Fachhochschule Köln
Cologne University of Applied Sciences

Campus Gummersbach

Open Universiteit
www.ou.nl

A Distributed Virtual Computer Security Lab with Central Authority

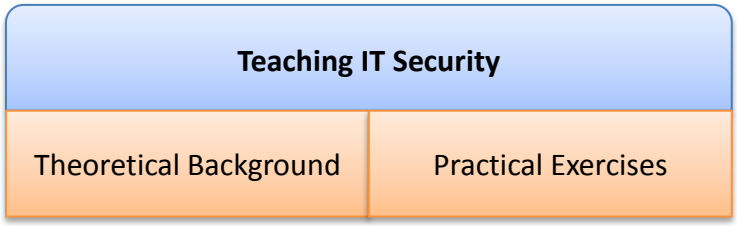
Jens Haag, Tobias Horsmann, Stefan Karsch
Cologne University of Applied Sciences, Germany
Harald Vranken
Open Universiteit, Netherlands

Agenda

- Introduction
  - Learning Scenario
  - Computer Security Lab
  - Virtual Computer Security Lab
- Distributed Virtual Computer Security Lab (DVCSL)
  - Functional Principles
  - Central Authority
- Conclusion and Future Perspectives

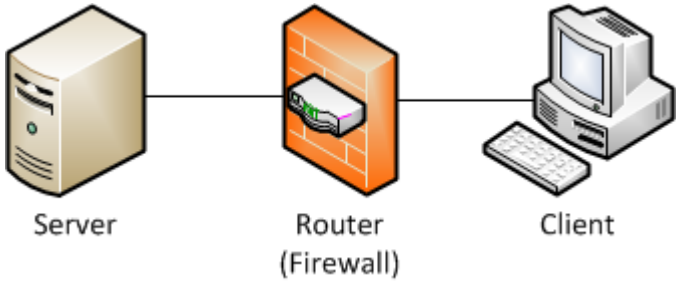
Learning Scenario I

- Teaching security of computer systems and networks (IT Security)
  - Basics (e.g. network configuration, routing)
  - Advanced content (e.g. security management and measurements, firewalls)
- Students learn the theoretical background
  - in lessons
  - from textbooks
- Knowledge is illustrated, deepened and anchored by carrying out practical exercises



Learning Scenario II

- Example exercise: Setup, configure and secure a network
  - Server, which provides services
  - Router with firewall
  - Client



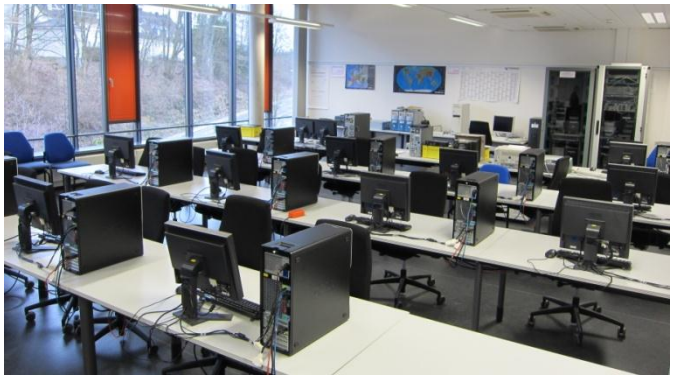
- Students can gather several practical experiences
  - Configuring the network interfaces and the routing
  - Providing and using internet services, e.g. HTTP, SSH, FTP
  - Adding security measures, e.g. configuring the firewall
  - Attacks („Hacking“), e.g. Buffer Overflows

Computer Security Lab

- At the university: Room containing computer systems
- Network isolated from the outside world (Internet)
- Group work is possible

- Students work with super-user rights
- After a session, reinstalling the operating system may be necessary

- Not suited for distance learning
- Significant administration effort necessary
- Usually closed in the evening hours and on weekends

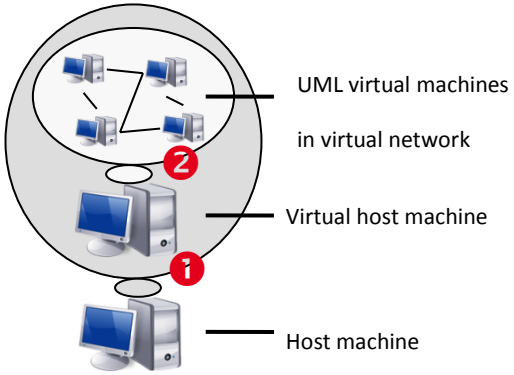


Virtual Computer Security Lab (VCSL)

- General idea: Move the lab to the students
- Computer Security Lab nested in a virtual environment
- Stand-alone environment, composed of two virtualization layers

- Layer ❶: Between Host machine and Virtual host machine
  - Based on free virtualization software (e.g. VMware player, Oracle VirtualBox)
  - Typically runs on almost all student computers

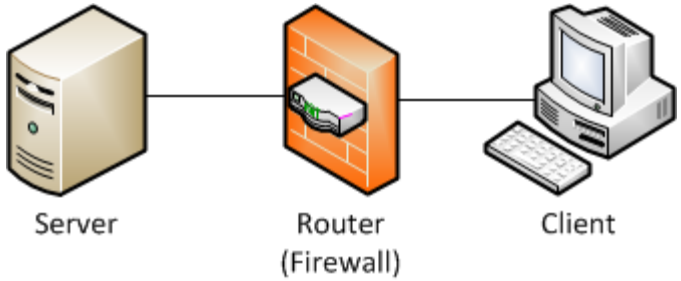
- Layer ❷: Between Virtual host machine and highly resource efficient scalable UML virtual machine
  - Based on Netkit, applies User Mode Linux (UML)
  - Multiple virtual hosts and isolated, virtual network(s)



Using the Virtual Computer Security Lab

- Our VCSL is an isolated secured software environment
- Provided to the students on a DVD
- Can easily install and run on students own computer

- (Virtual) Hardware setup for the example exercise
  - Start the VCSL environment
  - Issue 3 commands
  - Three virtual hosts
  - Network interfaces in separated broadcast domains




```
vstart server --eth0=netA
vstart router --eth0=netA --eth1=netB
vstart client --eth0=netB
```

Experiences with the Virtual Computer Security Lab

- Advantages of the current VCSL version
  - Students can work wherever and whenever they want
  - Reduced administration effort
  - Students can use their own computer, no extra hardware required
  - Decentralized approach usable for distance education
  - Already used successfully in a IT Security course by more than two hundred students with only few minor problems

- Disadvantages of the current VCSL version
  - Students work alone, group work is not possible
  - University cannot assist or grade the work of students

Distributed Virtual Computer Security Lab (DVCSL)

- Improvement: Distributing the virtual lab – DVCSL

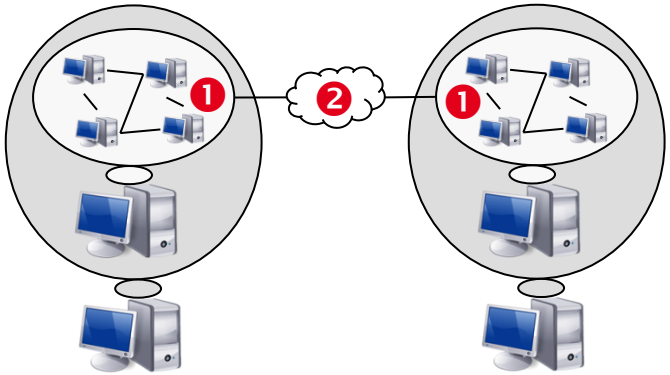
- Goals
  - Two or more geographically distant students connect their local VCSLs across an external intermediate network (e.g. the internet)
  - Synchronous group work, assigning roles and tasks for performing assignments (e.g. Hacker, Administrator, User)
  - Scenarios closer to the situation in real networks
  - Preserving the properties of a computer security lab (e.g. isolated network)

- DVCSL should unite the benefits of working independent from time and location, offering the possibility for group work

DVCSL Functional Principles - Requirements

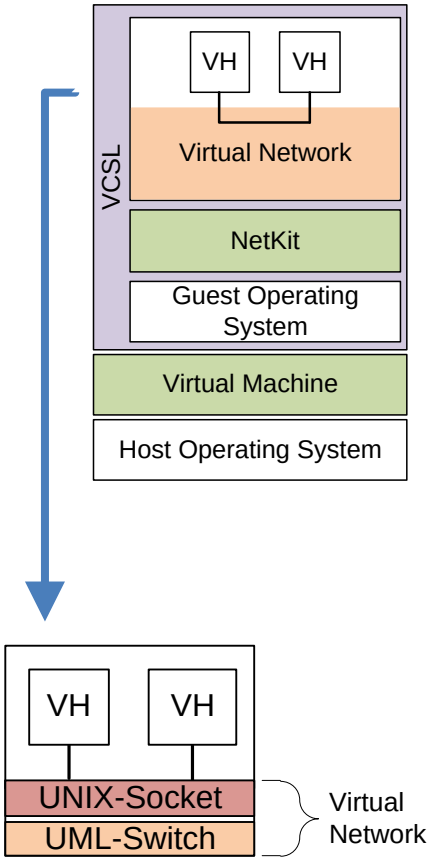
- Students should have the impression of being connected within an Ethernet LAN
  - Connected virtual networks must behave like a single broadcast domain
  - Requires transparent connection on OSI-layer 2 (Data Link Layer)

- Requirements
  - Extracting and injecting network data on virtual network level ❶
  - Sending and receiving the extracted network data ❷



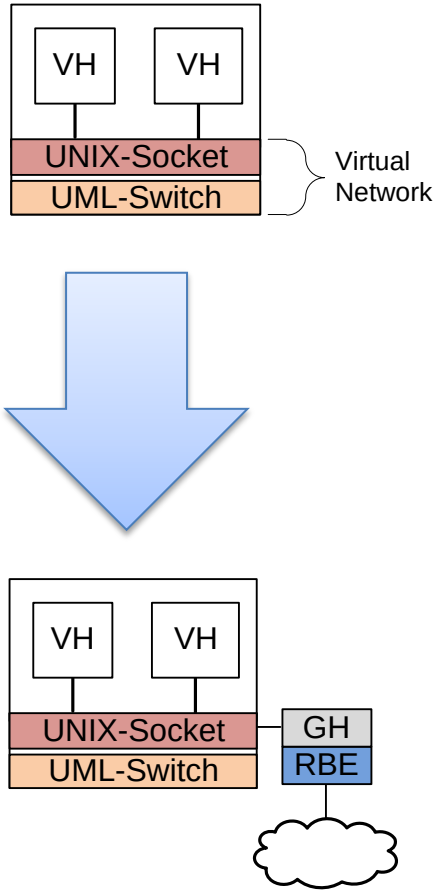
DVCSL Functional Principles – Basic Architecture

- Virtual networking architecture of Netkit/UML
  - Virtual hosts (VH) communicate with each other by using UNIX-Socket and UML-Switch
  - Each virtual network has its own, separated instance of UML-Switch and UML-Socket
- VH's and UML-Switch are attached to one UNIX-Socket
  - UML-Switch listens for incoming data sent by the VH
  - UML-Switch reads the data and writes it to all other connected VHs (HUB-like behavior)
  - A host in a virtual network receives all data sent by other hosts within the virtual network



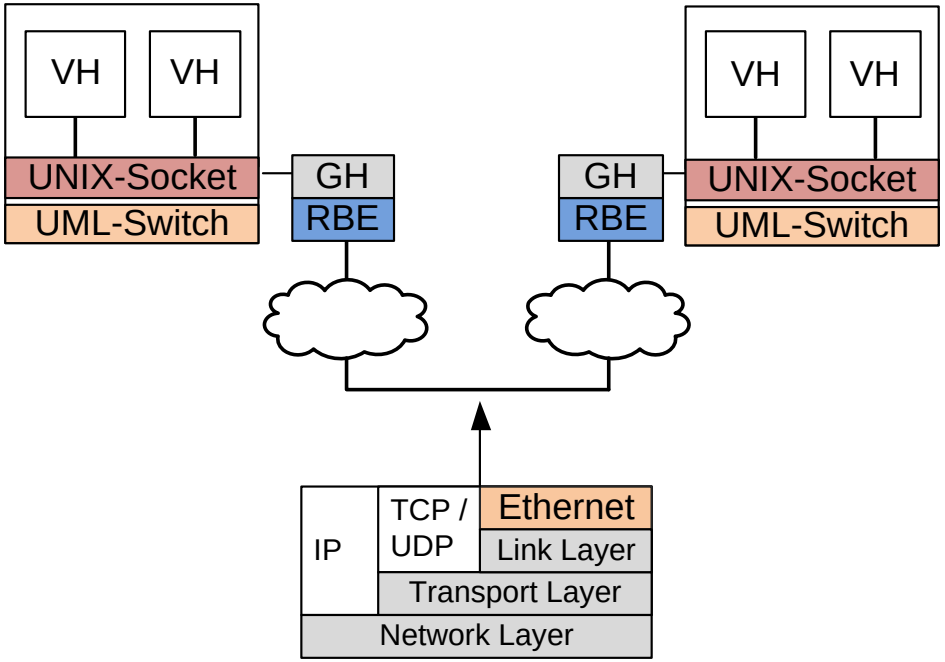
DVCSL Functional Principles - Extension

- Extending Netkit/UML with an additional interface
  - New software component, not jailed in the Netkit environment
  - Access to virtual network and WAN, e.g. the internet
- Extracting and injecting (ghost host - GH)
  - Invisible host connected to the virtual network
  - Can read and write ethernet frames via UNIX-Socket
- Sending and receiving (remote bridge endpoint - RBE)
  - Encapsulates ethernet frames in transport protocol (IP)
  - Distant networks cannot intercommunicate with WAN



DVCSL Functional Principles – Example Architecture

- Architecture layout example
  - Two students with VCSL's
  - Each VCSL has two virtual host
  - Direct (P2P) connection between students computer
  - Isolated network due to encapsulated ethernet frames



Experiences with the DVCSL

- Interface can connect two virtual networks
- Transparent
- Small and lightweight additional software component
- No need to modify Netkit or UML

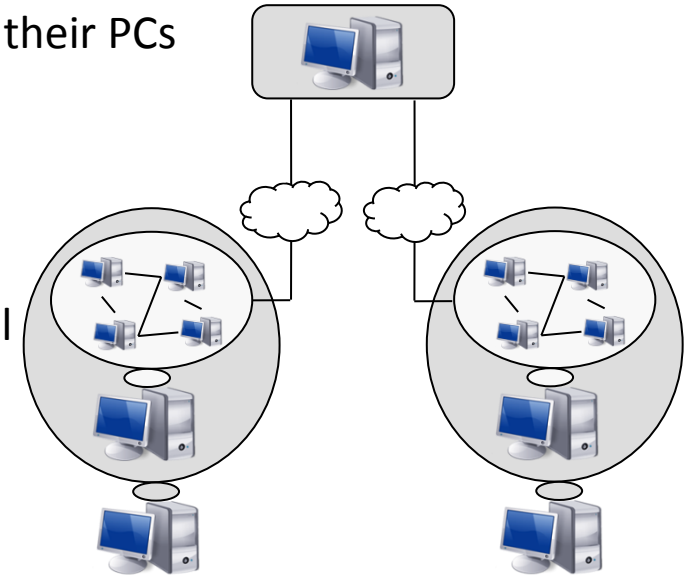
- Students must know the IP addresses of their remote partner(s)
- Additional network configuration on virtualization layer
- Connection of three or more VCSLs is possible, but can lead into infinite circular flow of network packets
- Requires careful planning by students
- May consume too much time

DVCSL with Central Authority (CA) I

- Further improvement: Central Authority

- Goals
  - Optimize organization and management when students work together
  - Reduce administration effort for students on their PCs

- DVCSL with CA
  - The CA is typically hosted at the university
  - Each student can connect one or more virtual networks to the CA in one session
  - Additional session can be opened and closed on demand



DVCSL with Central Authority II

- CA acts as a point of distribution (session layer)
  - Network data, which is send between virtual hosts in different VCSLs, is forwarded transparently
  - Deterministic behavior, no circular flow of network packets
- CA offers management functionality (control layer)
  - Different groups of students can work in separate DVCSLs
  - No additional network configuration needed
- CA can observe user interaction and network data
  - Identify course students (user authentication)
  - Inspecting network data
  - Active probing (e.g. ping) for monitoring remote VCSLs

Using the DVCSL with Central Authority

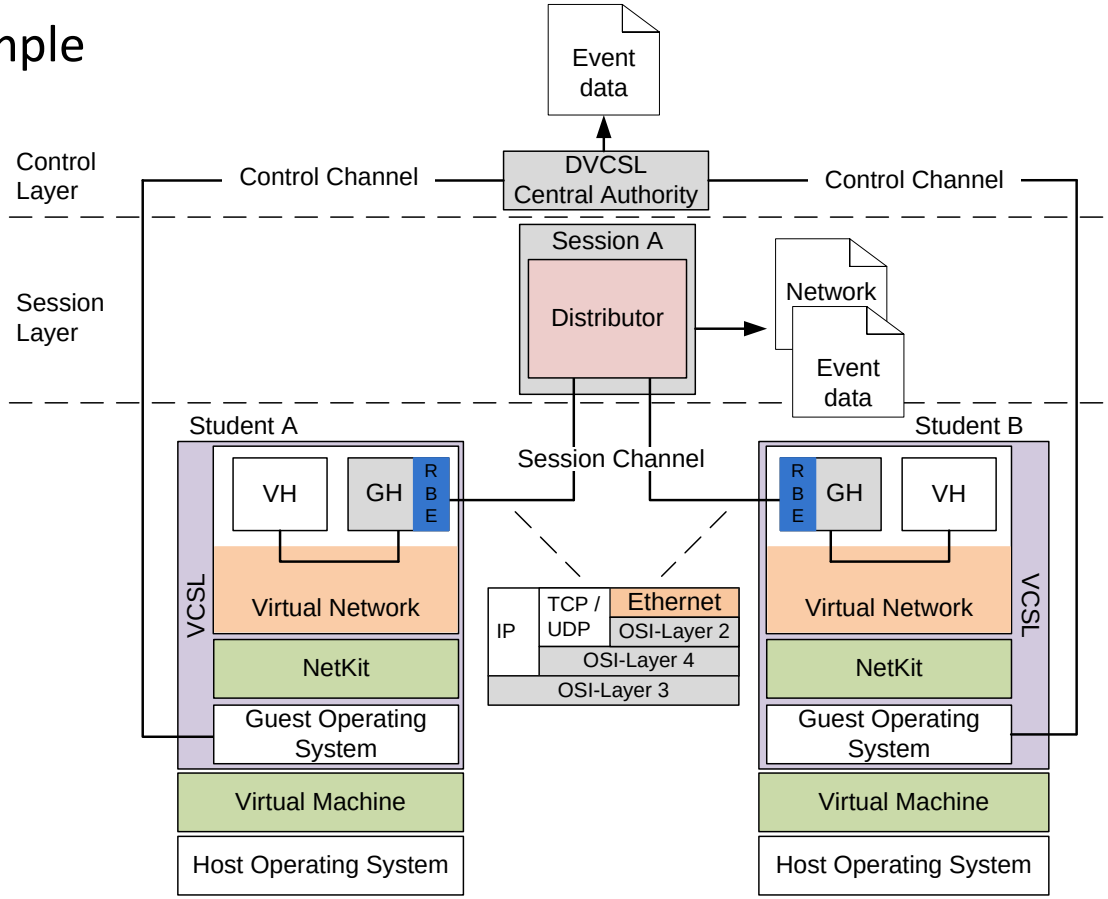
- Control layer used by students to operate the DVCSL
  - Querying a running session
  - Creating a new session
  - Joining a session
  - Deleting a session
- Simplified example: Connect two hosts
  - Student A opens a remote session_A
  - Student B joins session_A

```
Student A
// Start server with network
vstart server --eth0=netA
// Connect netA to CA
connect netA to <ip of CA>:<port>
// Create a new session
create session_A
// Join the session
join session_A
```

```
Student B
// Start client with network
vstart client --eth0=netA
// Connect netA to CA
connect netA to <ip of CA>:<port>
// Query running sessions on CA
query sessions
// Join the session from Student A
join session_A
```

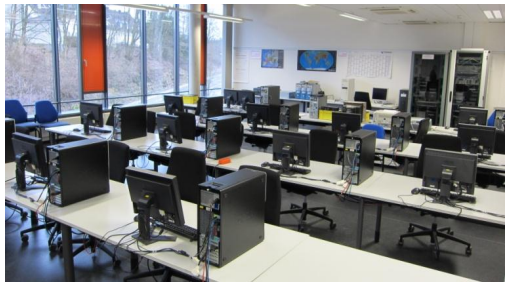
DVCSL with Central Authority – Example Architecture

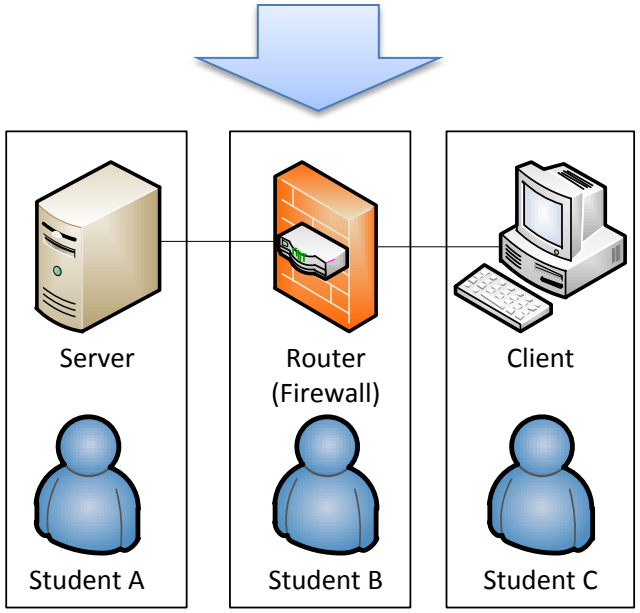
- Architecture layout example
  - Two students with remote VCSLs
  - Each VCSL runs one virtual host
  - Connected within one session
- Scalability
  - Students
  - Hosts
  - Virtual Networks
  - Sessions



Conclusion

- Remote students can perform network security exercises inside an encapsulated common networking environment
- Existing VCSL environment (Netkit/UML) enhanced by a communication interface
  - Ghost host
  - Remote bridge
  - No need to modify existing components
- Added a Central Authority
  - Simplifies the administration effort
  - Can assist students in various ways





Future Perspectives

- Short-term goals
  - Adding a suitable user management
  - Work out customized assignments
  - Create a final simplified DVCSL distribution
  - Using and evaluating the DVCSL-CA environment during next semester
- Long-term goals
  - Additional channel for user interaction (e.g. chat)
  - Assist students (e.g. automatic feedback program)
  - Use DVCSL's session data to grade students work

- Thank your for your attention!
- Please ask your questions…

- E-Mail: jens.haag@fh-koeln.de