Aufbau eines Ethical-Hacking-Labors mit automatisch generierten virtuellen Maschinen und zufällig ausgewählten Sicherheitslücken

33. DFN-Konferenz „Sicherheit in vernetzten Systemen“

Tobias Tefke[1] und Prof. Ralf C. Staudemeyer[2]

Hochschule Schmalkalden

[1] t.tefke@stud.fh-sm.de
[2] r.staudemeyer@hs-sm.de

Ausgangslage

Frühe Sensibilisierung erforderlich

SCHMALKALDEN UNIVERSITY OF APPLIED SCIENCES

- Sicherheitslücken allgegenwärtig
- Verantwortungsvoller Umgang elementar
- **Pflichtmodul „IT-Sicherheit“ (4. Semester):**
  - Grundlagenlehre
  - Eher allgemein und theoretisch
- **Wahlkurs „Security Hacking“ (6. Semester):**
  - Vertiefung theoretischer Grundlagen
  - Nutzung unseres Ethical-Hacking-Labors „SUASecLab“
  - Identifikation, Dokumentation und Behebung von Sicherheitslücken
  - Umsetzung eines praxisorientierten Projektes

Ausgangslage
Kurskonzeption

SCHMALKALDEN UNIVERSITY OF APPLIED SCIENCES

- Eigenes virtuelles Netzwerk für jeden Studenten (vier VMs)
- **VM 1:** allgemeiner Einstieg
  - Vermittlung der Grundlagen
  - Fest definierte Sicherheitslücken
- **VMs 2-4:** individuell generiert mit vergleichbarem Schwierigkeitsgrad
  - Auswahl von Netzwerkdiensten und Nutzerprogrammen
  - Zufällig gewählte Sicherheitslücken
  - Selbstständige Analyse durch Kursteilnehmer
  - Dokumentation als Prüfungsleistung
- **Sicherheitslücken der letzten vier Jahre:**
  - Fehlkonfigurationen
  - Softwarefehler

Ausgangslage

Technische Anforderungen

- Praxisnähe zu realen Systemen
- Zugriff auf individuelle Laborumgebung
- Schutz vor Aktivitäten außerhalb der abgesicherten Umgebung
- Selbstständige Lernstandkontrolle durch Capture-the-Flag-Umgebung
- Wettbewerbe durch Jeopardy- und Attack-Defense-Systeme
- Open Source zur Minimierung von Abhängigkeiten und Kosten

Hintergründe

Capture-the-Flag-Umgebungen und Cyber Ranges

- **Capture-the-Flag-Umgebungen:**
  - Einfachere, vorab definierte Szenarien
  - Intensive Nutzung von Gamification-Elementen
    - **Jeopardy:** dieselben Aufgaben für alle Teilnehmer
    - **Attack-Defense:**
      - Absicherung des eigenes Systems
      - Angriff des gegnerisches Systems
    - **Mixed:** Verknüpfung von Jeopardy- und Attack-Defense
- **Cyber Ranges:**
  - Komplexere (realitätsnähere) Systeme
  - Definition der Systeme durch Konfigurationssprachen

Hintergründe

Abgrenzung von bestehenden Umgebungen

SCHMALKALDEN UNIVERSITY OF APPLIED SCIENCES

- **Capture-the-Flag:**
  - Fast ausschließlich Jeopardy
  - Attack-Defense-CTFs haben Einschränkungen
  - Vordefinierte VMs oder „wenig Zufall“
- **Cyber Ranges:**
  - Keine CTF-Komponente $\Rightarrow$ keine selbständige Lernkontrolle
  - Zufallsmechanismen bieten (wenn vorhanden) bisher wenig Auswahl
- Sicherheitslücken schon einige Jahre bekannt
- **Neu:** Automatisierte CTF-Erstellung; VMs mit Zufallsauswahl neuerer Sicherheitslücken
- **Neu:** Unterstützung von Jeopardy- und Attack-Defense-Szenarien

Technischer Aufbau

Virtuelle Maschinen

- **Kali Linux:**
  - Analyse der anderen VMs
  - Einstieg mit Bash-CTF
- **Basic-VM:**
  - In Lehrveranstaltung besprochen
  - Leicht auffindbare, vordefinierte Sicherheitslücken
- **Prüfungs-VMs:**
  - Content-Management-System und Email-Server
  - Cloud- und Datei-Server
  - Developers' Box

Technischer Aufbau

Erzeugung der VMs

- **Infrastructure-as-Code:**
  - OS-Konfiguration mit Preseed-Datei
  - Installation von Programmen mit Bash-Skripten (über Packer)
  - Erzeugung zufallsbasierter VMs durch Python-Skripte
- **Auswahl der Sicherheitslücken:**
  - Definierte Wahrscheinlichkeiten pro Lücke
  - Mischung bekannter und neuer Lücken
  - Vermeidung von Frustration durch Pity-Mechanik
  - Speicherung in JSON-Datei

Technischer Aufbau

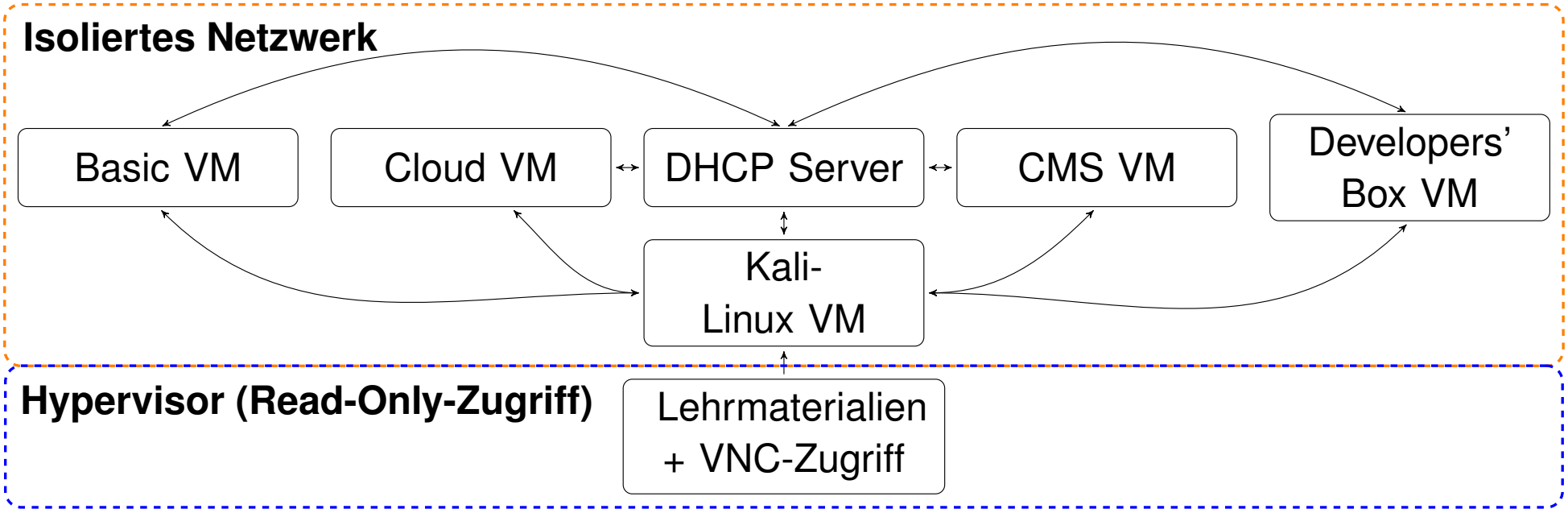
Verwaltung von Flags und VMs

SCHMALKALDEN UNIVERSITY OF APPLIED SCIENCES

- **Flags:**
  - Accounts mit zufälligen oder unsicheren Passwörtern
  - CVE-Nummern
  - Zufällig generierte Zeichenketten
- **Verwaltung der VMs:**
  - Rücksetzbarkeit mittels Backing-Stores
  - Replikation für Kali-VMs

Technischer Aufbau

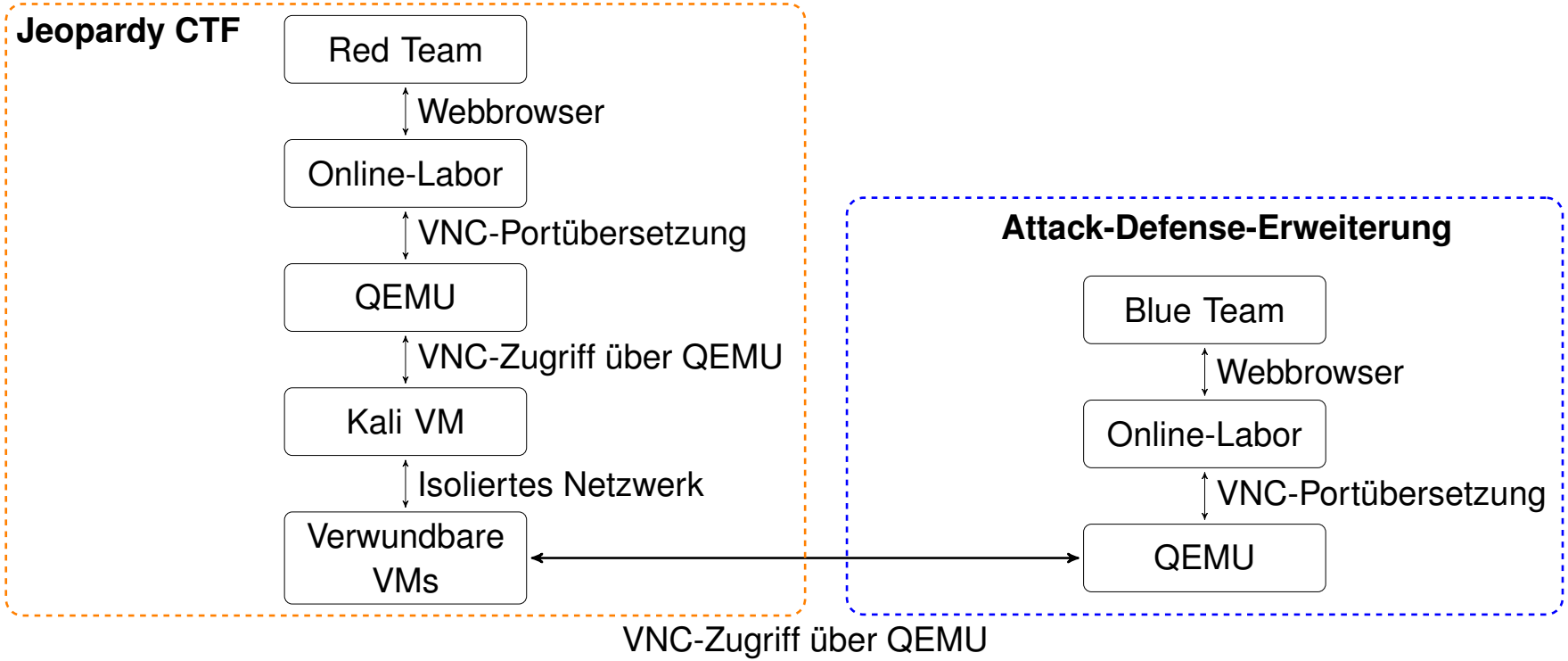
Netzwerkstruktur



Informationsfluss innerhalb einer VM-Gruppe

Technischer Aufbau

Zugriff auf VMs



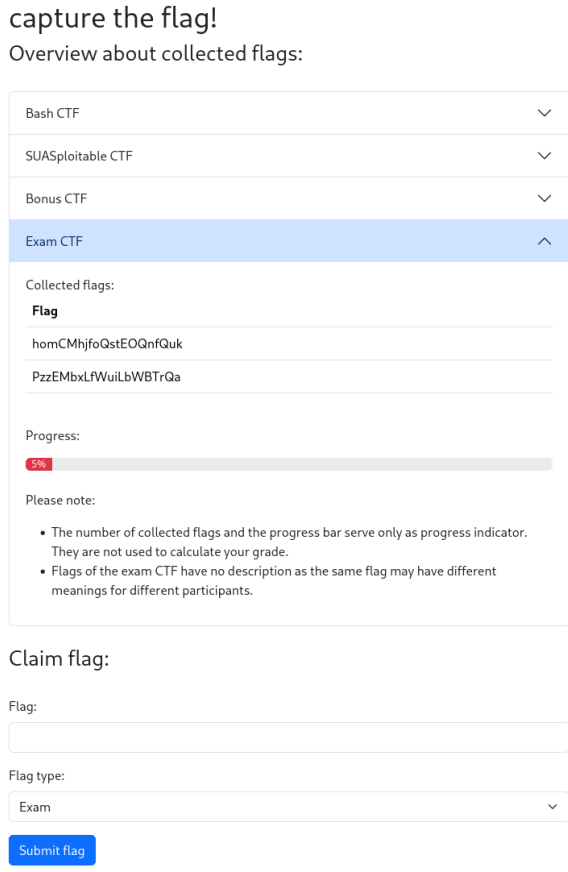
Zugriff auf die VMs. Links: Jeopardy-CTF. Rechts: Attack-Defense-Erweiterung.

Technischer Aufbau

Lernzielkontrolle

SCHMALKALDEN UNIVERSITY OF APPLIED SCIENCES

- **Überwachung des Lernziels:**
  - Mittels CTF-Umgebung
  - Überblick gesammelter Flags
- **Bewertung des Lernziels:**
  - Anhand des Berichtes
  - Vergleich mit JSON-Dateien von Erzeugung der VMs



Verwertbarkeit

Chancen und Verwertbarkeit unserer Lösung

SCHMALKALDEN UNIVERSITY OF APPLIED SCIENCES

- **Chancen:**
  - Flexibler Zugriff auf Labor
  - Lernstandüberwachung
  - Eigenständige Lösungsfindung erforderlich
- **Verwertbarkeit:**
  - Bereitstellung der Laborsoftware auf *GitHub*[3] als Open Source
  - Umsetzung eigener Hacking-Labore auf Basis unseres Systems möglich
  - Einfügen weiterer Sicherheitslücken jederzeit möglich

[3] https://github.com/SUASecLab