LAN-Kommunikation zwischen Hosts, die ihre öffentlichen IP-Adressen hinter einer ASA suchen

Inhalt

Einführung

In diesem Dokument werden verschiedene Netzwerkimplementierungen beschrieben, von denen es erforderlich ist, eine LAN-Kommunikation zwischen Hosts zuzulassen, die nach ihren öffentlichen IP-Adressen hinter einer Adaptive Security Appliance (ASA) suchen.

Voraussetzungen

Anforderungen

Cisco empfiehlt, über Kenntnisse in folgenden Bereichen zu verfügen:

- Cisco ASA NAT-Basiskonfiguration, Version 8.3 und höher
- Cisco ASA NAT-Basiskonfiguration, Version 8.2 und älter.

Verwendete Komponenten

Die Informationen in diesem Dokument basieren auf den folgenden Software- und Hardwareversionen:

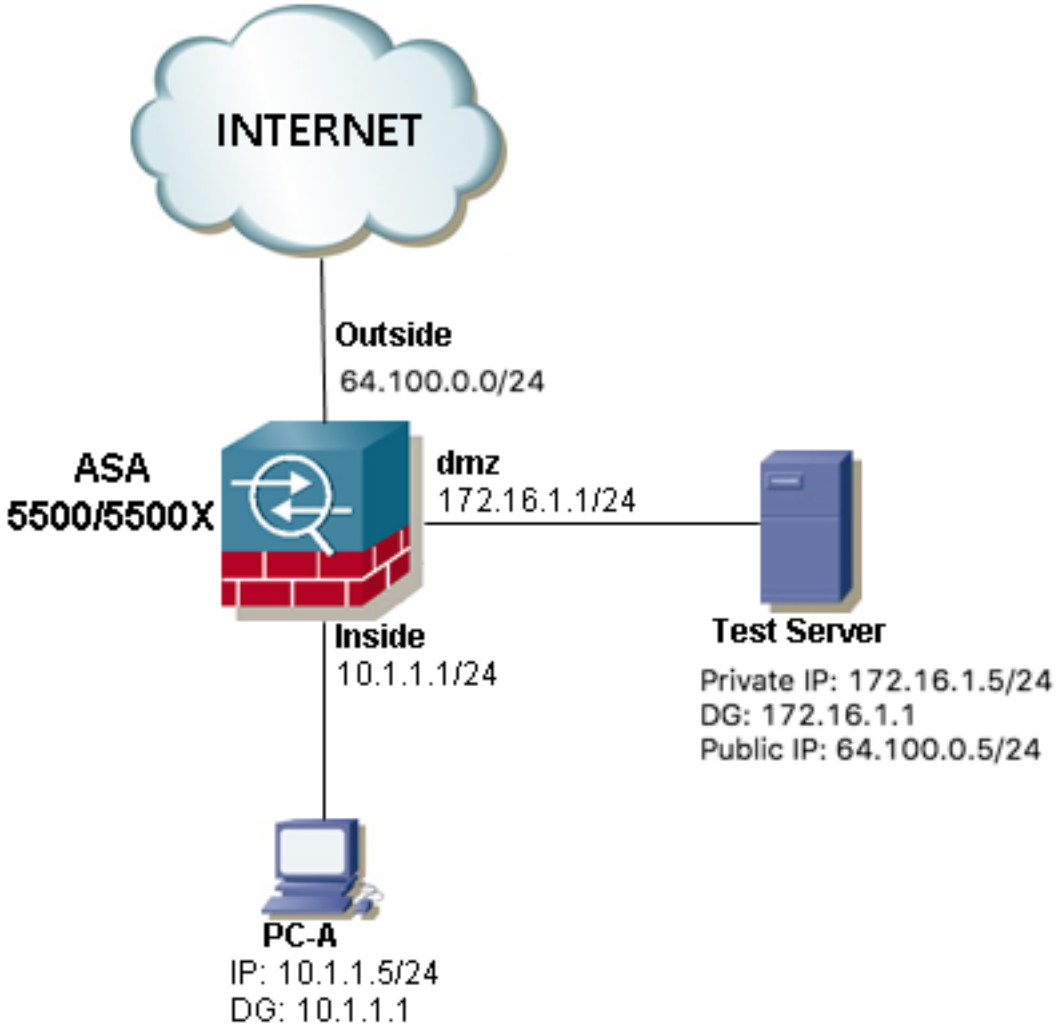
- Serie ASA 5500 und ASA 5500-X.
- Cisco ASA Version 8.3 und höher
- Cisco ASA ab Version 8.2

Die Informationen in diesem Dokument wurden von den Geräten in einer bestimmten Laborumgebung erstellt. Alle in diesem Dokument verwendeten Geräte haben mit einer leeren (Standard-)Konfiguration begonnen. Wenn Ihr Netzwerk in Betrieb ist, stellen Sie sicher, dass Sie die potenziellen Auswirkungen eines Befehls verstehen.

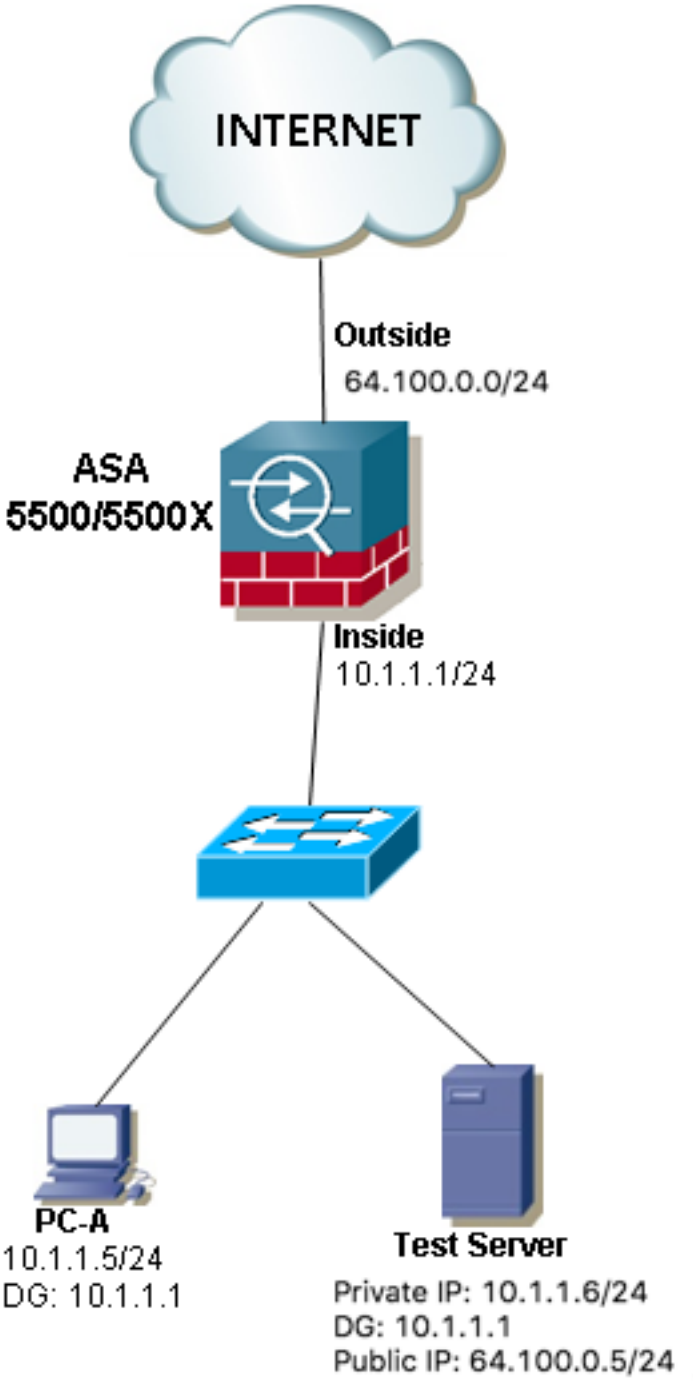
Problem: LAN-Kommunikation zwischen Hosts, die ihre öffentlichen IP-Adressen hinter einer ASA suchen

Im nächsten Abschnitt sehen Sie drei Topologiebeispiele, die diese Kommunikationsanforderung veranschaulichen, um die LAN-Kommunikation zwischen Hosts zu ermöglichen, die nach ihren öffentlichen IP-Adressen hinter einer ASA suchen.

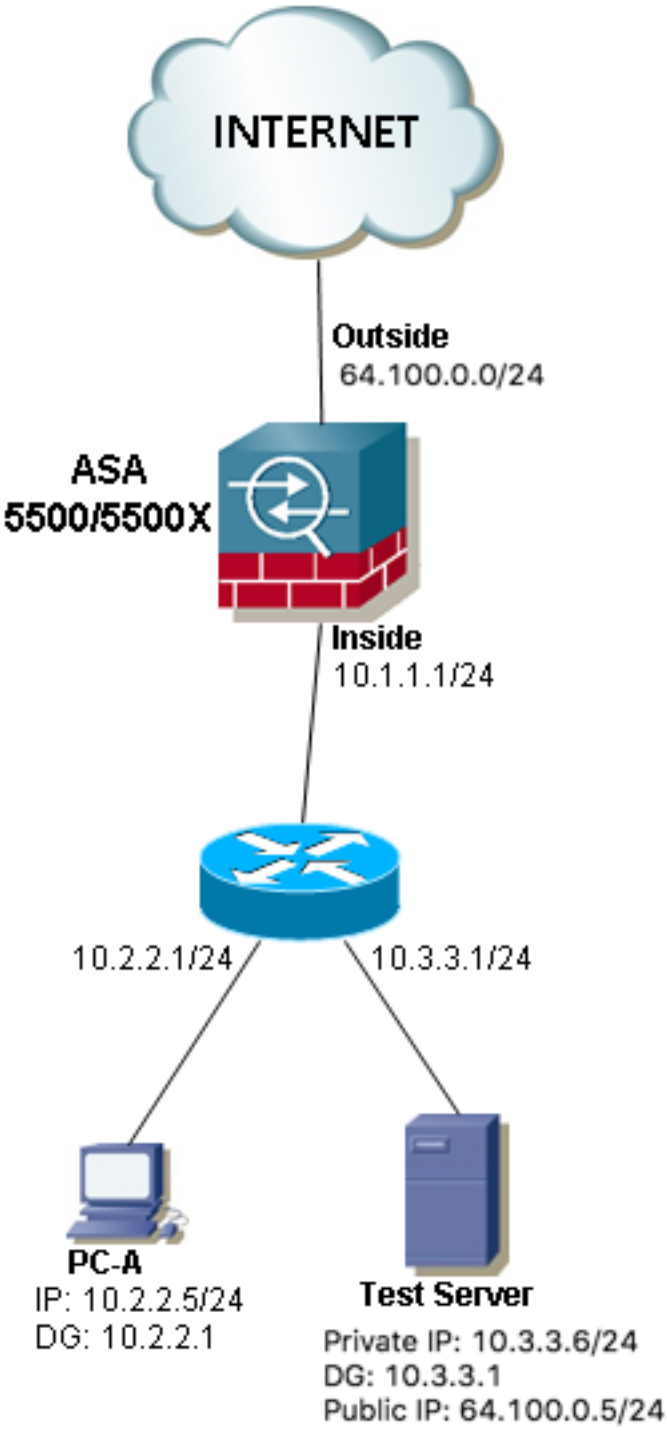
Beispiel 1. Der Quell-Host PC-A ist mit der internen ASA-Schnittstelle verbunden, während der Ziel-Host-Test-Server mit der DMZ-Schnittstelle verbunden ist.



Beispiel 2. Die Quell- und Ziel-Hosts PC-A und Test Server sind mit derselben ASA-Schnittstelle verbunden.



Beispiel 3. Die Quell- und Ziel-Hosts PC-A und Test Server sind mit der internen ASA-Schnittstelle, aber hinter einem anderen Layer-3-Gerät (Router oder Multilayer-Switch) verbunden.



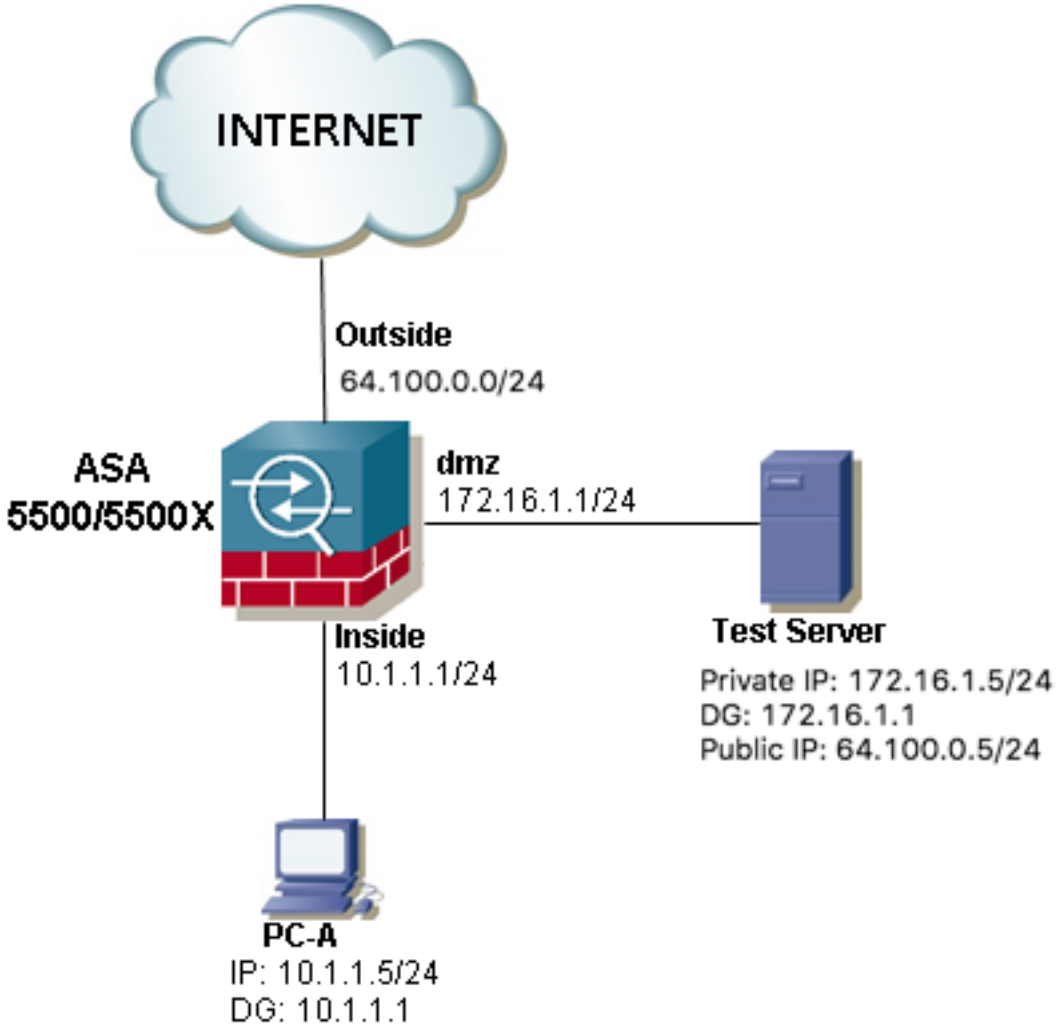
Hinweis: Der **Testserver** in den drei Images verfügt über eine statische Network Address Translation (NAT), die in der ASA konfiguriert ist. Diese statische NAT-Übersetzung wird von außen auf die entsprechende interne Schnittstelle angewendet, damit der **Testserver** von außen mit der öffentlichen IP-Adresse 64.100.0.5 erreichbar ist. Anschließend wird diese in die interne IP-Adresse des **Testservers** übertragen.

Lösung

Damit der Quell-Host PC-A den Ziel-Test-Server mit seiner öffentlichen IP-Adresse und nicht mit der privaten erreichen kann, muss eine Konfiguration mit doppelter NAT angewendet werden. Mit der Konfiguration der doppelten NAT können wir sowohl die Quell- als auch die Ziel-IP-Adresse der Pakete übersetzen, wenn der Datenverkehr die ASA passiert.

Hier finden Sie die Details der Konfiguration mit doppelter Nat für jede Topologie:

Beispiel 1. Der Quell-Host PC-A ist mit der internen ASA-Schnittstelle verbunden, während der Ziel-Host-Test-Server mit der DMZ-Schnittstelle verbunden ist.



Konfiguration

Zweifache NAT für ASA Version 8.3 und höher:

```
object network obj-10.1.1.5
host 10.1.1.5

object network obj-172.16.1.5
host 172.16.1.5

object network obj-64.100.0.5
host 64.100.0.5

nat (inside,dmz) source static obj-10.1.1.5 interface destination static obj-64.100.0.5 obj-
172.16.1.5
```

NOTE: After this NAT is applied in the ASA you will receive a warning message as the following:

```
WARNING: All traffic destined to the IP address of the outside interface is being redirected.
```

```
WARNING: Users may not be able to access any service enabled on the outside interface.
```

Zweifache NAT für ASA Version 8.2 und älter:

```
access-list IN-DMZ-INTERFACE extended permit ip host 10.1.1.5 host 64.100.0.5
static (inside,dmz) interface access-list IN-DMZ-INTERFACE

access-list DMZ-IN-INTERFACE extended permit ip host 172.16.1.5 host 172.16.1.1
static (dmz,inside) 64.100.0.5 access-list DMZ-IN-INTERFACE
```

Fehlerbehebung

Packet Tracer-Ausgabeversionen 8.3 und höher:

```
ASA# packet-tracer input inside tcp 10.1.1.5 123 64.100.0.5 80

Phase: 1
Type: ACCESS-LIST
Subtype:
Result: ALLOW
Config:
Implicit Rule
Additional Information:
MAC Access list

Phase: 2
Type: UN-NAT
Subtype: static
Result: ALLOW
Config:
nat (inside,dmz) source static obj-10.1.1.5 interface destination static obj-64.100.0.5 obj-
172.16.1.5
Additional Information:
NAT divert to egress interface dmz
Untranslate 64.100.0.5/80 to 172.16.1.5/80

Phase: 3
Type: NAT
Subtype:
Result: ALLOW
Config:
nat (inside,dmz) source static obj-10.1.1.5 interface destination static obj-64.100.0.5 obj-
172.16.1.5
Additional Information:
Static translate 10.1.1.5/123 to 172.16.1.1/123

Phase: 4
Type: NAT
Subtype: per-session
Result: ALLOW
Config:
Additional Information:

Phase: 5
Type: IP-OPTIONS
Subtype:
Result: ALLOW
Config:
Additional Information:

Phase: 6
```

```
Type: NAT
Subtype: rpf-check
Result: ALLOW
Config:
nat (inside,dmz) source static obj-10.1.1.5 interface destination static obj-64.100.0.5 obj-
172.16.1.5
Additional Information:

Phase: 7
Type: NAT
Subtype: per-session
Result: ALLOW
Config:
Additional Information:

Phase: 8
Type: IP-OPTIONS
Subtype:
Result: ALLOW
Config:
Additional Information:

Phase: 9
Type: FLOW-CREATION
Subtype:
Result: ALLOW
Config:
Additional Information:
New flow created with id 167632, packet dispatched to next module

Result:
input-interface: inside
input-status: up
input-line-status: up
output-interface: dmz
output-status: up
output-line-status: up
Action: allow
```

Packet Tracer-Ausgabeversionen 8.2 und älter:

```
ASA#packet-tracer input inside tcp 10.1.1.5 123 64.100.0.5 80

Phase: 1
Type: UN-NAT
Subtype: static
Result: ALLOW
Config:
static (dmz,inside) 64.100.0.5 access-list DMZ-IN-INTERFACE
match ip dmz host 172.16.1.5 inside host 172.16.1.1
static translation to 64.100.0.5
translate_hits = 0, untranslate_hits = 1
Additional Information:
NAT divert to egress interface dmz
Untranslate 64.100.0.5/0 to 172.16.1.5/0 using netmask 255.255.255.255

Phase: 2
Type: IP-OPTIONS
Subtype:
Result: ALLOW
Config:
Additional Information:
```

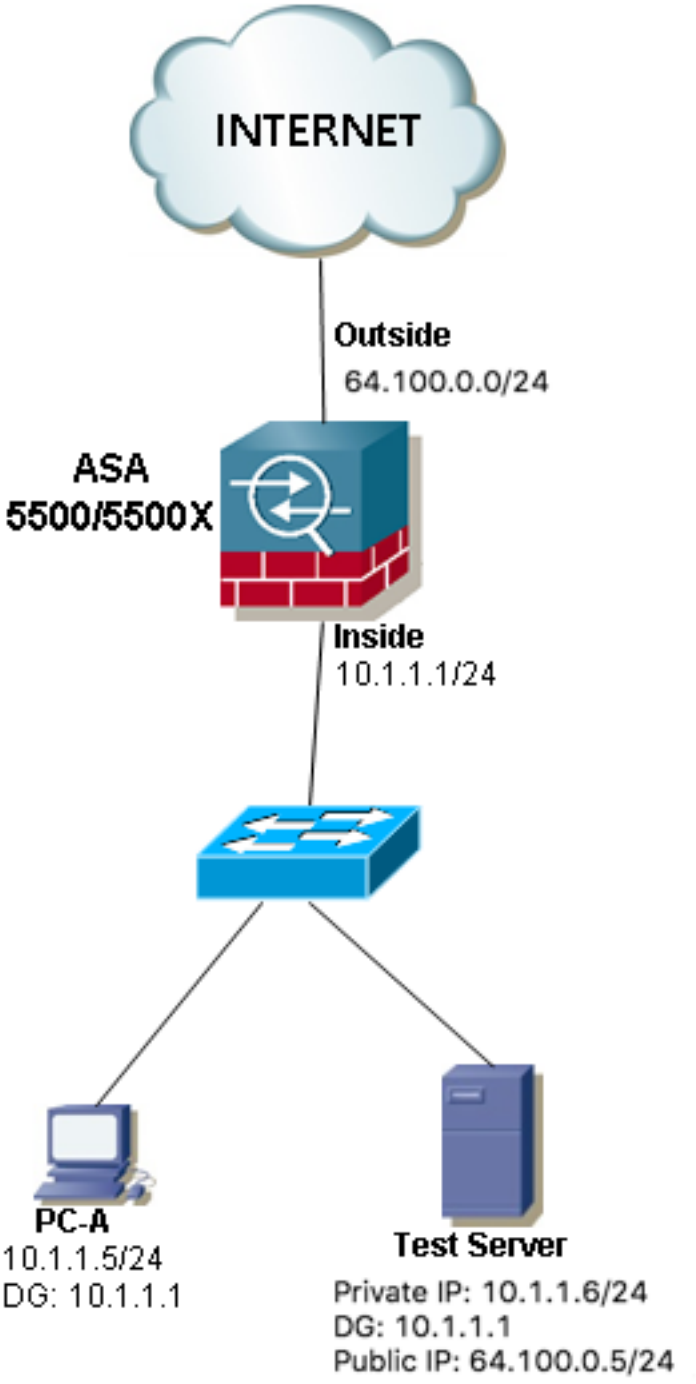
```
Phase: 3
Type: NAT
Subtype:
Result: ALLOW
Config:
static (inside,dmz) interface access-list IN-DMZ-INTERFACE
match ip inside host 10.1.1.5 dmz host 64.100.0.5
static translation to 172.16.1.1
translate_hits = 1, untranslate_hits = 0
Additional Information:
Static translate 10.1.1.5/0 to 172.16.1.1/0 using netmask 255.255.255.255

Phase: 4
Type: NAT
Subtype: host-limits
Result: ALLOW
Config:
static (inside,dmz) interface access-list IN-DMZ-INTERFACE
match ip inside host 10.1.1.5 dmz host 64.100.0.5
static translation to 172.16.1.1
translate_hits = 1, untranslate_hits = 0
Additional Information:

Phase: 5
Type: NAT
Subtype: rpf-check
Result: ALLOW
Config:
static (dmz,inside) 64.100.0.5 access-list DMZ-IN-INTERFACE
match ip dmz host 172.16.1.5 inside host 172.16.1.1
static translation to 64.100.0.5
translate_hits = 0, untranslate_hits = 1
Additional Information:

Phase: 6
Type: NAT
Subtype: host-limits
Result: ALLOW
Config:
static (dmz,inside) 64.100.0.5 access-list DMZ-IN-INTERFACE
match ip dmz host 172.16.1.5 inside host 172.16.1.1
static translation to 64.100.0.5
translate_hits = 0, untranslate_hits = 1
Additional Information:

Phase: 7
Type: IP-OPTIONS
Subtype:
Result: ALLOW
Config:
Additional Information:

Phase: 8
Type: FLOW-CREATION
Subtype:
Result: ALLOW
Config:
Additional Information:
New flow created with id 503, packet dispatched to next module

Result:
input-interface: inside
input-status: up
input-line-status: up
```

```
output-interface: dmz
output-status: up
output-line-status: up
Action: allow
```

Paketerfassung:

```
ASA# sh cap
capture capin type raw-data interface inside [Capturing - 1300 bytes]
match ip host 10.1.1.5 host 64.100.0.5
capture capout type raw-data interface dmz [Capturing - 1300 bytes]
match ip host 172.16.1.1 host 172.16.1.5

ASA# sh cap capin

10 packets captured
1: 12:36:28.245455 10.1.1.5 > 64.100.0.5: icmp: echo request
2: 12:36:28.269441 64.100.0.5 > 10.1.1.5: icmp: echo reply
3: 12:36:28.303451 10.1.1.5 > 64.100.0.5: icmp: echo request
4: 12:36:28.333692 64.100.0.5 > 10.1.1.5: icmp: echo reply
5: 12:36:28.372478 10.1.1.5 > 64.100.0.5: icmp: echo request
6: 12:36:28.395563 64.100.0.5 > 10.1.1.5: icmp: echo reply
7: 12:36:28.422402 10.1.1.5 > 64.100.0.5: icmp: echo request
8: 12:36:28.449241 64.100.0.5 > 10.1.1.5: icmp: echo reply
9: 12:36:28.481420 10.1.1.5 > 64.100.0.5: icmp: echo request
10: 12:36:28.507435 64.100.0.5 > 10.1.1.5: icmp: echo reply
10 packets shown

ASA1# sh cap capout

10 packets captured
1: 12:36:28.245730 172.16.1.1 > 172.16.1.5: icmp: echo request
2: 12:36:28.269395 172.16.1.5 > 172.16.1.1: icmp: echo reply
3: 12:36:28.303725 172.16.1.1 > 172.16.1.5: icmp: echo request
4: 12:36:28.333646 172.16.1.5 > 172.16.1.1: icmp: echo reply
5: 12:36:28.372737 172.16.1.1 > 172.16.1.5: icmp: echo request
6: 12:36:28.395533 172.16.1.5 > 172.16.1.1: icmp: echo reply
7: 12:36:28.422661 172.16.1.1 > 172.16.1.5: icmp: echo request
8: 12:36:28.449195 172.16.1.5 > 172.16.1.1: icmp: echo reply
9: 12:36:28.481695 172.16.1.1 > 172.16.1.5: icmp: echo request
10: 12:36:28.507404 172.16.1.5 > 172.16.1.1: icmp: echo reply
10 packets shown
```

Beispiel 2. Die Quell- und Ziel-Hosts PC-A und Test Server sind mit derselben ASA-Schnittstelle verbunden.



Konfiguration

Zweifache NAT für ASA Version 8.3 und höher:

```
object network obj-10.1.1.5
host 10.1.1.5

object network obj-10.1.1.6
host 10.1.1.6

object network obj-64.100.0.5
host 64.100.0.5

nat (inside,inside) source static obj-10.1.1.5 interface destination static obj-64.100.0.5 obj-
10.1.1.6
```

NOTE: After this NAT is applied in the ASA you will receive a warning message as the following:

```
WARNING: All traffic destined to the IP address of the outside interface is being redirected.
WARNING: Users may not be able to access any service enabled on the outside interface.
```

Zweifache NAT für ASA Version 8.2 und älter:

```
access-list IN-OUT-INTERFACE extended permit ip host 10.1.1.5 host 64.100.0.5
static (inside,inside) interface access-list IN-OUT-INTERFACE

access-list OUT-IN-INTERFACE extended permit ip host 10.1.1.6 host 10.1.1.1
static (inside,inside) 64.100.0.5 access-list OUT-IN-INTERFACE
```

> **Hinweis**: Die Hauptzweck der NAT-Übersetzung für die Quell-IP-Adresse von 10.1.1.5 zur ASA-Inside-IP-Adresse 10.1.1.1 besteht darin, die Antworten von Host 10.1.1.6 dazu zu zwingen, an die ASA zurückzukehren. Dies ist besonders erforderlich, um asymmetrisches Routing zu vermeiden und der ASA die Verarbeitung des gesamten Datenverkehrs zwischen den interessierten Hosts zu ermöglichen. Geben Sie die Quell-IP-Adresse wie in diesem Beispiel an, und dann blockiert die ASA den betroffenen Datenverkehr aufgrund von asymmetrischem Routing.

Fehlerbehebung

Packet Tracer-Ausgabeversionen 8.3 und höher:

```
ASA# packet-tracer input inside tcp 10.1.1.5 123 64.100.0.5 80

Phase: 1
Type: UN-NAT
Subtype: static
Result: ALLOW
Config:
nat (inside,inside) source static obj-10.1.1.5 interface destination static obj-64.100.0.5 obj-
10.1.1.6
Additional Information:
NAT divert to egress interface inside
Untranslate 64.100.0.5/80 to 10.1.1.6/80

Phase: 2
Type: NAT
Subtype:
Result: ALLOW
Config:
nat (inside,inside) source static obj-10.1.1.5 interface destination static obj-64.100.0.5 obj-
10.1.1.6
Additional Information:
Static translate 10.1.1.5/123 to 10.1.1.1/123

Phase: 3
Type: ACCESS-LIST
Subtype:
Result: ALLOW
Config:
Implicit Rule
Additional Information:

Phase: 4
Type: NAT
```

```
Subtype: per-session
Result: ALLOW
Config:
Additional Information:

Phase: 5
Type: IP-OPTIONS
Subtype:
Result: ALLOW
Config:
Additional Information:

Phase: 6
Type: NAT
Subtype: rpf-check
Result: ALLOW
Config:
nat (inside,inside) source static obj-10.1.1.5 interface destination static obj-64.100.0.5 obj-
10.1.1.6
Additional Information:

Phase: 7
Type: NAT
Subtype: per-session
Result: ALLOW
Config:
Additional Information:

Phase: 8
Type: IP-OPTIONS
Subtype:
Result: ALLOW
Config:
Additional Information:

Phase: 9
Type: FLOW-CREATION
Subtype:
Result: ALLOW
Config:
Additional Information:
New flow created with id 167839, packet dispatched to next module

Result:
input-interface: inside
input-status: up
input-line-status: up
output-interface: inside
output-status: up
output-line-status: up
Action: allow
```

Packet Tracer-Ausgabeversionen 8.2 und älter:

```
ASA# packet-tracer input inside tcp 10.1.1.5 123 64.100.0.5 80

Phase: 1
Type: UN-NAT
Subtype: static
Result: ALLOW
Config:
static (inside,inside) 64.100.0.5 access-list OUT-IN-INTERFACE
match ip inside host 10.1.1.6 inside host 10.1.1.1
```

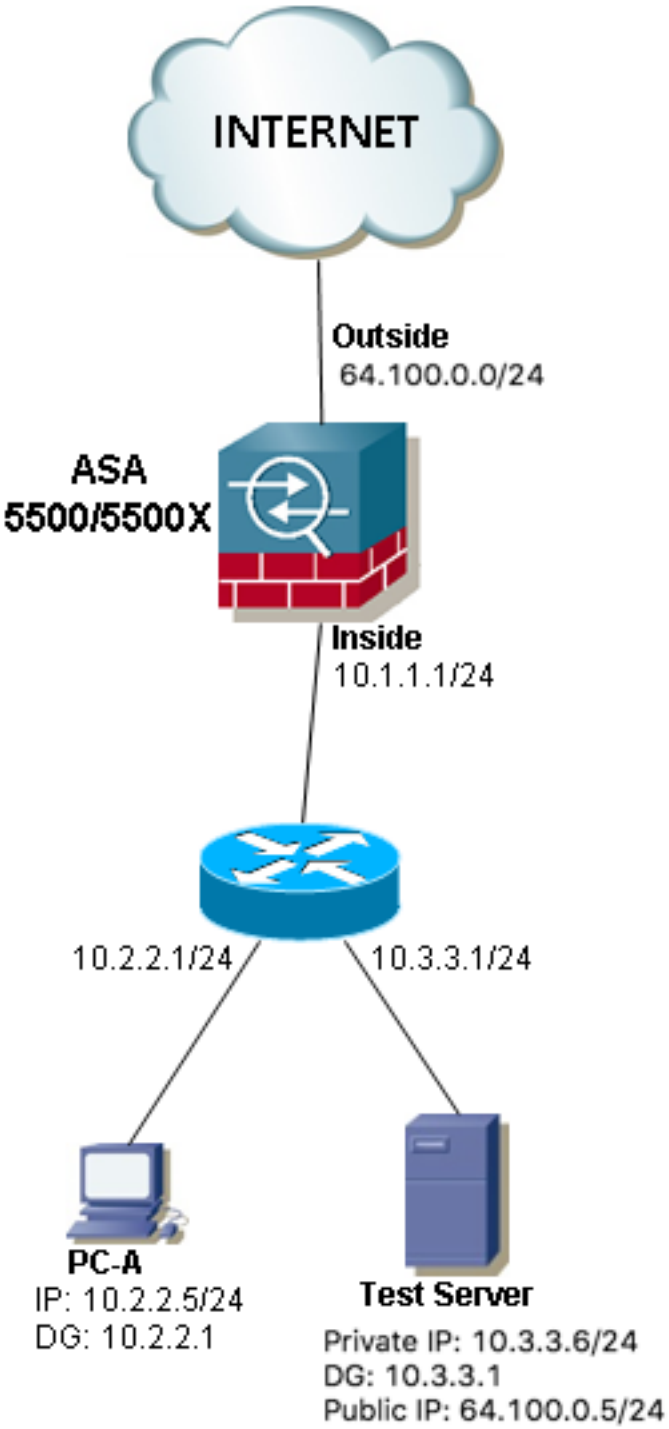
```
static translation to 64.100.0.5
translate_hits = 0, untranslate_hits = 1
Additional Information:
NAT divert to egress interface inside
Untranslate 64.100.0.5/0 to 10.1.1.6/0 using netmask 255.255.255.255

Phase: 2
Type: ACCESS-LIST
Subtype:
Result: ALLOW
Config:
Implicit Rule
Additional Information:

Phase: 3
Type: IP-OPTIONS
Subtype:
Result: ALLOW
Config:
Additional Information:

Phase: 4
Type: NAT
Subtype:
Result: ALLOW
Config:
static (inside,inside) interface access-list IN-OUT-INTERFACE
match ip inside host 10.1.1.5 inside host 64.100.0.5
static translation to 10.1.1.1
translate_hits = 1, untranslate_hits = 0
Additional Information:
Static translate 10.1.1.5/0 to 10.1.1.1/0 using netmask 255.255.255.255

Phase: 5
Type: NAT
Subtype: host-limits
Result: ALLOW
Config:
static (inside,inside) interface access-list IN-OUT-INTERFACE
match ip inside host 10.1.1.5 inside host 64.100.0.5
static translation to 10.1.1.1
translate_hits = 1, untranslate_hits = 0
Additional Information:

Phase: 6
Type: NAT
Subtype: rpf-check
Result: ALLOW
Config:
static (inside,inside) 64.100.0.5 access-list OUT-IN-INTERFACE
match ip inside host 10.1.1.6 inside host 10.1.1.1
static translation to 64.100.0.5
translate_hits = 0, untranslate_hits = 1
Additional Information:

Phase: 7
Type: NAT
Subtype: host-limits
Result: ALLOW
Config:
static (inside,inside) 64.100.0.5 access-list OUT-IN-INTERFACE
match ip inside host 10.1.1.6 inside host 10.1.1.1
static translation to 64.100.0.5
translate_hits = 0, untranslate_hits = 1
```

```
Additional Information:

Phase: 8
Type: IP-OPTIONS
Subtype:
Result: ALLOW
Config:
Additional Information:

Phase: 9
Type: FLOW-CREATION
Subtype:
Result: ALLOW
Config:
Additional Information:
New flow created with id 727, packet dispatched to next module

Result:
input-interface: inside
input-status: up
input-line-status: up
output-interface: inside
output-status: up
output-line-status: up
Action: allow
```

Paketerfassung:

```
ASA# sh cap
capture capin type raw-data interface inside [Capturing - 1300 bytes]
match ip host 10.1.1.5 host 64.100.0.5
capture capout type raw-data interface inside [Capturing - 1300 bytes]
match ip host 10.1.1.1 host 10.1.1.6

ASA# sh cap capin

10 packets captured
1: 12:50:39.304748 10.1.1.5 > 64.100.0.5: icmp: echo request
2: 12:50:39.335431 64.100.0.5 > 10.1.1.5: icmp: echo reply
3: 12:50:39.368389 10.1.1.5 > 64.100.0.5: icmp: echo request
4: 12:50:39.389368 64.100.0.5 > 10.1.1.5: icmp: echo reply
5: 12:50:39.398432 10.1.1.5 > 64.100.0.5: icmp: echo request
6: 12:50:39.418176 64.100.0.5 > 10.1.1.5: icmp: echo reply
7: 12:50:39.419732 10.1.1.5 > 64.100.0.5: icmp: echo request
8: 12:50:39.425103 64.100.0.5 > 10.1.1.5: icmp: echo reply
9: 12:50:39.434395 10.1.1.5 > 64.100.0.5: icmp: echo request
10: 12:50:39.438423 64.100.0.5 > 10.1.1.5: icmp: echo reply
10 packets shown

ASA2# sh cap capout

10 packets captured
1: 12:50:39.305282 10.1.1.1 > 10.1.1.6: icmp: echo request
2: 12:50:39.335386 10.1.1.6 > 10.1.1.1: icmp: echo reply
3: 12:50:39.368663 10.1.1.1 > 10.1.1.6: icmp: echo request
4: 12:50:39.389307 10.1.1.6 > 10.1.1.1: icmp: echo reply
5: 12:50:39.398706 10.1.1.1 > 10.1.1.6: icmp: echo request
6: 12:50:39.418130 10.1.1.6 > 10.1.1.1: icmp: echo reply
7: 12:50:39.419762 10.1.1.1 > 10.1.1.6: icmp: echo request
8: 12:50:39.425072 10.1.1.6 > 10.1.1.1: icmp: echo reply
9: 12:50:39.434669 10.1.1.1 > 10.1.1.6: icmp: echo request
10: 12:50:39.438392 10.1.1.6 > 10.1.1.1: icmp: echo reply
10 packets shown
```

Beispiel 3. Die Quell- und Ziel-Hosts PC-A und Test Server sind mit der internen ASA-Schnittstelle, aber hinter einem anderen Layer-3-Gerät (Router oder Multilayer-Switch) verbunden.



Konfiguration

Zweifache NAT für ASA Version 8.3 und höher:

```
object network obj-10.2.2.5
host 10.2.2.5

object network obj-10.3.3.6
```

```
host 10.3.3.6

object network obj-64.100.0.5
host 64.100.0.5

nat (inside,inside) source static obj-10.2.2.5 interface destination static obj-64.100.0.5 obj-
10.3.3.6
```

NOTE: After this NAT is applied in the ASA you will receive a warning message as the following:

```
WARNING: All traffic destined to the IP address of the outside interface is being redirected.
WARNING: Users may not be able to access any service enabled on the outside interface.
```

Zweifache NAT für ASA Version 8.2 und älter:

```
access-list IN-OUT-INTERFACE extended permit ip host 10.2.2.5 host 64.100.0.5
static (inside,inside) interface access-list IN-OUT-INTERFACE

access-list OUT-IN-INTERFACE extended permit ip host 10.3.3.6 host 10.1.1.1
static (inside,inside) 64.100.0.5 access-list OUT-IN-INTERFACE
```

> **Hinweis**: Die Hauptzweck der NAT-Übersetzung für die Quell-IP-Adresse von 10.1.1.5 zur ASA-Inside-IP-Adresse (10.1.1.1) besteht darin, die Antworten von Host 10.1.1.6 dazu zu zwingen, an die ASA zurückzukehren, um asymmetrisches Routing zu vermeiden und der ASA die Verarbeitung des gesamten Datenverkehrs zwischen den interessierten Hosts zu ermöglichen. Wenn wir die Quell-IP-Adresse nicht wie in diesem Beispiel übersetzen, blockiert die ASA den betroffenen Datenverkehr aufgrund von asymmetrischem Routing.

Fehlerbehebung

Packet Tracer-Ausgabeversionen 8.3 und höher:

```
ASA# packet-tracer input inside tcp 10.2.2.5 123 64.100.0.5 80

Phase: 1
Type: UN-NAT
Subtype: static
Result: ALLOW
Config:
nat (inside,inside) source static obj-10.2.2.5 interface destination static obj-64.100.0.5 obj-
10.3.3.6
Additional Information:
NAT divert to egress interface inside
Untranslate 64.100.0.5/80 to 10.3.3.6/80

Phase: 2
Type: NAT
Subtype:
Result: ALLOW
Config:
nat (inside,inside) source static obj-10.2.2.5 interface destination static obj-64.100.0.5 obj-
10.3.3.6
Additional Information:
Static translate 10.2.2.5/123 to 10.1.1.1/123

Phase: 3
Type: ACCESS-LIST
Subtype:
```

```
Result: ALLOW
Config:
Implicit Rule
Additional Information:

Phase: 4
Type: NAT
Subtype: per-session
Result: ALLOW
Config:
Additional Information:

Phase: 5
Type: IP-OPTIONS
Subtype:
Result: ALLOW
Config:
Additional Information:

Phase: 6
Type: NAT
Subtype: rpf-check
Result: ALLOW
Config:
nat (inside,inside) source static obj-10.2.2.5 interface destination static obj-64.100.0.5 obj-
10.3.3.6
Additional Information:

Phase: 7
Type: NAT
Subtype: per-session
Result: ALLOW
Config:
Additional Information:

Phase: 8
Type: IP-OPTIONS
Subtype:
Result: ALLOW
Config:
Additional Information:

Phase: 9
Type: FLOW-CREATION
Subtype:
Result: ALLOW
Config:
Additional Information:
New flow created with id 167945, packet dispatched to next module

Result:
input-interface: inside
input-status: up
input-line-status: up
output-interface: inside
output-status: up
output-line-status: up
Action: allow
```

Packet Tracer-Ausgabeversionen 8.2 und älter:

```
ASA# packet-tracer input inside tcp 10.2.2.5 123 64.100.0.5 80
```

```
Phase: 1
Type: UN-NAT
Subtype: static
Result: ALLOW
Config:
static (inside,inside) 64.100.0.5 access-list OUT-IN-INTERFACE
match ip inside host 10.3.3.6 inside host 10.1.1.1
static translation to 64.100.0.5
translate_hits = 0, untranslate_hits = 1
Additional Information:
NAT divert to egress interface inside
Untranslate 64.100.0.5/0 to 10.3.3.6/0 using netmask 255.255.255.255

Phase: 2
Type: ACCESS-LIST
Subtype:
Result: ALLOW
Config:
Implicit Rule
Additional Information:

Phase: 3
Type: IP-OPTIONS
Subtype:
Result: ALLOW
Config:
Additional Information:

Phase: 4
Type: NAT
Subtype:
Result: ALLOW
Config:
static (inside,inside) interface access-list IN-OUT-INTERFACE
match ip inside host 10.2.2.5 inside host 64.100.0.5
static translation to 10.1.1.1
translate_hits = 1, untranslate_hits = 0
Additional Information:
Static translate 10.2.2.5/0 to 10.1.1.1/0 using netmask 255.255.255.255

Phase: 5
Type: NAT
Subtype: host-limits
Result: ALLOW
Config:
static (inside,inside) interface access-list IN-OUT-INTERFACE
match ip inside host 10.2.2.5 inside host 64.100.0.5
static translation to 10.1.1.1
translate_hits = 1, untranslate_hits = 0
Additional Information:

Phase: 6
Type: NAT
Subtype: rpf-check
Result: ALLOW
Config:
static (inside,inside) 64.100.0.5 access-list OUT-IN-INTERFACE
match ip inside host 10.3.3.6 inside host 10.1.1.1
static translation to 64.100.0.5
translate_hits = 0, untranslate_hits = 1
Additional Information:

Phase: 7
Type: NAT
```

```
Subtype: host-limits
Result: ALLOW
Config:
static (inside,inside) 64.100.0.5 access-list OUT-IN-INTERFACE
match ip inside host 10.3.3.6 inside host 10.1.1.1
static translation to 64.100.0.5
translate_hits = 0, untranslate_hits = 1
Additional Information:

Phase: 8
Type: IP-OPTIONS
Subtype:
Result: ALLOW
Config:
Additional Information:

Phase: 9
Type: FLOW-CREATION
Subtype:
Result: ALLOW
Config:
Additional Information:
New flow created with id 908, packet dispatched to next module

Result:
input-interface: inside
input-status: up
input-line-status: up
output-interface: inside
output-status: up
output-line-status: up
Action: allow
```

Paketerfassung:

```
ASA# sh cap
capture capin type raw-data interface inside [Capturing - 1300 bytes]
match ip host 10.2.2.5 host 64.100.0.5
capture capout type raw-data interface inside [Capturing - 1300 bytes]
match ip host 10.1.1.1 host 10.3.3.6

ASA# sh cap capin

10 packets captured
1: 13:06:09.302047 10.2.2.5 > 64.100.0.5: icmp: echo request
2: 13:06:09.315276 64.100.0.5 > 10.2.2.5: icmp: echo reply
3: 13:06:09.342221 10.2.2.5 > 64.100.0.5: icmp: echo request
4: 13:06:09.381266 64.100.0.5 > 10.2.2.5: icmp: echo reply
5: 13:06:09.421227 10.2.2.5 > 64.100.0.5: icmp: echo request
6: 13:06:09.459204 64.100.0.5 > 10.2.2.5: icmp: echo reply
7: 13:06:09.494939 10.2.2.5 > 64.100.0.5: icmp: echo request
8: 13:06:09.534258 64.100.0.5 > 10.2.2.5: icmp: echo reply
9: 13:06:09.564210 10.2.2.5 > 64.100.0.5: icmp: echo request
10: 13:06:09.593261 64.100.0.5 > 10.2.2.5: icmp: echo reply
10 packets shown

ASA# sh cap capout

10 packets captured
1: 13:06:09.302367 10.1.1.1 > 10.3.3.6: icmp: echo request
2: 13:06:09.315230 10.3.3.6 > 10.1.1.1: icmp: echo reply
3: 13:06:09.342526 10.1.1.1 > 10.3.3.6: icmp: echo request
4: 13:06:09.381221 10.3.3.6 > 10.1.1.1: icmp: echo reply
```

```
5: 13:06:09.421517 10.1.1.1 > 10.3.3.6: icmp: echo request
6: 13:06:09.459174 10.3.3.6 > 10.1.1.1: icmp: echo reply
7: 13:06:09.495244 10.1.1.1 > 10.3.3.6: icmp: echo request
8: 13:06:09.534213 10.3.3.6 > 10.1.1.1: icmp: echo reply
9: 13:06:09.564500 10.1.1.1 > 10.3.3.6: icmp: echo request
10: 13:06:09.593215 10.3.3.6 > 10.1.1.1: icmp: echo reply
10 packets shown
```

Zugehörige Informationen

- ASA 8.3 Konfigurationsleitfaden: Voraussetzung für doppelte NAT
- ASA 8.4-Konfigurationsleitfaden: DNS und NAT
- ASA-Konfigurationsbeispiele für die Versionen vor 8.3 bis 8.3 NAT