# Summary Report

## Report # 2

# COINS Summer School 2015 on Could Security

**Prepared by:**
Nabeel Ali Albahbooh

Ph.D. Student in Information Security Field
Norwegian Information Security Lab (NISLab)
Gjøvik University College (GUC), Norway
nabeel.albahbooh@hig.no

Course: IMT6003, COINS Summer School
Instructor: Dr. Hanno Langweg
hanno.langweg@hig.no

Submission Date: 21st September 2015

# Table of Contents

# 1. Executive Summary

Cloud computing is an emerging concept and its main goal is to deliver on-demand services over Internet, from a remote location, rather than on user's desktop, laptop, mobile device, or even on an organization's servers. In other words, cloud computing is a combination of multiple computing entities, globally separated, but electronically connected.  A service provider can deliver its applications, computing power, and storage services via the web. Thus, cloud computing becomes a location and device independent, in a sense that it does not matter where data is hosted nor where processing is performed.

Although, cloud computing offers a concentration of resources, it poses some potential security issues such as virtualization security, distributed computing, application security, identity management, access control, data privacy, and authentication. A single breach can cause significant loss. For instant, in a cloud environment, a single user may have multiple accounts between single or multiple service providers. Sharing a user's identity information along with user's associated attributes across services could lead to information identity privacy breach and loss.

COINS has organized the 1st summer school on could security in Metochi at Lesbos island, Greece.  Six topics have been delivered and discussed by various international lectures as listed in Table 1.

| No. | Topic | Lecturer |
|---|---|---|
| 1 | **SDN Security** | Sandra Scott-Hayward |
| 2 | **All You Ever Wanted to Know About Virtual Machine Introspection** | Zhiqiang Lin |
| 3 | **End-to-End Defense against Kernel Rootkits in a Cloud Environment** | Sachin Shetty |
| 4 | **Cloud Application Security** | Daniel Hedin |
| 5 | **Security of Network Monitoring Systems (NMS) for Cloud and HPC** | Andrei Costin |
| 6 | **Privacy in Cloud Environment** | Kaniz Fatema |

**Table 1: List of Topics**

This report consists of the following sections:

1. SDN Security – Sandra presented and talked about the concept behind SDN and its relation to OpenFlow protocol. SDN architecture and its main characteristics were briefly discussed. During the session, Sandra demonstrated an SDN scenario on a test VM machine. Various implementation challenges were discussed. Since every system is vulnerable, some security challenges were explained with some attack scenarios. Then, Sandra moved to the most interesting topic that discusses how to identify the requirements of a secure, robust and resilient SDN controller along with a life demonstration. Furthermore, some ideas were presented on how SDN could be used to enhance network security. Finally, a high level discussion about various activities of ONF security working group was addressed.
2. All You Ever Wanted to Know About Virtual Machine Introspection – Zhiqiang described the main idea of virtualization and its potential security threats. Besides, the workflow of virtual machine introspection (VMI) and its main two modes were explained. More technical hypervisor definitions were presented such as bare metal, hosted, native and emulation. The current challenges with VMI (i.e., semantic gap) and the proposed countermeasures were addressed. Zhiqiang described in very details how VMI could be used in the context of security (detection, prevention and recovery). Finally, Hands-on-labs were given to the participants to verify the concept of VIM using kernel debugging tool.
3. End-to-End Defense against Kernel Rootkits in a Cloud Environment – Sachin provided an introduction about kernel rootkits and how attackers use them to control a target system. More attention was given on the possibility to attack a critical system within a cloud computer environment. A threat model and some assumptions were discussed. Some background was given about CPU and Linux kernel protections. A detailed description about the proposed RootkitDet system components was explained. The current challenges to RootkitDet design were addressed and their relevant solutions. Finally, Sachin described how RookitDet was implemented and some experiments that were conducted for evaluation purposes.

4. Cloud Application Security – Daniel talked about an interesting topic on how to protect web applications on cloud environments. Some principles relevant to could and cloud application were presented in a high level. The main objective of this session was to address how it could be ensured that user information given to the application is safe (i.e., confidentiality). Numerous attack scenarios (e.g., content injection, 3rd party code injection and XSS) were explained with some examples. Next, Daniel provided a full description about his proposed solution using Information Flow Control (IFC) concept with its relevant scenarios. A distinct between an explicit and implicit flow was addressed. Finally, hands-on exercises were given to the participants to understand how IFC works.
5. Security of Network Monitoring Systems (NMS) for Cloud and HPC – Andrei provided a session on the concept of NMS and HPC with some security analysis and recommended countermeasures. An attack lifecycle was discussed in the context of NMS with various attack scenarios. Hands-on exercises were presented to show to the exiting vulnerabilities could compromise critical networks via NMS. Finally, some safeguards and security controls were addressed to effectivity protect critical networks.
6. Privacy in Cloud Environment – Kaniz provided a general idea about privacy and its major security issues in the context of cloud environments. Some possible threats if privacy is breach were explained with some real examples. The difference and similarity between privacy and secure was given a very high attention. Different phases of data lifecycle was discussed with regards to cloud computing. A paper survey was distributed to the participants to show them how privacy could affect their life. Kaniz proposed and described a model called P-PAAS for policy enforcement in case of various data holders (user, providers and authorities). Finally, the requirements of EU data protection directive were briefly presented and their relevant challenges in terms of implementation.
7. Questions Raised – This section provides a single question for each topic discussed in this report.
8. References – This sections provides a list of references for each topic discussed in this report.

## 2. SDN Security

The main feature of Software Defined Networking (SDN) is the physical separation of the control plane form the forwarding plane. Thus, it is driven by desire to provide user-controlled management of forwarding in network nodes.

SDN consists of three main layers: infrastructure, controller and application as shown in Figure 1. The controller is the middleware communication layer between the infrastructure and application that resides between northbound API and southbound API. Designing the security features of this controller becomes an important aspect of SDN.

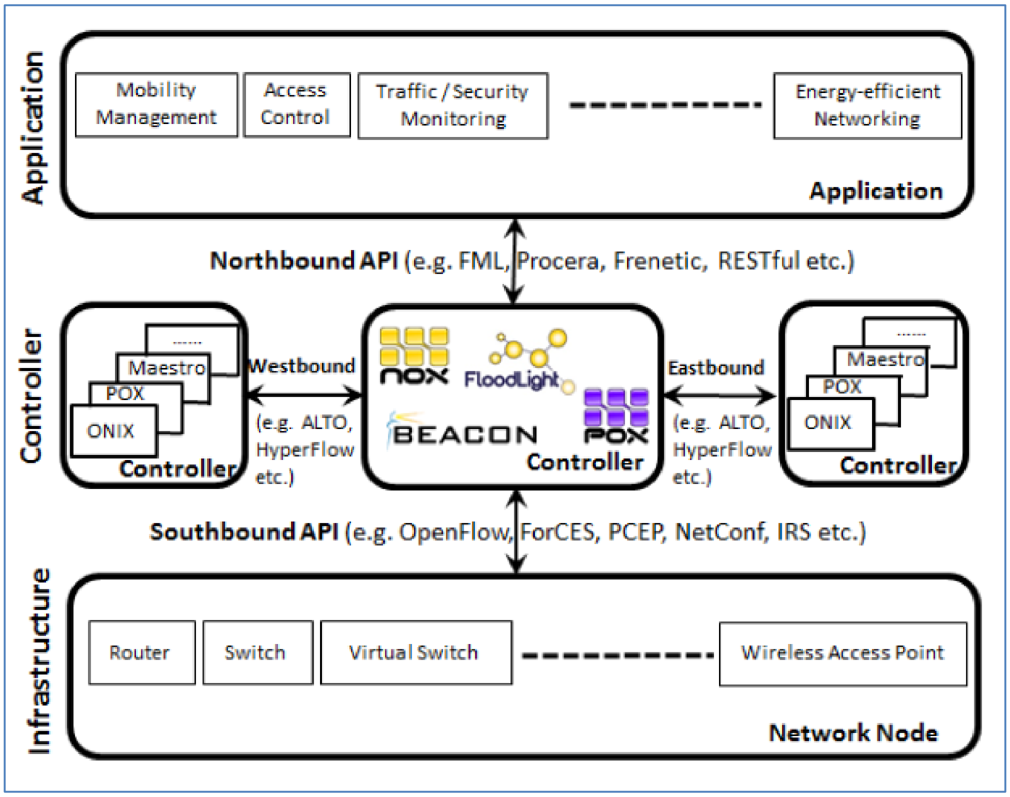


**Figure 1: SDN Architecture**

SDN relies on OpenFlow (OF) protocol to control the forwarding behavior of network switches. OF describes all type of functionalities and rules. The current researches focus on how to integrate SND with OF; and subsequently secure OF protocol in which the controller reduces the risk of attacks at the network control layer.

SDN faces implementation challenges in various aspects: performance, programmability, flexibility, scalability, interoperability and security. The network node within SDN must have a reliable processing speed (throughput and latency) with the capability to adopt any required changes of instructions that fulfil its new functions. On the other hand, SDN must be able to adapt systems that provide further features such as security controls. When SDN is implemented, it is important to consider the size and operation of the controller back-end database. Since SDN needs to be integrated with numerous networks, the interoperability challenges become a big concern during the implementation. Protection must be considered when designing and implementing SDN such as: DoS attack, man-in-the-middle (controller impersonation) and APIs issues as shown in Figure 2. In fact, increasing number of components and interfaces within SDN environment would introduce further security challenges of the controller. Thus, in order to mitigate those identified security risks, some security measures need to be introduced to the controller.

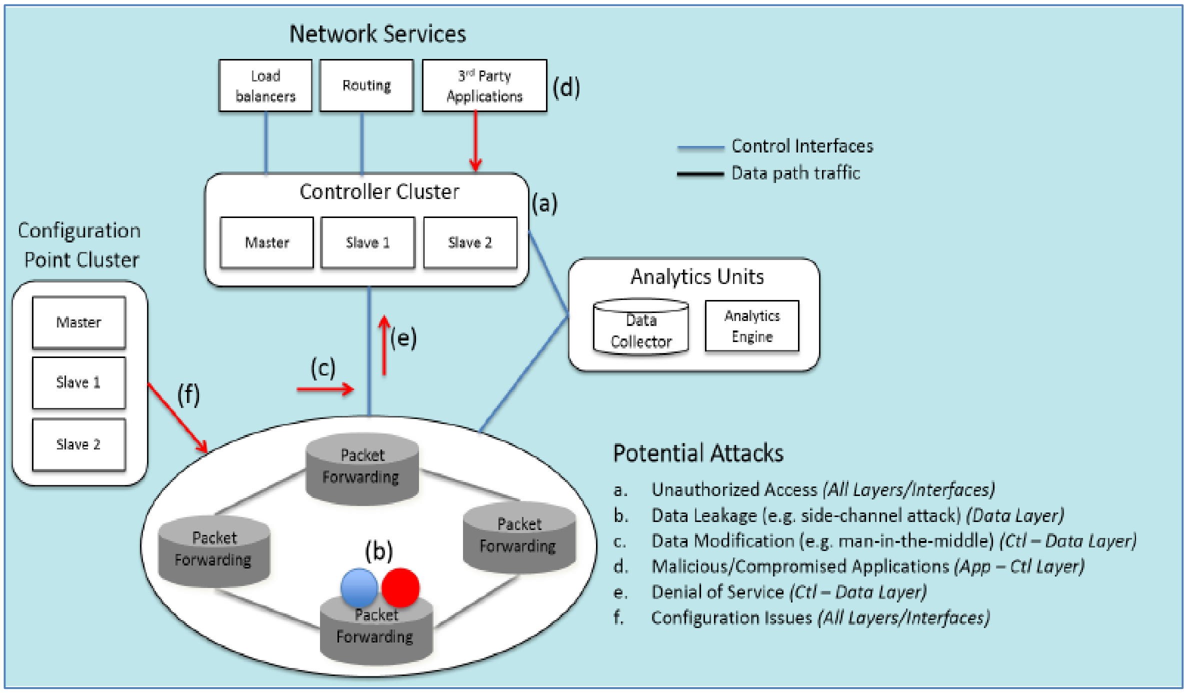

Network Services
Load balancers
Routing
3rd Party Applications
(d)
Control Interfaces
Data path traffic
Controller Cluster
(a)
Master
Slave 1
Slave 2
Configuration Point Cluster
Master
Slave 1
Slave 2
Analytics Units
Data Collector
Analytics Engine
(e)
(c)
(f)
Packet Forwarding
Packet Forwarding
Packet Forwarding
(b)
Packet Forwarding
Potential Attacks
a. Unauthorized Access (All Layers/Interfaces)
b. Data Leakage (e.g. side-channel attack) (Data Layer)
c. Data Modification (e.g. man-in-the-middle) (Ctl – Data Layer)
d. Malicious/Compromised Applications (App – Ctl Layer)
e. Denial of Service (Ctl – Data Layer)
f. Configuration Issues (All Layers/Interfaces)

**Figure 2: Potential Attacks and Vulnerabilities within SDN Enviornment**

Security best practices and solutions would be recommended in this regard:

- Identify misconfigurations, unauthorized access and irregularities to ensure correct functioning of the entities. This will prevent logic manipulation issues.
- Support mutual authentications between various domains to avoid insecure access to resources as well as impersonation of components.

- Slice and allocate network resources in order to establish an isolated environment. This will prevent data leakages and unauthorized access.
- Support application containerization in which the application would be authenticated during its setup.
- Use SDN configuration, control features and inherit security capabilities in a correct and secure manner.
- Monitor network events by using logging and forensics services within IDS/IPS.

## 3. All You Ever Wanted to Know About Virtual Machine Introspection

Virtualization (hypervisor) is the middleware layer between operating system and hardware. It performs the interaction between those layers and logically divides the resources for time sharing of different applications as presented in Figure 3. Virtual Machine Introspection (VMI) looks to insider from outsider.

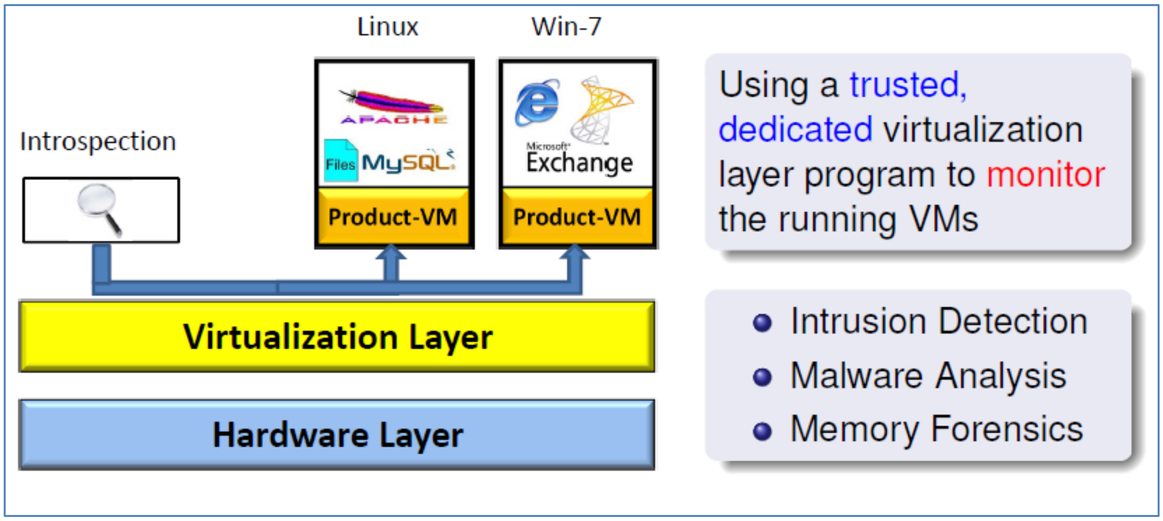


**Figure 3: Virtual Machine Introspection (VMI) Concept**

In general, there are two types of VM inspection: In-VM (runs on an operating system layer) and Out-of-VM (runs on a virtualization layer). In-VM provides many abstractions that allows to extract operating system and process state. In contract, Out-of-VM does not provide operating system abstractions, system call, file descriptor and variables. In-VM state can be directly accessed, and instantly executed without any trapping into hypervisor. Thus, In-VM offers fast processing speed rather than Out-of-VM because an additional address translation and word switching are required in Out-of-VM (time consumption). On the other hand, Out-of-VM is more secure that In-VM due to the fat that In-VM the security monitoring system can be disabled, and the security policy and the security enforcing mechanism can be tampered.

Using Out-of-VM for monitoring purposes would offer many advantages over traditional in-VM monitors because they run at a higher privilege level and are isolated from attacks within the guest operating system they monitor, and also because they are one layer below the guest operating system and can interpose all guest operating system events.

Since hypervisors only have access to low-level binary data, it is required to translate the data into higher-level abstractions to provide useful monitoring services.

There are two proposed approaches:

- Approach 1 – Redirecting kernel data of interest
- Approach 2 – Redirecting system call execution

In the first approach, we aim to use a native inspection software from SVM to transparently monitor and manage the state of guest VM (GVM), while in the second approach. On the other hand, the second approach pushes the execution of returning the process ID of the current process system call from SVM to GVM. We noted here that the system call is the only interface to requested operating system service.

Figure 4 illustrates that how we could reuse (legacy) binary code with a trust SVM to introspect the running GVM. The first approach is more fine-grained while the second approach is more practical. Also, the second approach is faster than the first approach in terms of performance.

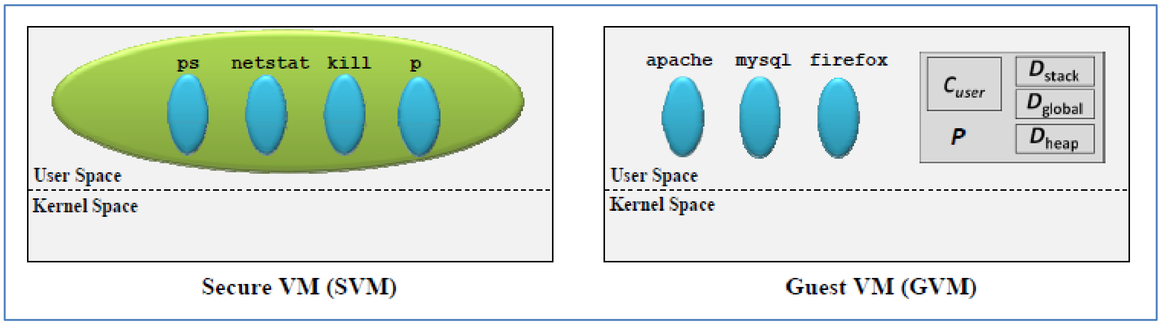


**Figure 4: Tow Approaches**

The key aspect of the proposed Out-of-VM solution is to respond to any potential attacks. It could be used for detecting attacks, preventing attacks and recovering attack.

In the detection mode, we can observe all changes (i.e., code or data modifications) to the kernel level or user level. If those changes are abnormal, then they will be considered as malicious behaviors. The prevention mode ensures that there is no authorized access to both kernel level and user level. The recovery mode is an automated fixing process that does require any human intervention. Moreover, this modes ensures that the system remains online without any interruption for high availability purposes.

## 4. End-to-End Defense against Kernel Rootkits in a Cloud Environment

Rootkits provide attackers camouflaged access to modify application, kernel data structures and code. The challenges with rootkits that they are difficult to detect because the malicious processes are hidden from security defense systems (e.g., antivirus, IPS and VMI).

In the context of cloud computing environment, there are specific requirements that need to be considered against kernel rootkits. The proposed defense system allows cloud administrators to quickly reverse the malicious modifications (i.e., end-to-end defense). It is compatible with existing commercial and open source cloud platforms.

Let us assume that an attacker intends to steal sensitive information from a mission critical system. He installs a rootkit tool in the operating system kernel by exploiting zero-day vulnerabilities in VMs. He gains control access over multiple VMs; and thus the risk is high.

Figure 5 presents a high level design of the proposed rootkit detection system (RootkitDet). It consists of five main components: registrar, detector, conductor, analyzer and inspector.

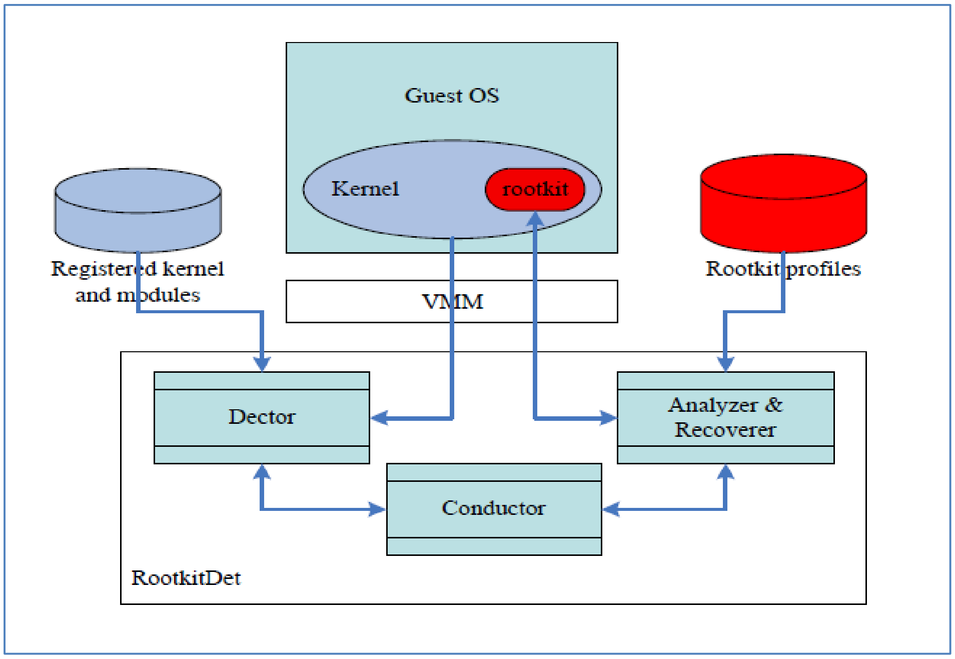


**Figure 5: RootkitDet High Level Architecture Design**

The registrar provides all necessary information of the kernel and legitimate codes. RootkitDet detects any possible suspicious codes in a memory or a region that holds an illegitimate code. This is accomplished by identifying all executable regions and comparing them with the expected executable regions which hold a legitimate code. The heart component of RootkitDet system is the conductor. It receives the detection report from the detector, then sends an alert to the administrator and activates the analyzer. It performs checksums of the loaded modules with their associated description. The analyzer diagnoses the detected rootkit by performing static analysis, categorizes it by matching the characteristic information with the profiles of known rootkits, and then it performs recovery of the guest operating system. The inspector is an interface used by the detector and analyzer to access the kernel space memory and CPU registers of guest operating system.

Although the RootkitDet offers great capabilities to detect and prevent rootkits, it has some limitations that need to be investigated. For instant, it cannot detect rootkits that are erased immediately after executed. Also, it may not able to detect all of hidden rootkits. It cannot prevent the installation of the kernel-level rootkits. Furthermore, it cannot certainly recover all modifications made by the rootkits.

## 5. Cloud Application Security

There are many aspects relevant to cloud such as: Internet, Web 2.0, inexpensive, virtualization, multi-tenancy, pay-as-you-go, utility storage, on demand 3rd parties and others.

Could web application is an application that offers over http as webpage such as JavaScript, html and CSS.

Figure 6 illustrates a high level design of the web application. It comprises of two main sides: client (web application) and server (the application backend). In the client side, all resourced are fetched for both 1st and 3rd parties. Clients use Content Delivery Networks (CDN) to relieve load on server. The communications between client and the cloud is maintained by AJAX. In the server side, on the other hand, both static and generated pages are provided to clients. It could be used for various cloud services such as: storage and authentication.

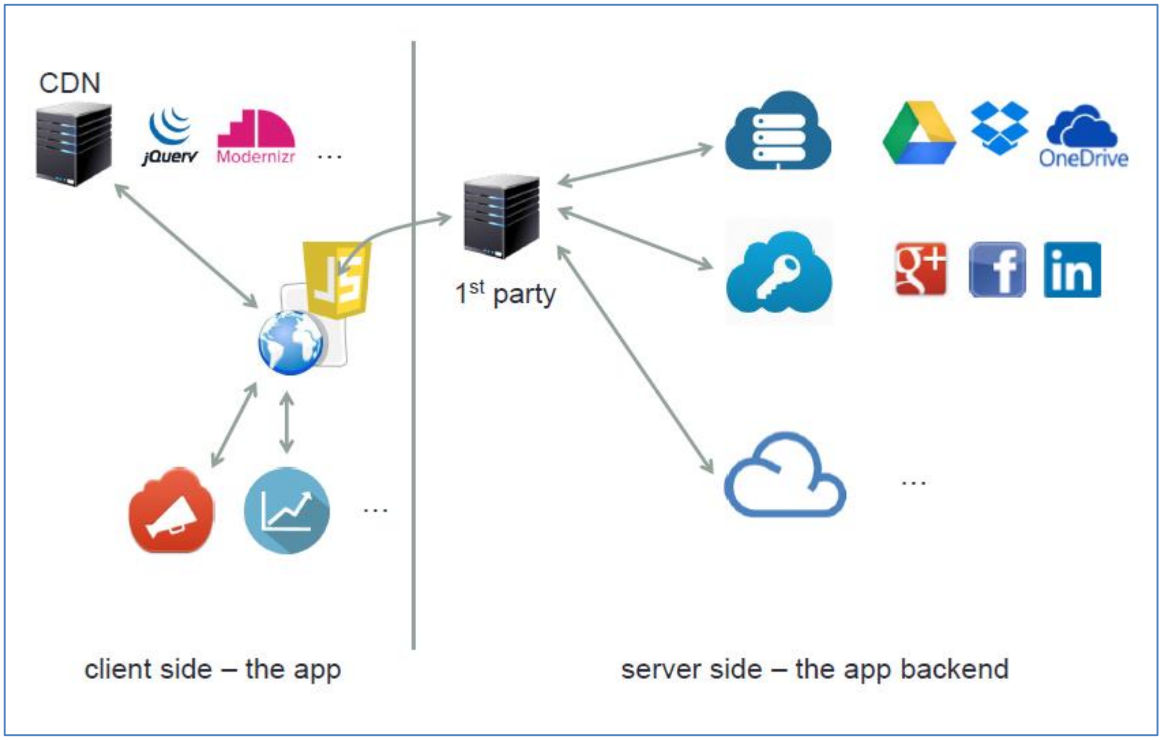


**Figure 6: A Simple Web Application**

One of the big challenge with cloud is the confidentiality aspects, especially those relevant to user data such as:

- How can we ensure that user information given to the applications is safe?

- What happens when a user enters sensitive data during login time?
- How can we guarantee that the credentials are only sent back to the 1st party and are not stolen?

Since The conventional access control does not protect after access has been granted. Thus, it is not enough to rely on it for protecting g the confidentiality of user data. For that reason, an Information Flow Control (IFC) is proposed to protect the confidentiality of user data against potential attacks such content injection, 3rd party code injection and cross site scripting (XSS). IFC intends to defend all attacks targeting websites.

IFC define policies where the information flows, how the information flows during computation (i.e., it analyzes flows at runtime). It blocks the information flows that violate the policy.

Generally, there are two main sources of information flow in programs. Explicit flows (data flow) and implicit flows (control flow). Explicit flows correspond to the direct copying of secrets. For instant, coping information from the password field to the variable *pwd* or sending a value over network using *XMLHttpRequest*. On the other hand, implicit flows comes from differences in side effects that encode sensitive information.

To overcome the security issue with the explicit flows, a static enforcement needs to be applied to tack explicit flows. It performs an inspection on the code before execution to determine if it contains any illegal flows; and then blocks execution if found. A proper classification of control flow is required to prevent the implicit flows.

## 6. Security of Network Monitoring Systems (NMS) for Cloud and HPC

High Performance Computing (HPC) refers to the practice of aggregating computing power in a way that delivers much higher performance than one could get out of a typical desktop computer in order to solve large problems in science, engineering or business.

Monitoring of the HPC systems and their components, such as clusters, grids and federations of clusters, is performed using Network Monitoring Systems (NMS) that helps system administrators in assessing and enhancing the health of their network infrastructure as shown in Figure 7. Ganglia, Cacti and Observium are the most popular and widely used NMS platforms.

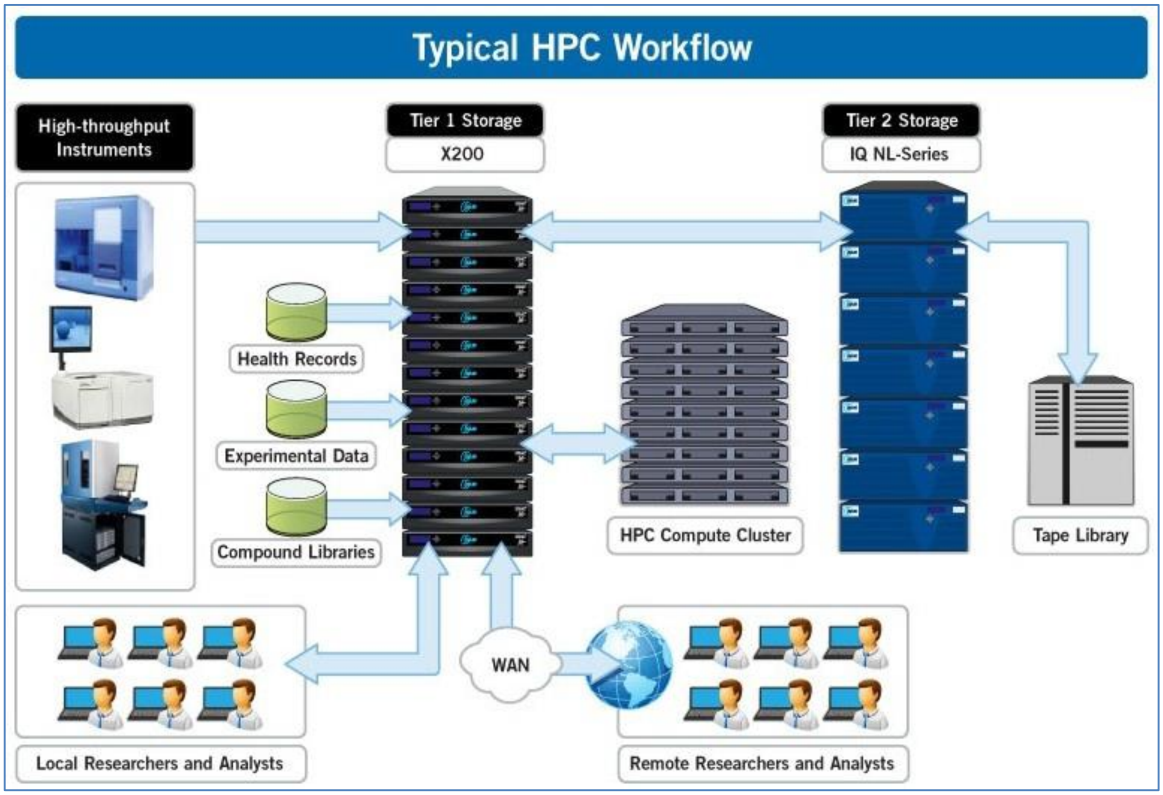


**Figure 7: HPC Concept**

NMS is vulnerable to several attacks such as information leakage (login bruteforce and social engineering), XSS, SQL injection, remote code execution, buffer overflows and kernel exploits.

Mimicry and lending attacks are well known attacks to NMS. An attacker can evade IDS by blending the normal network traffic or by crafting to blend with normal program flows and instructions. Similarly, an attacker can abuse the information gathered about the network infrastructure using openly accessible NMS in order to increase the chances of a successful infrastructure penetration on target systems.

Some countermeasures need to be considered in order to prevent any potential attacks on NMS. It is recommended to perform monitoring tasks on a segregated network that uses zero-knowledge protocols. Enabling password authentication on all monitoring interfaces are important. Securing the web application is required through deploying HTTP based authentication solution such as DDA or by implementing HTML forms authentication. Furthermore, the web interfaces need to be configured with HTTPs instead of unsecure HTTP. The use of proper CA-signed certificates non-vulnerable TLS/SSL implementations is an effective countermeasure. Appling IP-based access control list (ACL) and software security hardening are recommended whenever possible.

## 7. Privacy in Cloud Environment

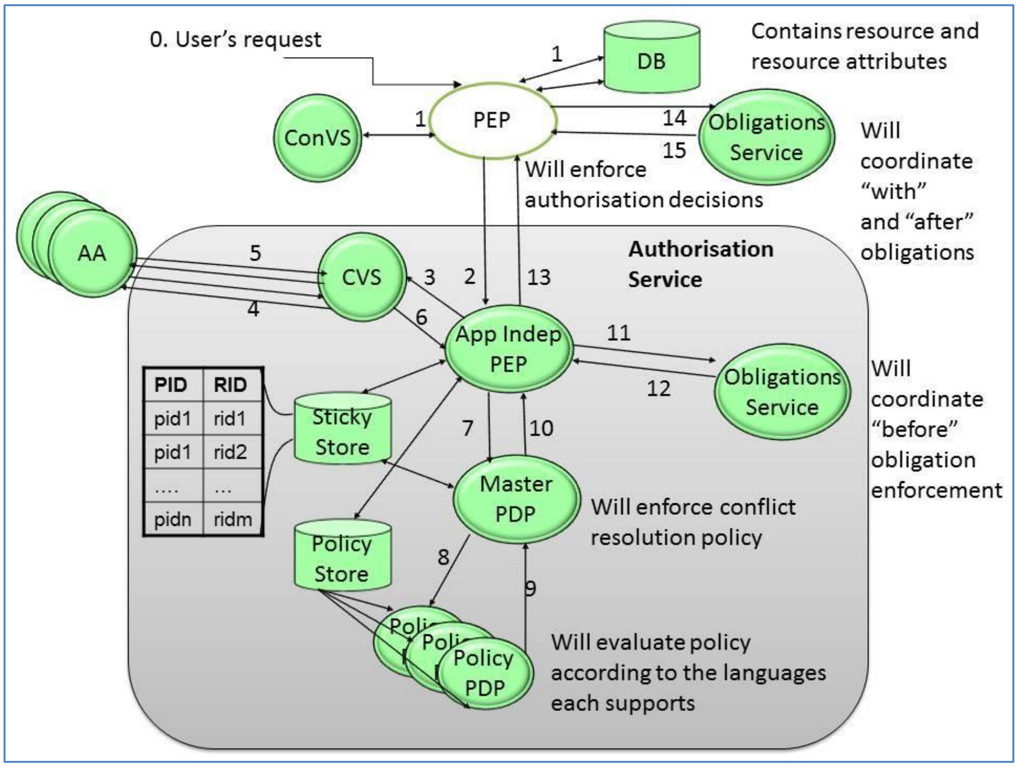
Privacy is a legal right that could be defined as the protection of an individual's independence, integrity, dignity, secrecy, anonymity, solitude and protection against intrusion into an individual's personal life or affairs.

Privacy could be viewed from different perspectives: personal body, personal behaviors, personal communications and personal data.

There are many privacy challenges in the context of cloud computing including: sharing data among multiple providers with different services, polices and location. Those providers may not have a proper control over their employees or physical location of data. Furthermore, there is a challenge on how to manage duplicated data and apply identity management. When data is distributed among various providers, conflicting laws from different jurisdictions becomes a nightmare.

Providers need to implement proper privacy policies of their users to control who has the right to access their personal data. A Privacy-Preserving Advanced Authorization System (P-PAAS) model is proposed to ensure multiple and combined privacy policies from providers and various authorities: data issuer and law are properly enforced.

The P-PAAS consists of several components: Application Independent Policy Enforcement Point (AIPEP), Credential Validation Service (CVS), Master PDP, policy store, obligation service and ontology mapping server as illustrated in Figure 8. The AIPEP coordinates all actions performed between various application authorization components. The CVS validates and checks all direct and indirect issued credentials. The master PDP collects multiple PDPs, obtains their authorization decisions, and then resolves any conflicts between these decisions. The policy is saved in a dedicated store, and protected (e.g., encrypted or digitally singed) if the store is not trustworthy. Three user actions (before, after or simultaneously with the performance of user action) are performed as part of obligations. It might be performed before a user is given access, after he has been granted access or during the access. The ontology mapping server is held as a lattice, and returns the relationship between two different terms.


0. User's request
1
DB
Contains resource and resource attributes
ConVS
1
PEP
14
15
Obligations Service
Will coordinate "with" and "after" obligations
Will enforce authorisation decisions
AA
5
4
CVS
3
2
13
6
Authorisation Service
App Indep PEP
11
12
Obligations Service
Will coordinate "before" obligation enforcement
Sticky Store
7
10
Master PDP
Will enforce conflict resolution policy
Policy Store
8
9
Policy PDP
Will evaluate policy according to the languages each supports

| PID | RID |
|---|---|
| pid1 | rid1 |
| pid1 | rid2 |
| …. | … |
| pidn | ridm |

**Figure 8: Privacy-Preserving Advanced Authorization System (P-PAAS) Infrastructure**

The proposed P-PAAS mode faces some limitations. It does not perform monitoring for data access and sharing. It requires to have compliance verification methods and a mechanism to identify and verify privacy metrics. Also, it needs to address technical means to satisfy the right of data portability.

# 8. Questions Raised

## 8.1 SDN Security

**Q: How the firewall differ from open table in terms of functionality?**

A: The open table does not work on a stateful inspection mode as the firewall does.

A stateful inspection is works at the network layer. Unlike static packet filtering, which examines a packet based on the information in its header, stateful inspection tracks each connection traversing all interfaces of the firewall and makes sure they are valid.

## 8.2 All You Ever Wanted to Know About Virtual Machine Introspection

**Q: Which one is better trust or untrust scenario?**

A: It depends on the situation. However, we prefer to use trust applications.

## 8.3 End-to-End Defense against Kernel Rootkits in a Cloud Environment

**Q: There are some rootkits work in a very fast manner and subsequently the detector could not detect them, why?**

A: Because these rootkits work in millisecond and modify some files. The detector would be able to catch them. The solution for this issue is to run continuous detection analysis.

## 8.4 Cloud Application Security

**Q: Is there any limitation to your proposed ICF solution? Does it work in all environment?**

A: In fact, my suggested ICF uses JSFlow which is a Java IFC interpreter. Thus, it is limited to Java environment.

## 8.5 Security of Network Monitoring Systems (NMS) for Cloud and HPC

**Q: Have you noticed that educational websites are not well secure? Why?**

A: Yes, most of educational websites are insecure due to the lack of security awareness (not aware of risks). Thus, I really encourage researchers to study about the mutuality security level of educational website and its relevant security awareness aspects.

## 8.6 Privacy in Cloud Environment

**Q: Who cares about privacy in the market?**

A: Only small size organizations care about user right and privacy rather than big enterprises. This because if there is a privacy breach, they have money to pay for any consequences (e.g., paying fine).

# 9. References

## 9.1 SDN Security

[1] Sandra Scott-Hayward, “SDN Security”, COINS Summer School, Sessions Notes, August 2015.

[2] Scott-Hayward S. S, Natarajan S, Sezer S, “A Survey of Security in Software Defined Networks”, IEEE Communication Surveys and Tutorials, Volume PP, Issue 99, July 2015.

[3] Scott-Hayward S, Kane C, Sezer S, “Operation Checkpoint: SDN Application Control”, 2nd IEEE International Conference on Network Protocols (ICNP), Raleigh, NC, USA, October 2014, pp. 618 – 623.

[4] Scott-Hayward S, “Design and Deployment of Secure, Robust, and Resilient SDN Controllers”, 1st IEEE International Conference on Network Softwarization (NetSoft), London, UK, April 2015, pp. 1 – 5.

## 9.2 All You Ever Wanted to Know About Virtual Machine Introspection

[5] Zhiqiang Lin, “All You Ever Wanted to Know About Virtual Machine Introspection”, COINS Summer School on Cloud Security, Sessions Notes, August 2015.

[6] Bauman E, Ayoade G, Lin Z, “A Survey on Hypervisor-Based Monitoring: Approaches, Applications, and Evolutions”, ACM Computing Surveys (CSUR), Volume 48, Issue 1, Article No. 10, August 2015.

## 9.3 End-to-End Defense against Kernel Rootkits in a Cloud Environment

[7] Sachin Shetty, “End to End Defense against Rootkits in Cloud Environment”, COINS Summer School on Cloud Security, Sessions Notes, August 2015.

[8] Zhang L, Shetty S, Liu P, Jing J, “RootkitDet: Practical End-to-End Defense against Kernel Rootkits in a Cloud Environment”, 19th European Symposium on Research in Computer Security (ESORICS), Wroclaw, Poland, September 2014, Volume 8713, pp. 475 – 493.

## 9.4 Cloud Application Security

[9] Daniel Hedin, "Cloud Application Security", COINS Summer School on Cloud Security, Sessions Notes, August 2015.

[10] Hedin D, Sabelfeld A, "A Perspective on Information-Flow Control", Chalmers University of Technology, Gothenburg, Sweden.

## 9.5 Security of Network Monitoring Systems (NMS) for Cloud and HPC

[11] Andrei Costin, "Security of Network Monitoring Systems (NMS) for Cloud and HPC", COINS Summer School on Cloud Security, Sessions Notes, August 2015.

[12] Andrei Costin, "All Your Cluster-Grids are Belong to Us: Monitoring The (in)security of Infrastructure Monitoring Systems", 1st IEEE Workshop on Security and Privacy in the Cloud, , Florence, Italy, September 2015.

## 9.6 Privacy in Cloud Environment

[13] Kaniz Fatema, "Privacy in Cloud Environment", COINS Summer School on Cloud Security, Sessions Notes, August 2015.

[14] Fatema K, Chadwick D W, Lievens S, "A Multi-privacy Policy Enforcement System", , 6th IFIP WG 9.2, 9.6/11.7, 11.4, 11.6/PrimeLife International Summer School, Helsingborg, Sweden, August 2010, Volume 352, pp. 297 – 310.