Comunicazione LAN tra gli host che ricercano gli indirizzi IP pubblici dietro un'ASA

Sommario

Introduzione

In questo documento vengono descritte diverse implementazioni di rete da cui è necessario consentire la comunicazione LAN (Local Area Network) tra gli host che ricercano i propri indirizzi IP pubblici su un'appliance ASA (Adaptive Security Appliance).

Prerequisiti

Requisiti

Cisco raccomanda la conoscenza dei seguenti argomenti:

- Configurazione Cisco ASA NAT di base, versione 8.3 e successive.
- Configurazione Cisco ASA NAT di base, versione 8.2 e precedenti.

Componenti usati

Le informazioni fornite in questo documento si basano sulle seguenti versioni software e hardware:

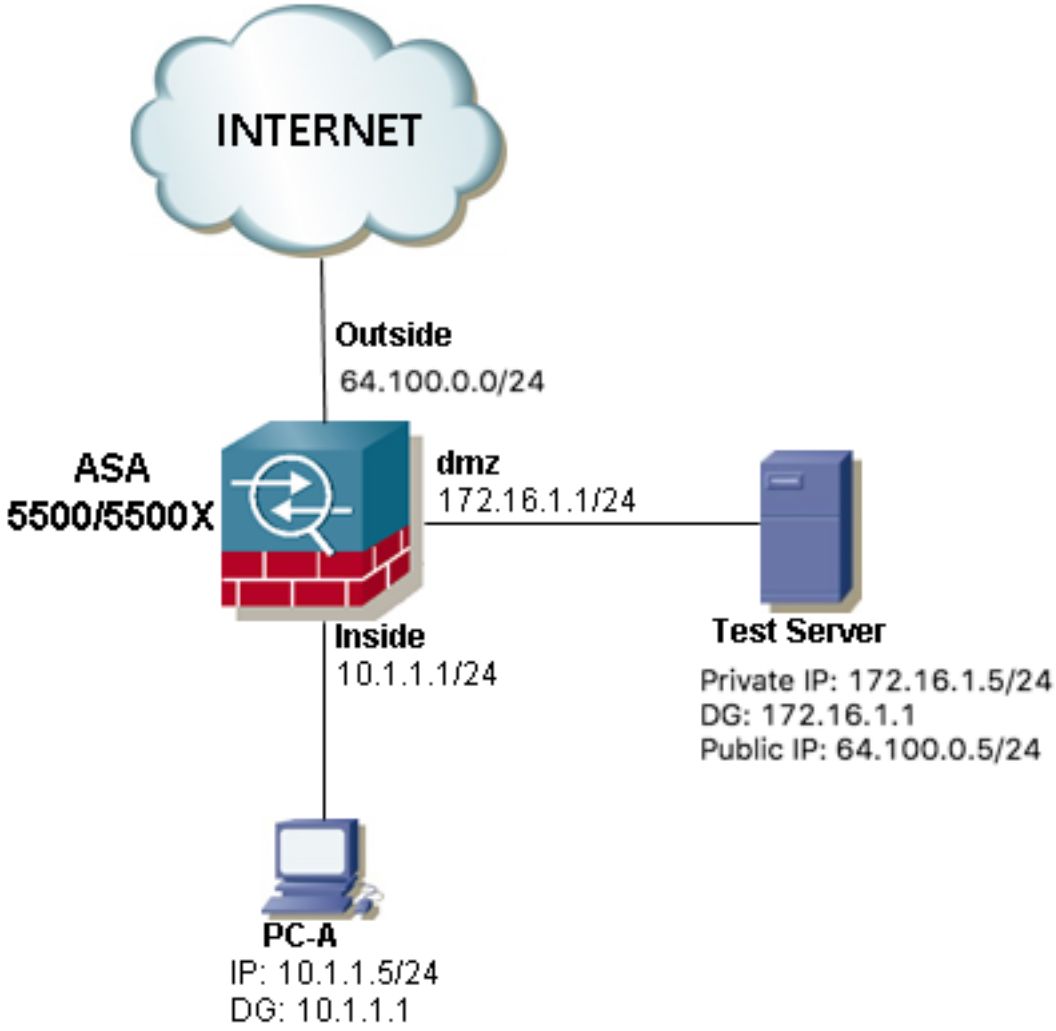
- Serie ASA5500 e ASA5500-X.
- Cisco ASA versione 8.3 e successive.
- Cisco ASA versione 8.2 e precedenti.

Le informazioni discusse in questo documento fanno riferimento a dispositivi usati in uno specifico ambiente di emulazione. Su tutti i dispositivi menzionati nel documento la configurazione è stata ripristinata ai valori predefiniti. Se la rete è operativa, valutare attentamente eventuali conseguenze derivanti dall'uso dei comandi.

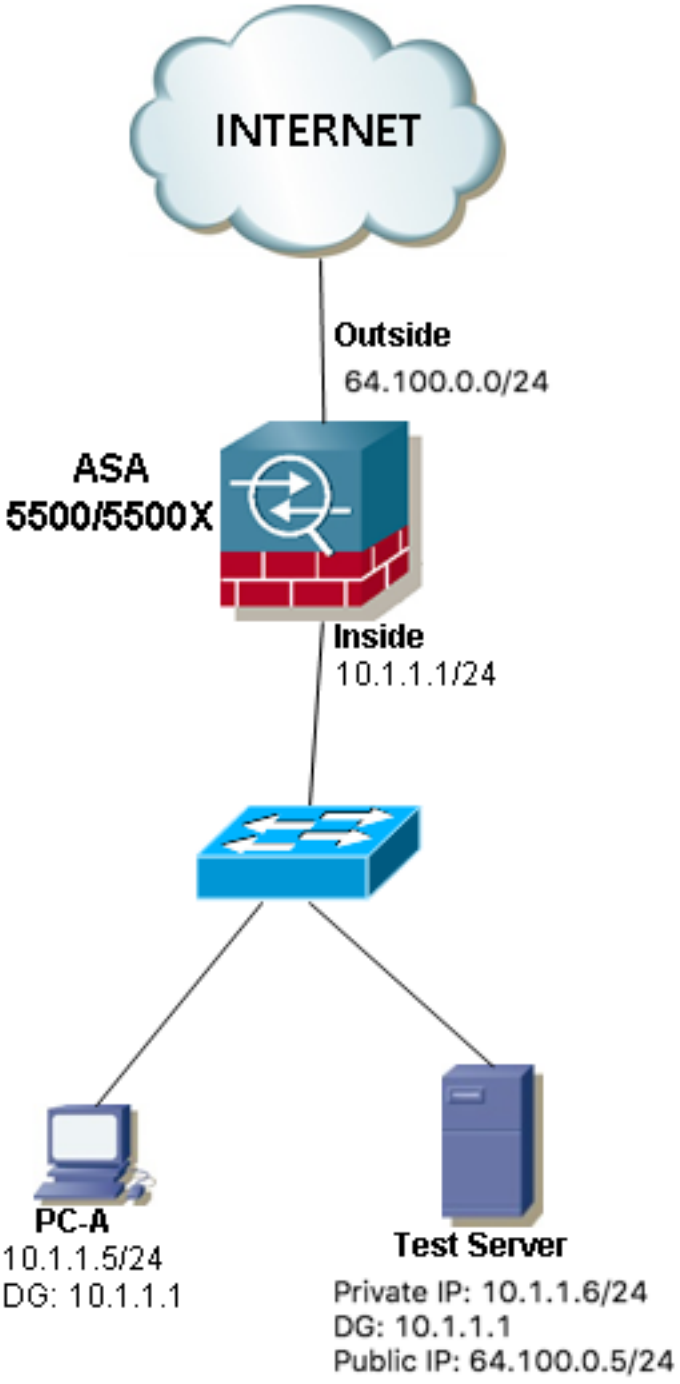
Problema: Comunicazione LAN tra host che ricercano i propri indirizzi IP pubblici dietro un'ASA

Nella sezione successiva, sono riportati tre esempi di topologia che mostrano i requisiti di comunicazione per consentire la comunicazione LAN tra gli host che cercano gli indirizzi IP pubblici su un'appliance ASA.

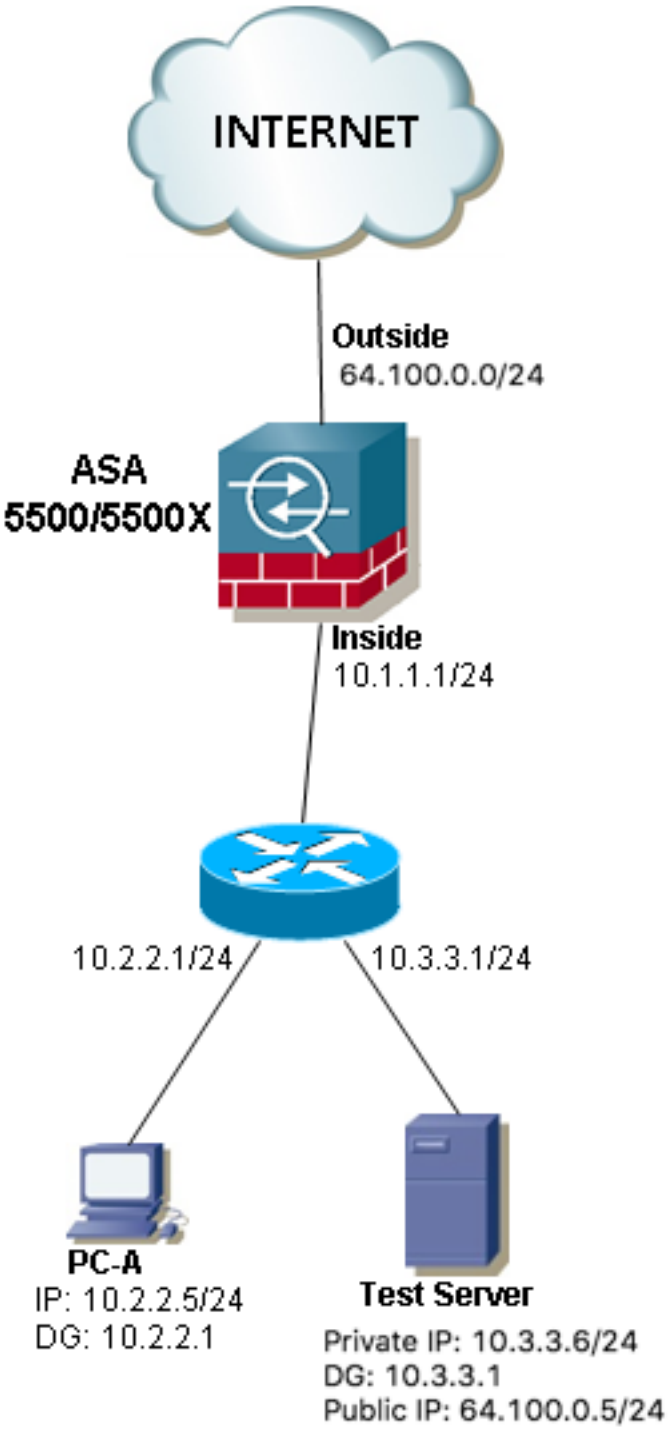
Esempio 1. L'host di origine PC-A è connesso all'interfaccia ASA interna, mentre l'host di destinazione Test Server è connesso all'interfaccia DMZ.



Esempio 2. Gli host di origine e destinazione PC-A e Test Server sono connessi alla stessa interfaccia ASA.



Esempio 3. L'host di origine e di destinazione PC-A e il server di prova sono collegati all'interfaccia ASA interna, ma dietro un altro dispositivo di livello 3 (potrebbe essere un router o uno switch multilivello).



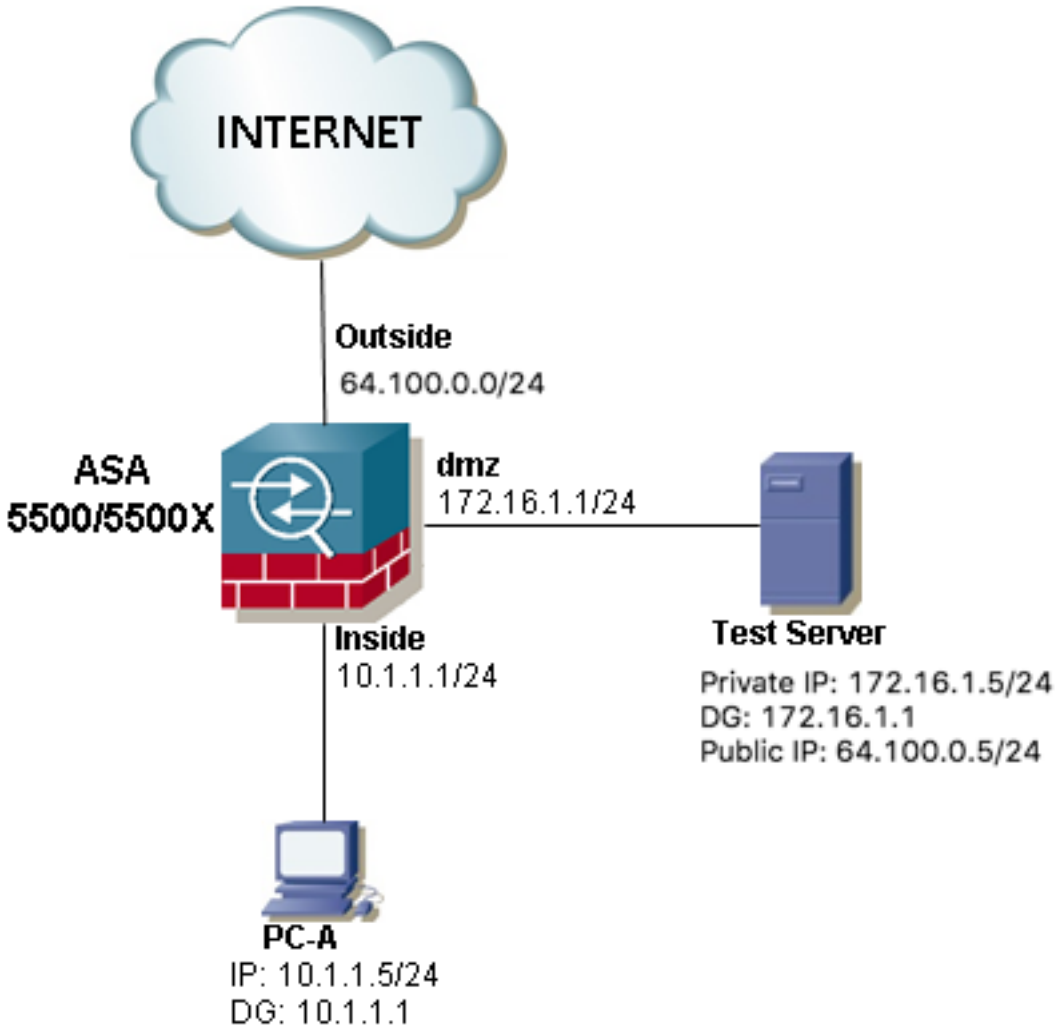
Nota: Il **server di test** nelle tre immagini ha un NAT (Network Address Translation) statico configurato nell'ASA. Questa traduzione NAT statica viene applicata dall'esterno all'interfaccia interna corrispondente per consentire al **server di test di** essere raggiungibile dall'esterno con l'indirizzo IP pubblico 64.100.0.5, quindi viene convertito nell'indirizzo IP privato interno del **server di test**.

Soluzione

Per consentire all'host di origine PC-A di raggiungere il server di test di destinazione con il suo indirizzo IP pubblico anziché quello privato, è necessario applicare una configurazione NAT doppia. La configurazione NAT a due velocità ci aiuta a tradurre sia gli indirizzi IP di origine che quelli di destinazione dei pacchetti quando il traffico attraversa l'ASA.

Ecco i dettagli della configurazione NAT doppia richiesta per ciascuna topologia:

Esempio 1. L'host di origine PC-A è connesso all'interfaccia ASA interna, mentre l'host di destinazione Test Server è connesso all'interfaccia DMZ.



Configurazione

Due volte NAT per ASA versione 8.3 e successive:

```
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.
```

Due volte NAT per ASA versione 8.2 e precedenti:

```
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
```

Risoluzione dei problemi

Output Packet Tracer versioni 8.3 e successive:

```
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
```

Output Packet Tracer versioni 8.2 e precedenti:

```
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:
```

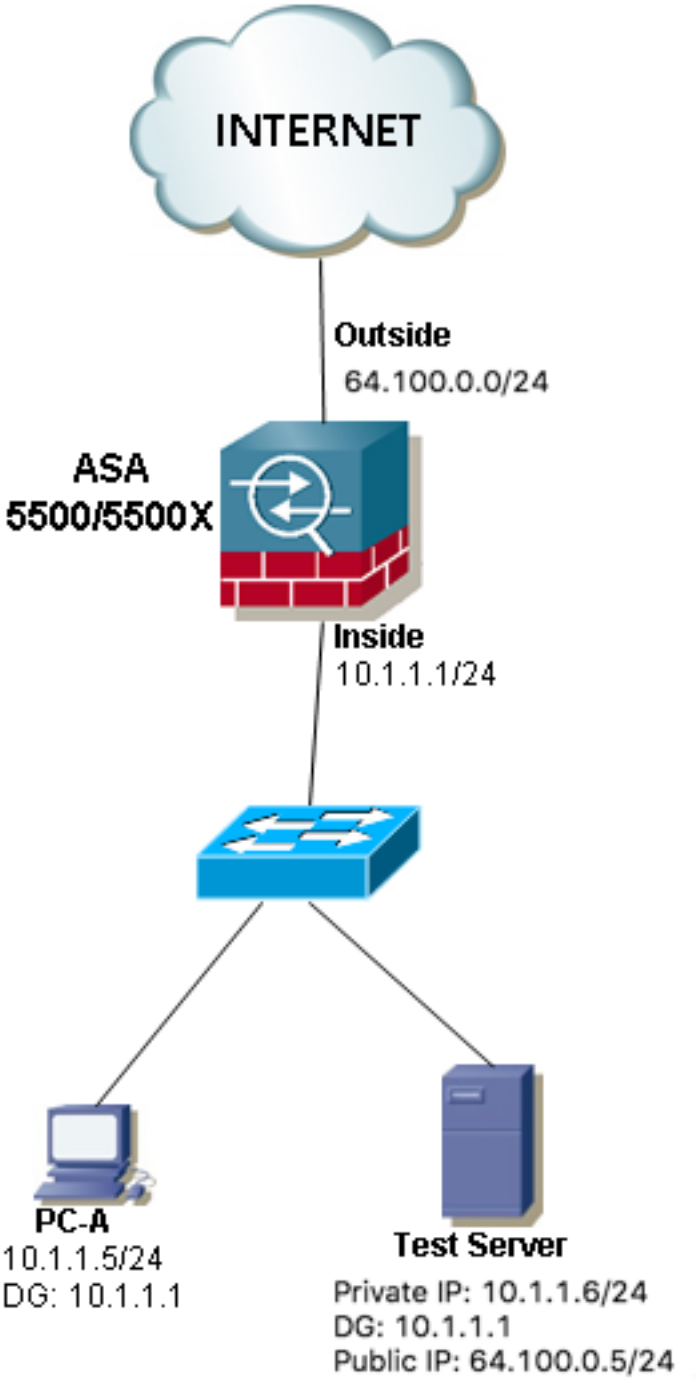
```
Phase: 3
Type: NAT
Subtype:
Result: ALLOW
Config:
static (inside,dmz) interface access-list IN-DMZ-INTERFACE
match ip inside host 10.1.1.5 dmz host 64.100.0.5
static translation to 172.16.1.1
translate_hits = 1, untranslate_hits = 0
Additional Information:
Static translate 10.1.1.5/0 to 172.16.1.1/0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

Acquisizioni pacchetti:

```
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
```

Esempio 2. Gli host di origine e destinazione PC-A e Test Server sono connessi alla stessa interfaccia ASA.



Configurazione

Due volte NAT per ASA versione 8.3 e successive:

```
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.
```

Due volte NAT per ASA versione 8.2 e precedenti:

```
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
```

> **Nota:** L'intenzione principale della conversione NAT dell'indirizzo IP di origine da 10.1.1.5 all'indirizzo IP dell'interfaccia interna ASA 10.1.1.1 è forzare le risposte provenienti dall'host 10.1.1.6 a tornare all'appliance ASA. Questa operazione è estremamente necessaria per evitare il routing asimmetrico e per consentire all'appliance ASA di elaborare tutto il traffico tra gli host interessati. Se non si traduce l'indirizzo IP di origine come è stato fatto nell'esempio, l'appliance ASA bloccherà il traffico interessato a causa del routing asimmetrico.

Risoluzione dei problemi

Output Packet Tracer versioni 8.3 e successive:

```
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
```

Output Packet Tracer versioni 8.2 e precedenti:

```
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
```

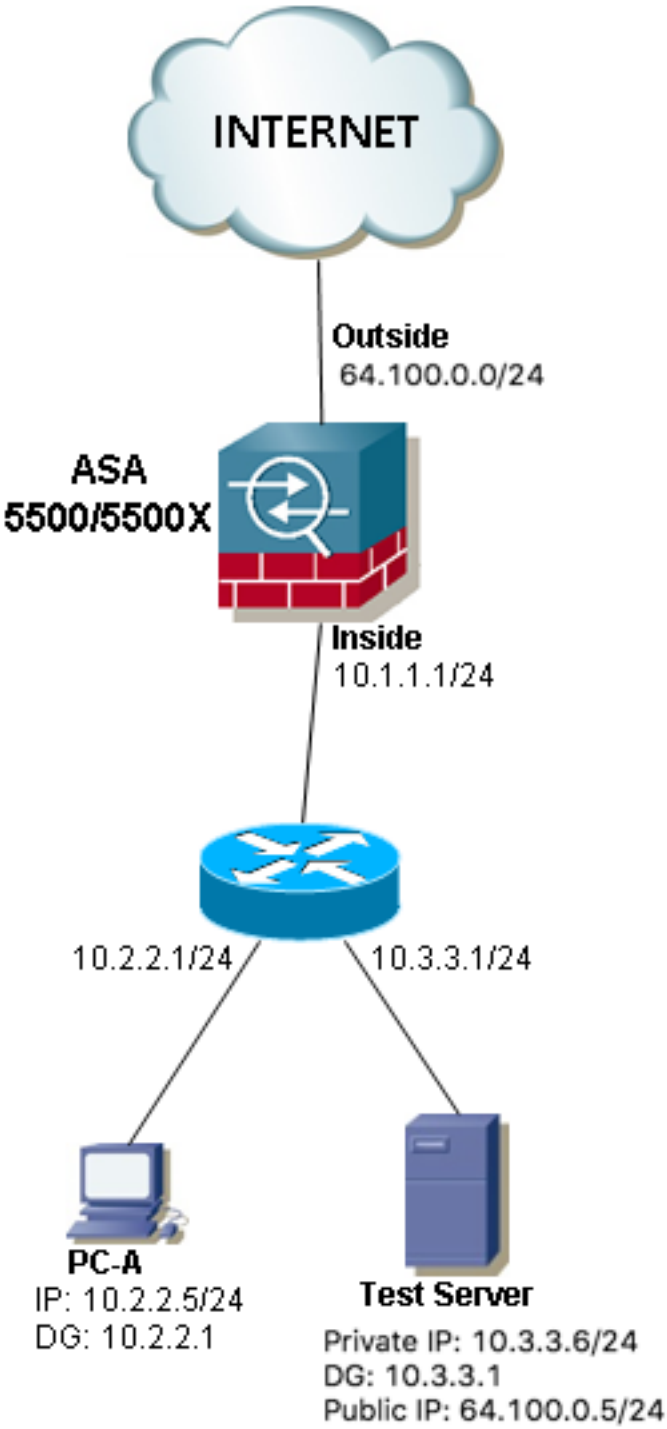
```
static translation to 64.100.0.5
translate_hits = 0, untranslate_hits = 1
Additional Information:
NAT divert to egress interface inside
Untranslate 64.100.0.5/0 to 10.1.1.6/0 using netmask 255.255.255.255

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
```

Acquisizioni pacchetti:

```
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
```

Esempio 3. L'host di origine e di destinazione PC-A e il server di prova sono collegati all'interfaccia ASA interna, ma dietro un altro dispositivo di livello 3 (potrebbe essere un router o uno switch multilivello).



Configurazione

Due volte NAT per ASA versione 8.3 e successive:

```
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.
```

Due volte NAT per ASA versione 8.2 e precedenti:

```
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
```

> **Nota**: l'intenzione principale della conversione NAT dell'indirizzo IP di origine da 10.1.1.5 all'indirizzo IP dell'interfaccia interna ASA (10.1.1.1) è forzare le risposte provenienti dall'host 10.1.1.6 a tornare all'appliance ASA. Questa operazione è estremamente necessaria per evitare il routing asimmetrico e per consentire all'appliance ASA di elaborare tutto il traffico tra gli host interessati. Se non si converte l'indirizzo IP di origine come nell'esempio, l'appliance ASA bloccherà il traffico interessato a causa del routing asimmetrico.

Risoluzione dei problemi

Output Packet Tracer versioni 8.3 e successive:

```
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
```

Output Packet Tracer versioni 8.2 e precedenti:

```
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
```

Acquisizioni pacchetti:

```
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
```

Informazioni correlate

- Guida alla configurazione di ASA 8.3: Prerequisito per due NAT
- Guida alla configurazione di ASA 8.4: DNS e NAT
- Esempi di configurazione ASA NAT precedenti alla versione 8.3-8.3