

Praxisbericht: Strukturierung eines Hochschul- netzwerkes mit VLANs

Dr. Klaus Schmoltzi, Warp9 GmbH, Münster

Überblick

- Ausgangssituation und Ziele
- Exkurs: VLAN Tagging nach 802.1Q
- Pilotierung
- Inbetriebnahme des Produktivsystems
- Reglementierung des Internetzugriffs
- Ausblick

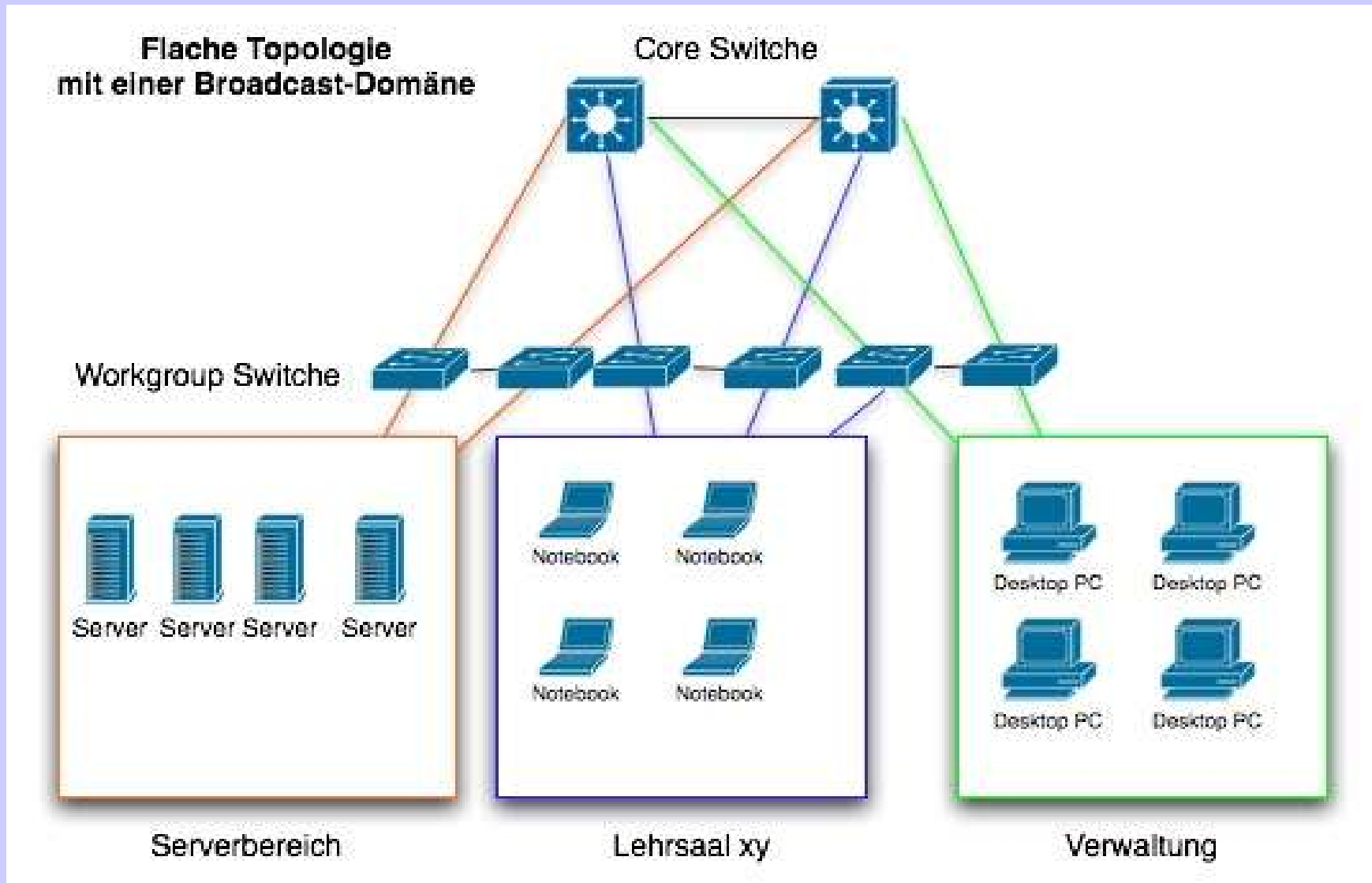
Ausgangssituation

- Bildungszentrum der Bundesfinanzverwaltung (u.a. Fachhochschule des Bundes)
- Die Studenten , sowie die Teilnehmer an Aus- und Fortbildung sind keine anonymen Anwender
- Flache Netzwerkstruktur d.h. alle Geräte befinden sich in einer einzigen Broadcast-Domäne
- Alle Studenten wurden mit Notebooks ausgerüstet, die nun an dieses Netz angeschlossen werden sollen
- Gleichzeitig können sich bis zu 800 Notebooks im Netz befinden. Zusätzlich existieren noch ca. 200 Verwaltungsrechner

Ausgangssituation

- Die Struktur der Verkabelung und der Switches ist neu erstellt worden und war somit vorgegeben
- Clients aus dem Verwaltungsbereich und Notebooks aus den Lehrsälen können sich am **selben Switch** befinden
- Die Client-Rechner werden durch Remoteinstallation über TFTP-Boot mit ihrem jeweiligen Betriebssystem (NT bzw. XP) „betankt“
- Die Anmeldeserver versorgen die Client-Rechner zusätzlich mit ihrer IP-Konfiguration per DHCP

Ausgangssituation



Ziele

- Durch eine Unterteilung der flachen Netzwerk-Topologie sollen die Broad- und Multicast-Pakete im jeweiligen Segment reduziert werden
- Leichtere räumliche Zuordnung der Client-Rechner anhand der über DHCP zugewiesenen IP-Adresse
- Implementierung von Sicherheitsrichtlinien
- Beibehaltung aller bisher verwendeten zentralen Softwarekomponenten (Remoteinstallation, Anmeldeserver, etc.)

Ziele

- Keine Änderung der bestehenden Verkabelung
- Kostengünstige Lösung
- Möglichkeit der Einbindung eines einfachen Mechanismus zur Reglementierung des Internetzugriffs aus den Lehrsälen heraus

Lösungsvorschlag

- Einsatz von Virtual LAN (VLAN) nach 802.1Q
- Aufbau eines VLAN-Routers unter Linux
- Einbindung eines DHCP-Servers auf dem VLAN-Router
- Einsatz von iptables zur Umsetzung von Sicherheitsrichtlinien
- Reglementierung des Internetzugangs durch einen Squid Proxy-Server

Exkurs: VLAN Tagging nach 802.1Q

- Erweiterung des Ethernet-Frames um 4 Byte
- Ein VLAN wird über einen **Tag** (Marker) in dieser Erweiterung identifiziert
- Der Tag wird durch eine Zahl zwischen 1 – 4094 dargestellt, dies ist die **VLAN ID**
- Ein Switch stellt Pakete mit einem Tag nur an die Ports zu, die zum selben VLAN gehören
- Die Ports eines Switches lassen sich durch VLAN Tagging somit segmentieren

Exkurs: VLAN Tagging nach 802.1Q

Aufnahme von 802.1Q Paketen mit Ethereal:

```
▼ Ethernet II, Src: Gvc_f5:96:a8 (00:c0:a8:f5:96:a8), Dst: Ibm_82:16:ca (00:11:25:82:16:ca)
  Destination: Ibm_82:16:ca (00:11:25:82:16:ca)
  Source: Gvc_f5:96:a8 (00:c0:a8:f5:96:a8)
  Type: 802.1Q Virtual LAN (0x8100)
▼ 802.1Q Virtual LAN
  000. .... = Priority: 0
  ...0 .... = CFI: 0
  .... 0000 0000 1010 = ID: 10
  Type: IP (0x0800)
  Trailer: 4665DB4A
▶ Internet Protocol, Src: 192.168.10.253 (192.168.10.253), Dst: 192.168.10.1 (192.168.10.1)
```

Die neu hinzugekommenen 4 Bytes teilen sich wie folgt auf:

1. Ether Type 802.1Q (2 Byte)
2. Tag Control Information – TCI (2 Byte)

Exkurs: VLAN Tagging nach 802.1Q

- Der Type „**802.1Q Virtual LAN**“ hat immer den Wert **0x8100** und signalisiert damit das Vorhandensein eines VLAN Tags
- Die Tag Control Information (TCI) gliedert sich in folgende Felder:
 - **User Priority Field** (3 Bit)
 - **Canonical Format Indicator** (1 Bit)
 - **VLAN ID** (12 Bit)

Exkurs: VLAN Tagging nach 802.1Q

Wie können VLAN Tags von einem Switch zu einem anderen Gerät (Switch bzw. Router) transportiert werden?

- Konfiguration eines sogenannten „Trunk“ bzw. „Tagged“ Port
- An diesem Port werden die VLAN Tags im Ethernet-Frame mit ausgegeben. Die VLAN Informationen können damit vom empfangenden Switch (oder auch Router) ausgewertet werden
- Vorsicht bei der Konfiguration:
Je nach Hersteller kann der Begriff „Trunk“ auch unabhängig von VLAN Tagging Verwendung finden

Exkurs: VLAN Tagging nach 802.1Q

Beispiel: HP ProCurve

```
===== TELNET - MANAGER MODE =====
Switch Configuration - VLAN - VLAN Port Assignment

Port  DEFAULT_VLAN  vlan2  vlan8  vlan9  vlan10
----  +  -----  -----  -----  -----  -----
10    | No             No     No     No     Untagged
11    | Untagged      No     No     No     No
12    | Untagged      No     No     No     No
13    | Untagged      No     No     No     No
14    | Untagged      No     No     No     No
15    | Untagged      No     No     No     No
16    | Untagged      No     No     No     No
17    | Untagged      No     No     No     No
18    | Untagged      No     No     No     No
19    | Untagged      No     No     No     No
20    | Untagged      No     No     No     No
21    | Tagged        Tagged Tagged Tagged Tagged

Actions->  Cancel  Edit  Save  Help
```

Exkurs: VLAN Tagging nach 802.1Q

Tipps zum Mitschneiden von VLAN Tags mit Ethereal:

wiki.ethereal.com/CaptureSetup/VLAN

Auf dem VLAN-Router selbst sind die Tags mit einem Sniffer nicht sichtbar! Der Treiber entfernt die Tags aus den Paketen, bevor die pcap-Library sie sieht. Daher sollte man sich einen Monitoring-Port am Switch konfigurieren und an diesem mit einem anderen Computer mitschneiden.

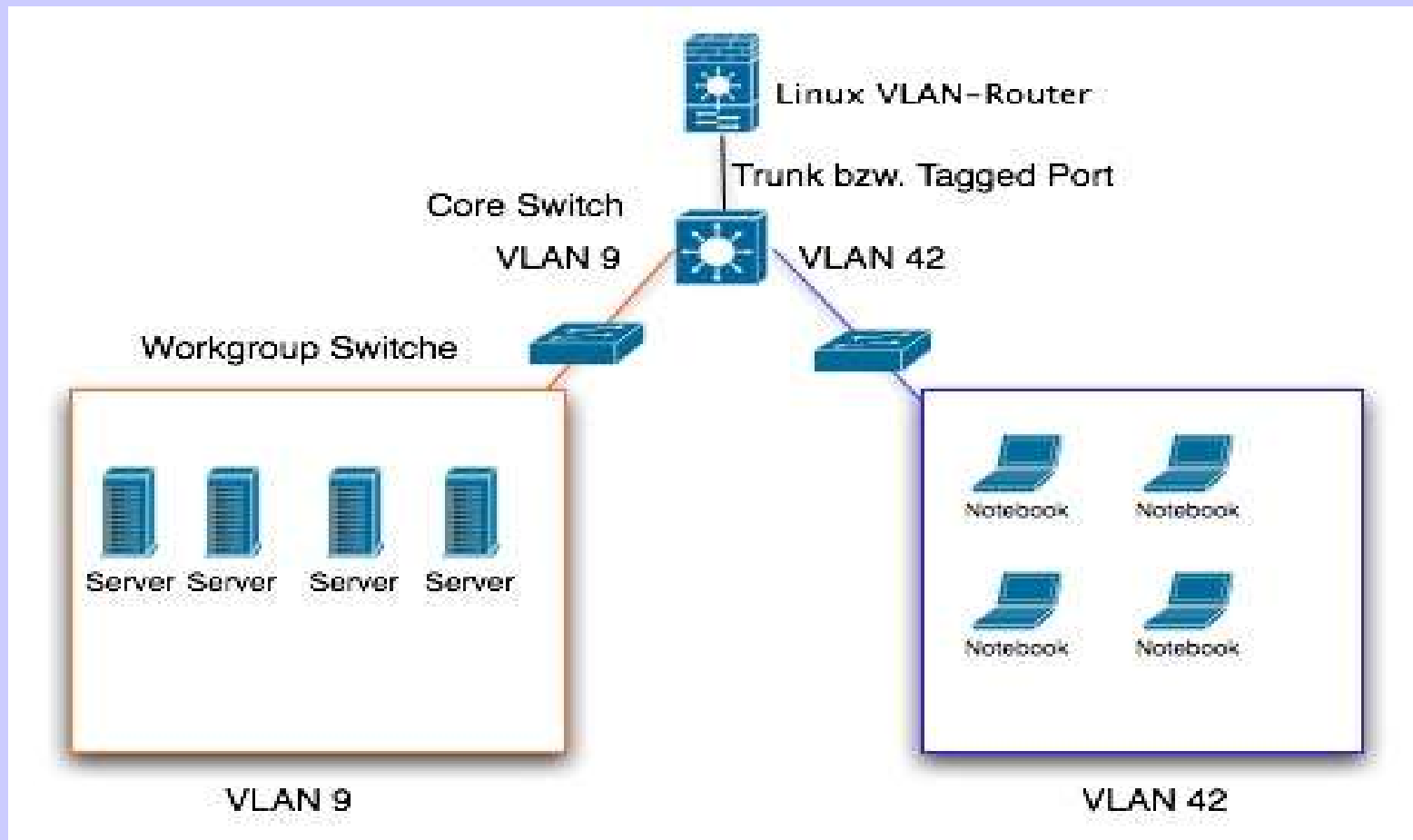
Tool zum Angriff auf VLANs: Yersinia*

yersinia.sourceforge.net

*Yersinia pestis – Erreger der Pest

Pilotierung

Pilotaufbau mit VLANs nach 802.1Q:



Pilotierung

Konfiguration von VLAN Tagging unter Linux:

- Das vlan RPM-Paket (Ver. 1.8) muss installiert sein
- Aktivierung des 802.1Q Kernelmoduls:

```
modprobe 8021q
```

- VLANs erzeugen

```
vconfig add eth0 [VLAN-ID]
```

Pilotierung

Namenskonvention für die erzeugten VLAN-Interfaces:

Über den Befehl

```
vconfig set_name_type VLAN_PLUS_NO_PAD
```

wird erreicht, dass die Interfacenamen **vlan9** statt **eth0.9** lauten

Hinweis:

Die VLAN-ID 1 sollte vermieden werden, da dies häufig die ID des Default_VLAN von vielen Switchen ist

Pilotierung

Konfiguration des DHCP-Servers unter Linux:

- Range: 192.168.[VLAN-ID].10 - 192.168.[VLAN-ID].254
- Spezielle Optionen für den TFTP-Boot der Clients wurden mit Ethereal ermittelt und in die DHCP-Konfiguration integriert

Pilotierung

Konfiguration der Core-Switche (Cisco Catalyst):

- Port im VLAN 42:

```
switchport access vlan 42  
switchport mode access
```

- Trunk bzw. Tagged Port:

```
switchport trunk encapsulation dot1q  
switchport mode trunk
```

Die Konfiguration der Workgroup-Switche wurde in dieser Phase nicht geändert

Pilotierung

Ergebnisse:

- Alle getesteten Softwarekomponenten arbeiten auch im gerouteten Netzwerk
- Nach einigen Anpassungen konnte der DHCP-Dienst von den Anmeldeservern auf den VLAN-Router übertragen werden
- **Unschön:** Die gewünschten Sicherheitseinstellungen mit iptables ließen sich nur durch Policies, die auf MAC-Adressen arbeiten realisieren

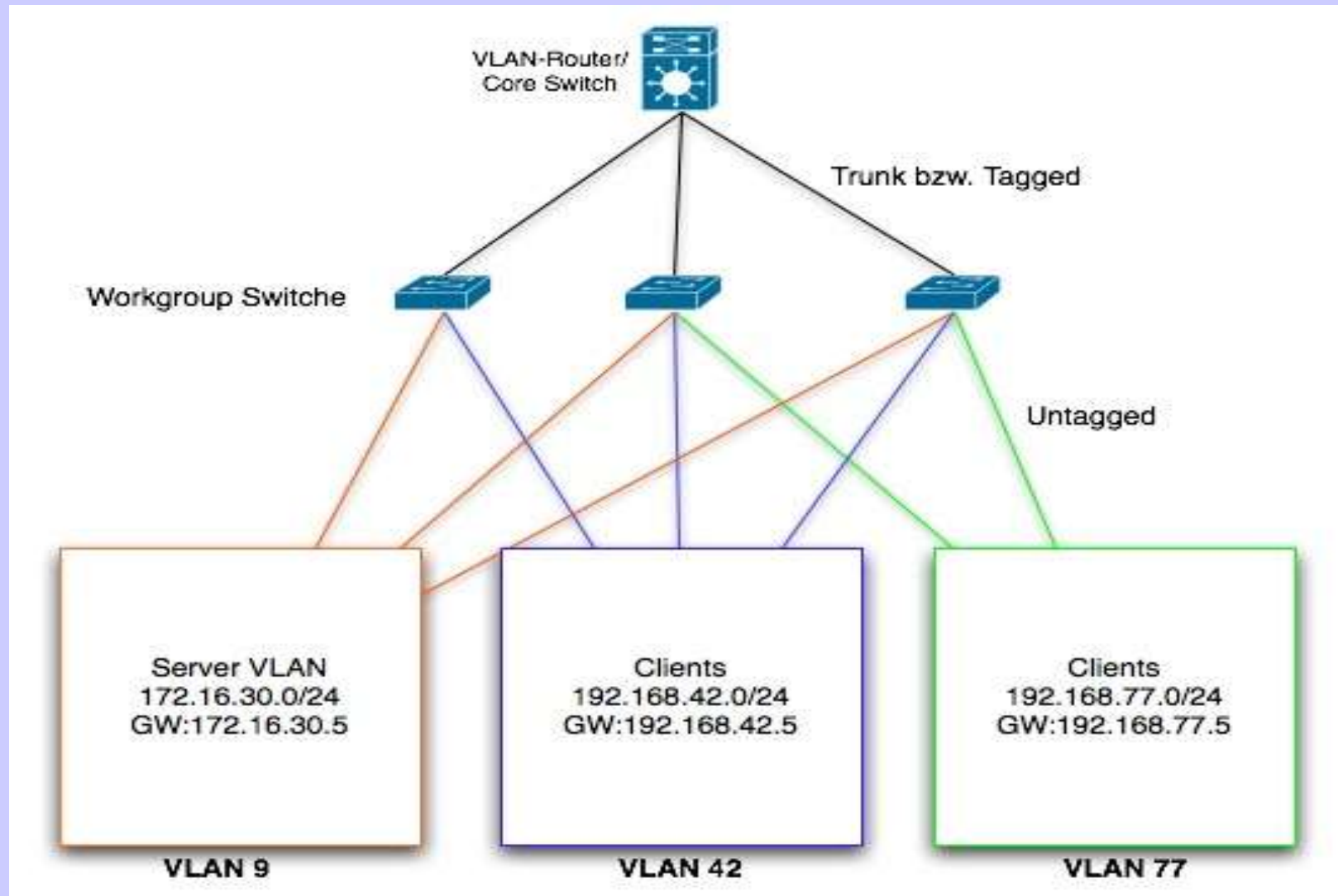
Inbetriebnahme Produktivsystem

Änderungen gegenüber dem ersten Pilotsystem:

- Da sowohl Verwaltungsmitarbeiter, wie auch Studenten an ein und demselben Switch angeschlossen sind, müssen die VLANs auf den Workgroup-Switchen konfiguriert werden
- Die VLAN-Informationen werden über das VLAN Trunk Protocol (VTP) zwischen den Cisco Switchen bekannt gemacht
- Als Backup-System wird ein identischer VLAN-Router unter Linux im **Cold Standby** betrieben. Über ein separates Netzwerkinterface werden Änderungen mit einem **rsync-Script** abgeglichen

Inbetriebnahme Produktivsystem

Topologie des Produktivsystems:



Inbetriebnahme Produktivsystem

Anpassungen und Ergebnisse:

- Erst im Produktivnetz stellte sich heraus, dass Anmelde-skripte verwendet werden die sich zu Netzwerkressourcen über den Namen verbinden, z.B.

\\Rechner-47\Freigabe

- Auch der verwendete Antiviren-Update-Server arbeitet über die NetBIOS-Namen
- Da es an dem Standort weder ein DNS-Konzept noch einen DNS-Server gibt, wurde unter Samba ein WINS-Server aktiviert

Inbetriebnahme Produktivsystem

Anpassungen und Ergebnisse:

- In Zusammenarbeit mit Windows XP Clients kommt es in der WINS-Datenbank leider zu Inkonsistenzen, da die XP-Clients in der Standardinstallation kein explizites Release ihrer alten IP-Adresse an den WINS-Server senden
- Der WINS-Dienst wurde mittlerweile durch einen Dynamischen DNS (DDNS) auf dem VLAN-Router ersetzt. Dabei registriert der DHCP-Dienst seine Adressvergabe im DNS.
- Alle anderen Komponenten funktionieren auch mit hunderten Clients problemlos

Reglementierung Internetzugriff

- Idee: Der Dozent einer Lehrveranstaltung soll über ein Web-Frontend den Zugriff der Studenten auf Seiten im Internet sperren bzw. freischalten können
- Zur Realisierung wird der Squid Proxy-Server verwendet
- Die Sperrung erfolgt auf Netzeben (Net-ID). Damit es keine Einschränkung für den Dozenten gibt, wird dieser über einen separaten Proxy geleitet. Dieser ist für die Studenten per Firewall-Regel verboten.

Reglementierung Internetzugriff

Konfiguration Squid Proxy-Server:

- Neue Access Control List (ACL):

```
acl gesperrte_netze  
    src „/etc/squid/gesperrte_netze.txt“
```

- Neue http_access Anweisung:

```
http_access deny gesperrte_netze
```

Über das Web-Interface werden nun Skripte angesprochen, die die Datei geperrte_netze.txt bearbeiten und die Konfiguration des Squid neu laden (reload).

Reglementierung Internetzugriff

Web-Frontend für Squid aus Sicht des Dozenten:

Raum EDV06

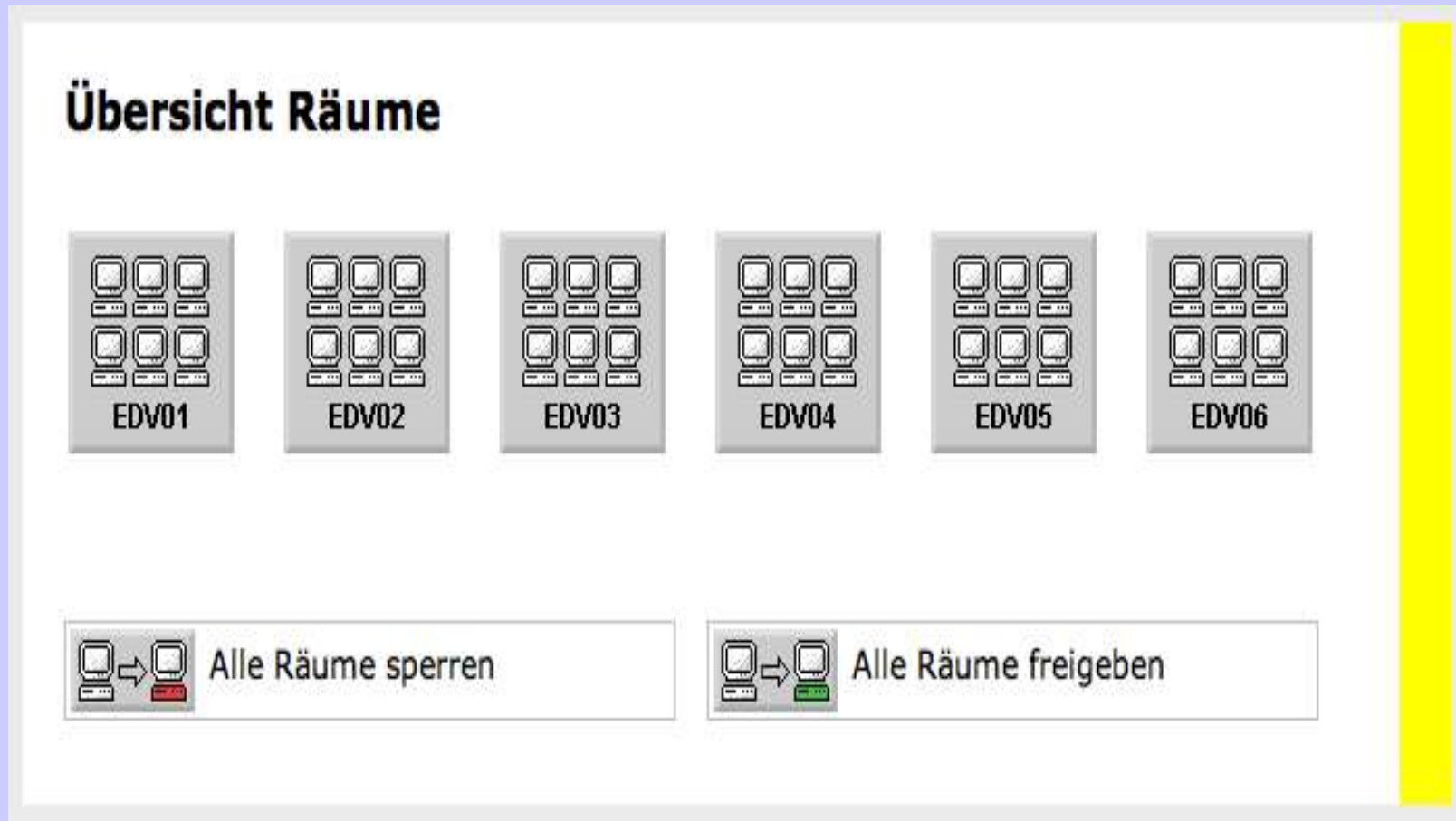
 WS113	 WS114	 WS115
 WS116	 WS117	 WS118
 WS119	 WS120	 WS121
 WS122	 WS123	 WS124

  → Alle Rechner sperren

  → Alle Rechner freigeben

Reglementierung Internetzugriff

Web-Frontend für Squid aus Sicht der Administratoren:



Ausblick

- Verbesserung des Sicherheitskonzepts
 - Keine MAC-basierten Firewall-Rules
 - Einsatz von VPN für Verwaltungsclients
 - Konifguration der Switche mit weiteren Optionen, wie z.B. DHCP Snooping

- Vereinfachung des Redundanz-Konzepts:

Cold Standby → Failover Lösung

***Vielen Dank für Eure
Aufmerksamkeit***

Fragen?

Download des Vortrags auf www.warp9.de