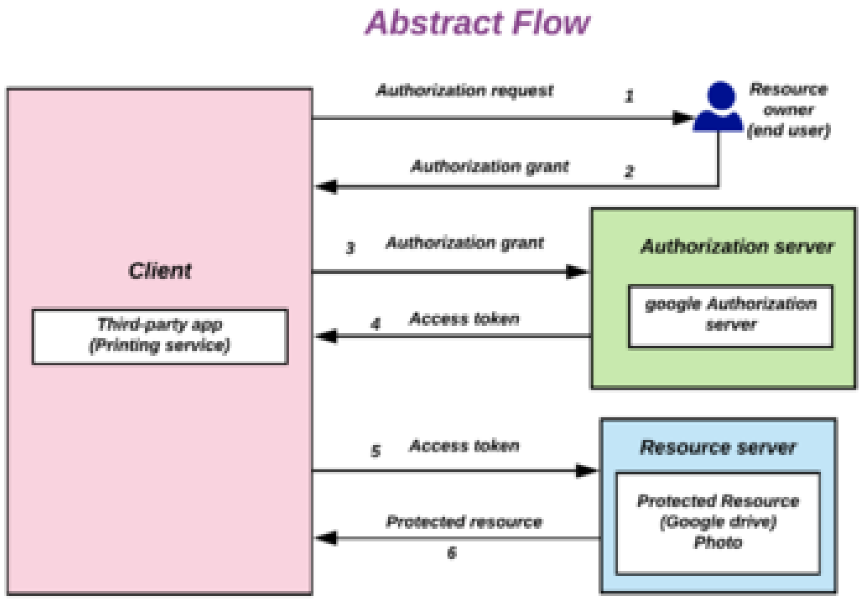
Source: OAuth - Wikipedia

OAuth (2.0) Use Case


World without OAuth
Third-party Application
Access photo using credentials
2
User's Photo Stored in google drive
Google Account Share user's email & password with third party
1
Google Account
user google email address
user google password
User

A third party application in order to access user's photo stored in a google drive, google needs to share user's email address and password with the third party.

❌ *Nobody want this Right ?*



Source: OAuth - Wikipedia

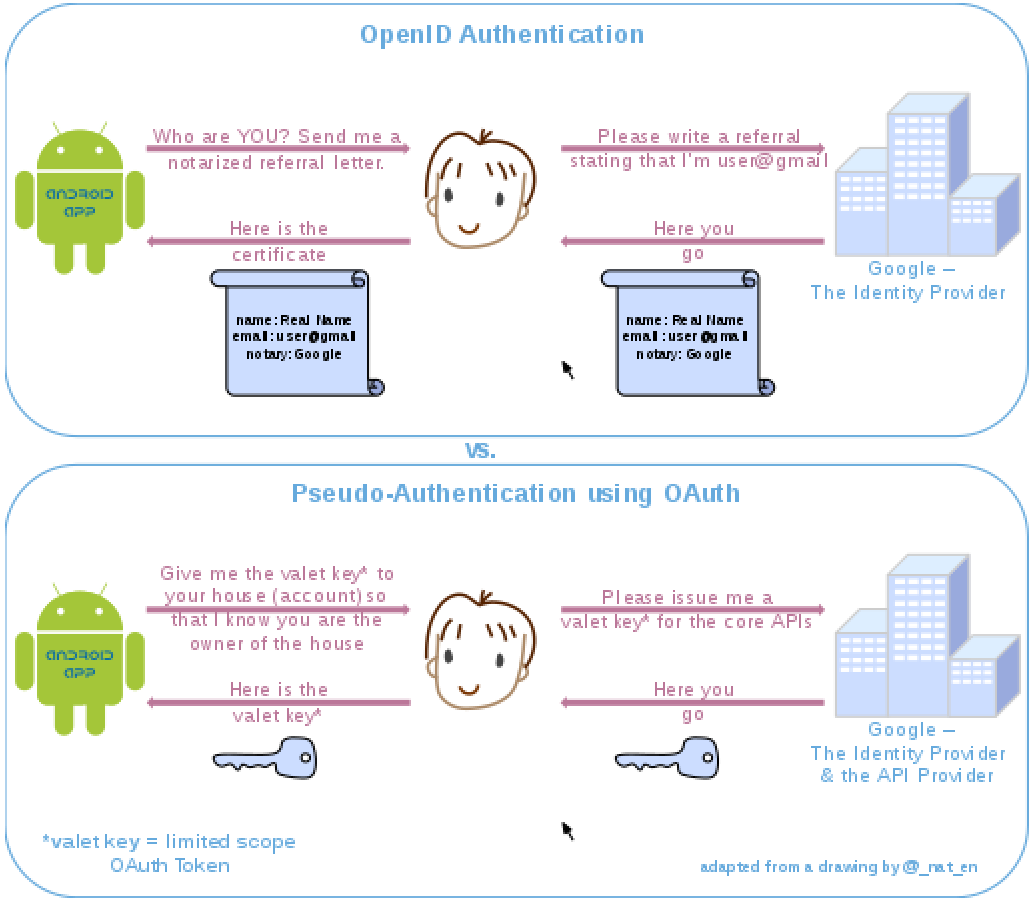
OAuth Vs SAML Vs Open ID

- OpenID Connect is built on the OAuth 2.0 protocol and uses an additional JSON Web Token (JWT), called an ID token
  - standardizes areas that OAuth 2.0 leaves up to choice, such as scopes and endpoint discovery
  - It is specifically focused on user authentication and is widely used to enable user logins on consumer websites and mobile apps
- SAML is independent of OAuth, relying on an exchange of messages to authenticate in XML SAML format, as opposed to JWT
  - It is more commonly used to help enterprise users sign in to multiple applications using a single login

Source: What's the Difference Between OAuth, OpenID Connect, and SAML? | Okta

OAuth Vs Open ID

- OpenID is specifically designed as an authentication protocol and OAuth for authorization


OpenID Authentication
android app
Who are YOU? Send me a notarized referral letter.
Please write a referral stating that I'm user@gmail
Here is the certificate
Here you go
name: Real Name
email: user@gmail
notary: Google
Google – The Identity Provider
vs.
Pseudo-Authentication using OAuth
Give me the valet key* to your house (account) so that I know you are the owner of the house
Please issue me a valet key* for the core APIs
Here is the valet key*
Here you go
Google – The Identity Provider & the API Provider
*valet key = limited scope OAuth Token
adapted from a drawing by @_nat_en

[Source: OAuth - Wikipedia]

innovate achieve lead

BITS Pilani

Pilani Campus

Case Study: Identity and Access in Microsoft Azure

Microsoft
ʌerPoint Presentat

BITS Pilani

Pilani Campus

<SSCSZG570 , Cloud, IoT and Enterprise Security>

Lecture No. 15: Cloud Security

Infrastructure Security and Application Security

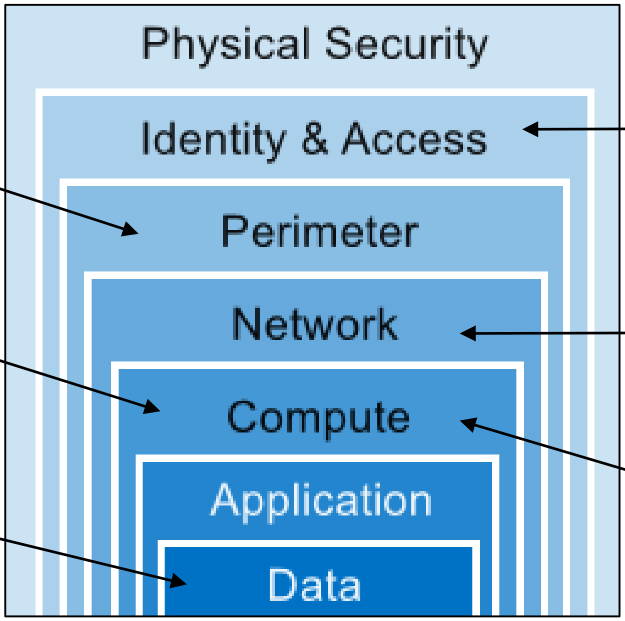
- **Source Disclaimer**: Content for some of the slides is from the course Textbook:
  - *Ronald L. Krutz, Russell Dean Vines, Cloud Security: A Comprehensive Guide to Secure Cloud Computing, John Wiley & Sons, 2010*
- Some of the slides are taken from Microsoft Educator Learn Material (Microsoft Azure Security Technologies)
- Material for some of the other slides is from Kubernetes public documentation:
  - *https://kubernetes.io/docs/concepts/security/*

Defense in Depth

DDoS
Azure Firewall

Host Security

Microsoft Defender for Cloud for SQL -and- Microsoft Defender for Cloud for Storage -and- Azure Information Protection



Privileged Identity Management
Conditional Access

Network Security Groups
Application Security Groups

Network Micro-Segmentation

Container Security

Provide a layered approach and multiple levels of protection

Infrastructure Security

- Perimeter Security to protect your “virtual network” via combination of:
  - DDoS mitigation solutions
  - Firewall services (Network Firewalls and Web Application Firewalls)
  - VPN services
- Network Security
  - Network segmentation (e.g. hub and spoke vnets, Network Service Groups)
    - Use of security rules to allow or deny network traffic
    - Can be associated to a subnet or a network interface
- Host Security
  - End-point protection services (e.g. anti-malware)
  - Disk encryption
  - Update Management
- Container Security
  - Container Registry with Signed Container Images
  - Authenticated access/ RBAC to Registry
  - Network ACL for access of control plane APIs of container management solution (e.g. Kubernetes)

Virtualization Security Management

- The important thing to remember from a security perspective is that there is a more significant impact when a host OS with user applications and interfaces is running outside of a VM at a level lower than the other VMs (i.e., a Type 2 architecture)
- Because of its architecture, the Type 2 environment increases the potential risk of attacks
- For example, a laptop running VMware with a Linux VM on a Windows XP system inherits the attack surface of both OSs, plus the virtualization code (VMM)

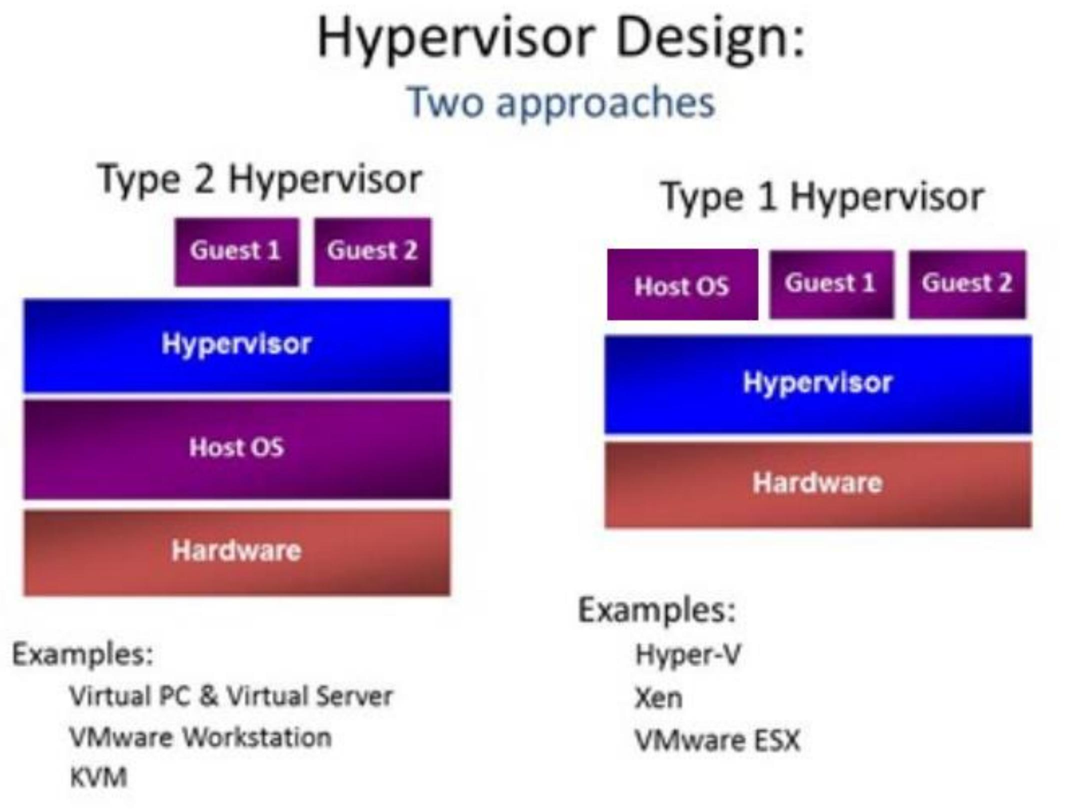


Image Source: https://www.tenforums.com/virtualization/119469-hypervisor-type-1-type-2-a.html

Hypervisor Risks

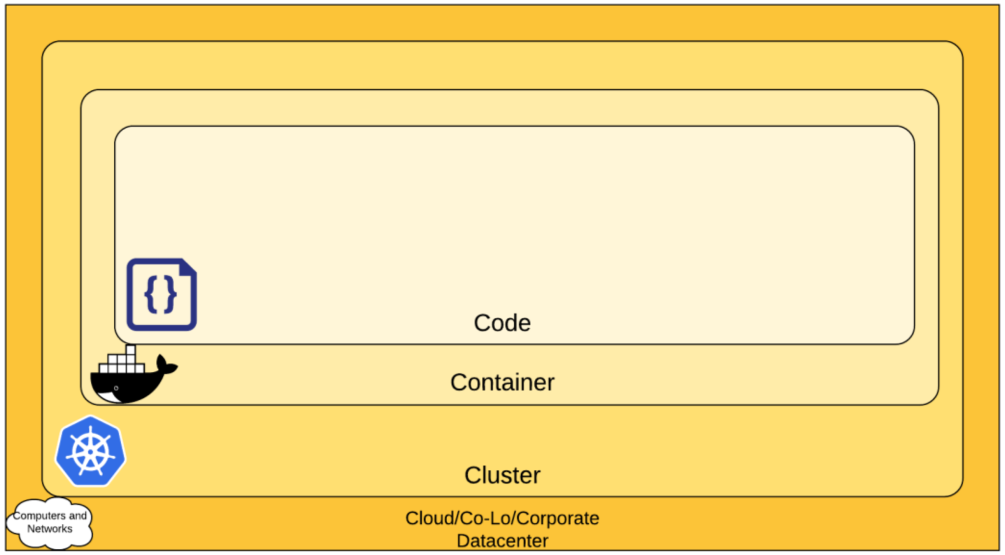
- The ability of the hypervisor to provide the necessary isolation during an attack greatly determines how well the virtual machines can survive risks
- Ideally, software code operating within a defined VM would not be able to communicate or affect code running either on the physical host itself or within a different VM;
- However, several issues, such as bugs in the software, or limitations to the virtualization implementation, may put this isolation at risk
- Major vulnerabilities inherent in the hypervisor consist of rogue hypervisor rootkits, external modification to the hypervisor, and VM escape*

* refers to the attacker's ability to execute arbitrary code on the VM's physical host, by "escaping" the hypervisor

VM Security Practices

- Hardening the Host OS and limiting physical access to the host
- Hardening the VM
- Hardening the Hypervisor
- Implement only one primary function per VM
- Use Unique NICs for Sensitive VMs
- Secure VM Remote Access

4Cs of Cloud Native Security



Containers are packages of software that contain all of the necessary elements to run in any environment. Contains all libraries and dependencies required for the application. OS is virtualized. User mode of the OS is included with the containerized application.

The Code layer benefits from strong base (Cloud, Cluster, Container) security layers. You cannot safeguard against poor security standards in the base layers by addressing security at the Code level

Cloud Security

- The Cloud Service Provider (CSP) offers Infrastructure Security across all shared-responsibility models
- There are a variety of compliance frameworks that can serve as a roadmap for security of the cloud environment
- These standards are designed to assure consistency and security for consumers
  - ISO/IEC 27017 and ISO/IEC 27018 are frameworks designed for cloud computing providers for the protection of their clients
    - The first focuses primarily on security controls, the second more on privacy concerns
  - The Service Organization Control (SOC) is a standard of compliance that has three types of certification, named SOC 1, SOC 2 and SOC 3
    - SOC 1 is primarily meant for banks, investment firms and other such companies that house financial data, and SOC 2 is for non-financial companies that house or process data, which could happen to be financial or otherwise
    - It's this latter certification (SOC 2) that software and cloud providers often use to verify their technology controls and processes
    - Obtaining a SOC 2 certification is a rigorous process, since a third-party CPA firm comes to the vendor's datacenter site and performs an assessment of their availability and security stance

Cloud Security (2)

- Security documentation of some of the popular cloud service providers:

IaaS Provider	**Link**
Alibaba Cloud	https://www.alibabacloud.com/trust-center
Amazon Web Services	https://aws.amazon.com/security/
Google Cloud Platform	https://cloud.google.com/security/
IBM Cloud	https://www.ibm.com/cloud/security
Microsoft Azure	https://docs.microsoft.com/en-us/azure/security/azure-security
Oracle Cloud Infrastructure	https://www.oracle.com/security/
VMWare VSphere	https://www.vmware.com/security/hardening-guides.html

Source: https://kubernetes.io/docs/concepts/security/overview/

Infrastructure Security for K8s Cluster

Area of Concern for Kubernetes Infrastructure	**Recommendation**
Network access to API Server (Control plane)	All access to the Kubernetes control plane is not allowed publicly on the internet and is controlled by network access control lists restricted to the set of IP addresses needed to administer the cluster.
Network access to Nodes (nodes)	Nodes should be configured to *only* accept connections (via network access control lists) from the control plane on the specified ports, and accept connections for services in Kubernetes of type NodePort and LoadBalancer. If possible, these nodes should not be exposed on the public internet entirely.
Kubernetes access to Cloud Provider API	Each cloud provider needs to grant a different set of permissions to the Kubernetes control plane and nodes. It is best to provide the cluster with cloud provider access that follows the principle of least privilege for the resources it needs to administer. The Kops documentation provides information about IAM policies and roles.
Access to etcd	Access to etcd (the datastore of Kubernetes) should be limited to the control plane only. Depending on your configuration, you should attempt to use etcd over TLS. More information can be found in the etcd documentation.
etcd Encryption	Wherever possible it's a good practice to encrypt all storage at rest, and since etcd holds the state of the entire cluster (including Secrets) its disk should especially be encrypted at rest

Source: https://kubernetes.io/docs/concepts/security/overview/

Cluster Security

- Protecting a cluster from accidental or malicious access can be done via:
  - Passing all API calls via Authentication and Authorization
  - Encrypting all API communication in the cluster is with TLS
- Controlling the runtime capabilities of a workload can be done via:
  - Defining Resource quota limits to limit the amount of CPU, memory, or persistent disk a namespace can allocate, and also control how many pods, services, or volumes exist in each namespace
  - Control the privileges associated with containers using the Pod security policies
- Restricting network access
  - Application authors can restrict which pods in other namespaces may access pods and ports within their namespaces

Container Security

Area of Concern for Containers	**Recommendation**
Container Vulnerability Scanning and OS Dependency Security	As part of an image build step, you should scan your containers for known vulnerabilities.
Image Signing and Enforcement	Sign container images to maintain a system of trust for the content of your containers.
Disallow privileged users	When constructing containers, create users inside of the containers that have the least level of operating system privilege necessary in order to carry out the goal of the container.

Source: https://kubernetes.io/docs/concepts/security/overview/

BITS Pilani

Pilani Campus

Case Study: Platform Security Features in Microsoft Azure

Microsoft
werPoint Presentat