

Application Note

# Configuring Serial Interface WAN and LAN for SSG Firewall/VPN Products

---

Version 1.0

Richard Kim  
Advanced JTAC  
Tier 3 Customer Support Engineer



Juniper Networks, Inc.  
1194 North Mathilda Avenue  
Sunnyvale, CA 94089 USA  
408 745 2000 or 888 JUNIPER  
[www.juniper.net](http://www.juniper.net)

## Contents

Contents.....	2
Introduction.....	3
Included Platforms and ScreenOS.....	3
Telco/ISP Provided Information.....	3
Use of the Loopback Interface .....	4
Network Topology.....	5
Configuration Example.....	5
Private IP Configuration.....	5
Serial Interface and PPP Configuration .....	5
Loopback Interface and DNS Configuration .....	6
Route-Based VPN Configuration.....	7
MIP on Loopback Configuration .....	8
Policy Configuration .....	8
Verifying Configuration .....	9
Debug Commands .....	12

## Introduction

The Juniper Networks Secure Services Gateway Series (SSG) represents a new class of purpose-built security appliance that delivers a mix of high performance, security and LAN/WAN connectivity for regional and branch office deployments. Prior to the introduction of the SSG Series, these sites would normally employ an Ethernet-based NetScreen Firewall and a separate WAN routing device for connectivity to the Telco or ISP. The SSG Series combines the two into one platform with the same functionality and security as the rest of the NetScreen Family of firewall/VPN products.

This document gives a typical deployment example for the SSG with a WAN interface. In this example we are utilizing a T1 for Telco connectivity and PPP for the data link encapsulation. This example also assumes that you are given a WAN IP and also a public IP subnet range. This document also includes a typical MIP scenario using one of the available public IPs mapped to a private IP on the Trust zone. And finally this document includes an example of how to configure a VPN using the public IP of the SSG as the peer endpoint.

This document is not intended to show all possible configuration examples. Nor will this document cover extensive details about VPNs, MIPs or policies. More information regarding these topics are available in our Concepts and Examples Reference Guides.

## Included Platforms and ScreenOS

This document applies to any ScreenOS-based platform that supports WAN interfaces. However this does not include ADSL or dialup modem connections. Refer to the *ScreenOS 5.4 Concepts and Examples Guide, Volume 12: WAN, ADSL, Dial, and Wireless*, for more information about ADSL or dialup modem applications.

The product list includes the following:

- SSG5/SSG20
- SSG140
- SSG520/SSG550
- SSG520M/SSG550M

Note, although the configuration example uses ScreenOS 5.4.0 but also applies to 5.1.0 branch for the SSG520/550.

## Telco/ISP Provided Information

Your Telco/ISP will provide the necessary information to configure the SSG. The information provided may vary amongst various providers, but there are several basic things which are required to properly configure the SSG to access the Internet. These include but may not be limited to:

- WAN interface physical settings (T1, E1, Serial or DS3 options)
- Data link encapsulation settings (PPP, Frame Relay, Cisco HDLC)
- WAN address and subnet mask (Static IP or unnumbered)

- LAN or publicly routable IP address/subnet mask and default gateway
- If PPP, User name/password and auth type (may be optional)
- If Frame Relay, DLCI, LMI and other FR options.
- DNS IP and also possibly domain name

Below is an example of common information provided by a Telco/ISP for a T1 with PPP encapsulation, PAP authentication and static IP address for the serial interface. Refer to the *ScreenOS 5.4 Concepts and Examples Guide, Volume 12: WAN, ADSL, Dial, and Wireless*, for more information regarding WAN and PPP/Frame Relay options.

T1 framing: **ESF**  
T1 line encoding: **B8ZS**  
T1 byte encoding: **nx64** (8 bits per byte)  
TDM Time slots: **1-3 & 22-24**  
Clocking: **External**  
  
Encapsulation: **PPP**  
Auth method: **PAP**  
Username: **username**  
Password: **password**  
  
WAN IP: **2.2.2.2/30**  
LAN IP: **3.3.3.1/28**  
Default Gateway: **2.2.2.1**  
DNS Primary: **4.2.2.1**  
DNS Secondary: **4.2.2.2**

## Use of the loopback Interface

In the above, we are given two IPs. The WAN IP is designated for the serial interface IP and is usually a /30 subnet. You also have the LAN IP which can be a single public IP or a subnet range as ordered by your ISP. This scenario is sometimes referred to as interface-based routing as opposed to system-based routing. However in ScreenOS you cannot assign more than one IP subnet to the serial interface.

This is why we use the loopback interface. A loopback interface is a logical interface that emulates a physical interface on the SSG. However, unlike a physical interface, a loopback interface is always in the up state as long as the device on which it resides is up. Like a physical interface, you must assign an IP address to a loopback interface and bind it to a security zone. For more details regarding the loopback interface refer to the *ScreenOS 5.4 Concepts and Examples Guide, Volume 2: Fundamentals*.

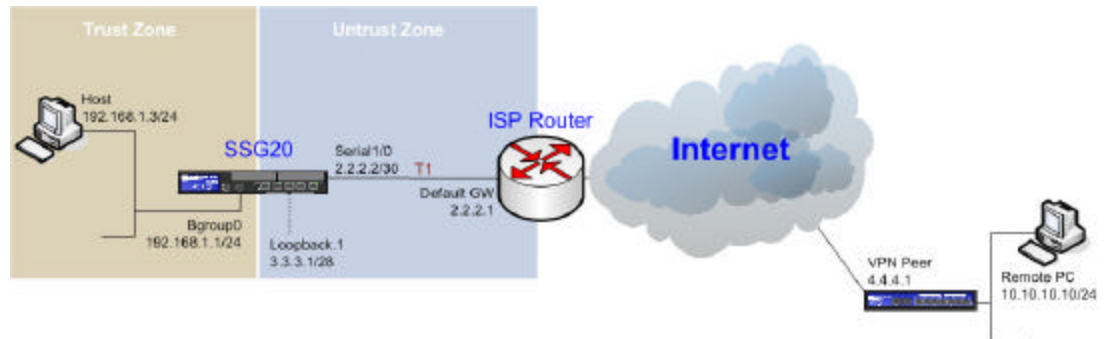
### Note:

The WAN IP may not necessarily be a publicly routable IP address. This is possible because the serial interface is a point-to-point link. Thus the upstream router would only need to know your WAN IP to forward your publicly routable IP subnet to your SSG.

## Network Topology

Refer to Figure 1 below for Network Topology used for this configuration example.

**Figure 1.**



## Configuration Example

Below are the sample settings for the SSG platform. This assumes that your remote VPN peer has the correct matching VPN configuration for the SSG peer. For more information regarding VPN configuration refer to the *ScreenOS 5.4 Concepts and Examples Guide, Volume 5: Virtual Private Networks*.

### Private IP Configuration on Trust Zone

#### WebUI

*Network > Interfaces > List > Edit (bgroup0)*  
 Zone Name: **Trust (trust-vr)**  
 Static IP  
 IP Address/Netmask: **192.168.1.1/24**  
 Interface Mode: **NAT**

#### CLI

```
set interface bgroup0 zone trust
set interface bgroup0 ip 192.168.1.1/24
set interface bgroup0 nat
```

### Serial Interface and PPP Configuration

#### WebUI

*Network > PPP > PPP Profile > New*  
 PPP Profile: **ISP**  
 Authentication: **PAP**  
 Static IP: **Check box**  
 Netmask: **255.255.255.255**  
 Local Name: **username**  
 Password: **password**

*Network > Interfaces > List > Edit (serial1/0)*

WAN Encapsulation: **PPP**  
Binding a PPP Profile: **ISP**  
Zone Name: **Untrust (trust-vr)**  
Fixed IP  
IP Address/Netmask: **2.2.2.2/30**  
Interface Mode: **Route**

*Network > Interfaces > List > Edit (serial1/0) > WAN (T1/E1)*

Clocking: **External**  
Line Encoding: **B8ZS (8-bits Zero Suppression)**  
Byte Encoding: **8-bits per byte**  
Framing Mode: **Extended Super Frame**  
Time Slots: **1-3,22-24**

*Network > Routing > Destination > New*

IP Address/Netmask: **0.0.0.0/0**  
Next Hop  
Gateway  
Interface: **serial1/0**  
Gateway IP Address: **2.2.2.1**

## CLI

```
set interface serial1/0 t1-options framing esf
set interface serial1/0 t1-options line-encoding b8zs
set interface serial1/0 t1-options timeslots 1-3,22-24
set interface serial1/0 clocking external
set interface serial1/0 zone Untrust
set interface serial1/0 encap ppp
set interface serial1/0 ip 2.2.2.2/30
set interface serial1/0 route
set interface serial1/0 gateway 2.2.2.1
set ppp profile "ISP"
set ppp profile ISP static-ip
set ppp profile ISP auth type pap
set ppp profile ISP auth local-name "username"
set ppp profile ISP auth secret "password"
set interface serial1/0 ppp profile ISP
```

## Note:

When configuring PPP, you **MUST** specify a PPP profile for the serial interface even if no authentication is required and you are given a static IP. Assuming physical link is up, command **get interface serial1/0** will show “phy-link up” but “link down, link protocol down” unless you specify a profile.

## Loopback Interface and DNS Configuration

### WebUI

*Network > Interfaces > List > New (Loopback IF)*

Zone: **Untrust (trust-vr)**  
IP Address/Netmask: **3.3.3.1/28**

*Network > DNS > Host*

Primary DNS Server: **4.2.2.1**  
Secondary DNS Server: **4.2.2.2**

#### CLI

```
set interface loopback.1 zone Untrust
set interface loopback.1 ip 3.3.3.1/28
set interface loopback.1 route
set dns host dns1 4.2.2.1
set dns host dns2 4.2.2.2
```

### Route-based VPN Configuration

#### WebUI

*Network > Interfaces > List > New (Tunnel IF)*

Tunnel Interface Name: **tunnel.1**  
Zone (VR): **Untrust (trust-vr)**  
Unnumbered  
Interface: **loopback.1 (trust-vr)**

*VPNs > AutoKey Advanced > Gateway > New*

Gateway Name: **TestIke**  
Security Level: **Standard**  
Static IP Address  
IP Address/Hostname: **4.4.4.1**  
Preshared Key: **netscreen**  
Outgoing Interface: **loopback.1**

*VPNs > AutoKey IKE > New*

VPN Name: **TestVPN**  
Security Level: **Standard**  
Remote Gateway: **TestIke**  
Advanced  
Security Level: **Standard**  
Bind to  
Tunnel Interface: **tunnel.1**

*Network > Routing > Destination > New*

IP Address/Netmask: **10.10.10.0/24**  
Next Hop  
Gateway  
Interface: **tunnel.1**

#### CLI

```
set interface tunnel.1 zone Untrust
set interface tunnel.1 ip unnumbered interface loopback.1
set ike gateway "TestIke" address 4.4.4.1 outgoing loopback.1 presh netscreen sec-lev
standard
set vpn "TestVPN" gateway "TestIke" no-replay tunnel idletime 0 sec-level standard
set vpn "TestVPN" bind interface tunnel.1
set route 10.10.10.0/24 interface tunnel.1 preference 20
```

## MIP on Loopback Interface

### WebUI

*Network > Interfaces > List > Edit (**loopback.1**) > MIP > New*

Mapped IP: **3.3.3.3**

Host IP: **192.168.1.3**

Netmask: **255.255.255.255**

Host Virtual Router Name: **trust-vr**

*Policies > From Untrust to Trust > New*

Source Address - Address Book Entry: **Any**

Destination Address - Address Book Entry: **MIP(3.3.3.3)**

Service: **ANY**

Action: **Permit**

### CLI

```
set interface loopback.1 mip 3.3.3.3 host 192.168.1.3 netmask 255.255.255.255
```

```
set policy from Untrust to Trust Any MIP(3.3.3.3) ANY permit
```

## Permitting Outgoing and Untrust Intrazone Traffic

### WebUI

*Objects > Addresses > List > New (**Untrust**)*

Address Name: **Public\_Subnet**

IP Address/Domain Name

IP Address/Netmask: **3.3.3.1/28**

*Policies > From Trust to Untrust > New*

Source Address - Address Book Entry: **Any**

Destination Address - Address Book Entry: **Any**

Service: **ANY**

Action: **Permit**

*Policies > From Untrust to Untrust > New*

Source Address - Address Book Entry: **Any**

Destination Address - Address Book Entry: **Public\_Subnet**

Service: **ANY**

Action: **Permit**

*Policies > From Untrust to Untrust > New*

Source Address - Address Book Entry: **Public\_Subnet**

Destination Address - Address Book Entry: **Any**

Service: **ANY**

Action: **Permit**

### CLI

```
set address Untrust "Public_Subnet" 3.3.3.1/28
```

```
set policy from Trust to Untrust Any Any ANY permit
```

```
set policy from Untrust to Untrust Any Public_Subnet ANY permit
```

```
set policy from Untrust to Untrust Public_Subnet Any ANY permit
```

**Note:**

You must have policies to allow Untrust intrazone traffic. This is necessary to allow the traffic to and from loopback.1 and serial1/0 which both need to be in the same zone. An alternative to using Untrust to Untrust policies is to disable intrazone block on the Untrust zone. This is considered less secure since it enables all intrazone traffic in the zone. This command is done on a per zone basis with the following command: **unset zone Untrust block**.

## Verifying Configuration

To check interface configuration and status, use command: **get interface**.

```
ssg20-> get interface
```

A - Active, I - Inactive, U - Up, D - Down, R - Ready

Interfaces in vsys Root:

Name	IP Address	Zone	MAC	VLAN	State	VSD
serial0/0	0.0.0.0/0	Null	0014.f6e6.2d4d	-	D	-
eth0/0	0.0.0.0/0	Untrust	0014.f6e6.2d40	-	D	-
eth0/1	0.0.0.0/0	DMZ	0014.f6e6.2d45	-	D	-
bgroup0	192.168.1.1/24	Trust	0014.f6e6.2d49	-	U	-
eth0/2	N/A	N/A	N/A	-	D	-
eth0/3	N/A	N/A	N/A	-	D	-
eth0/4	N/A	N/A	N/A	-	U	-
bgroup1	0.0.0.0/0	Null	0014.f6e6.2d4a	-	D	-
bgroup2	0.0.0.0/0	Null	0014.f6e6.2d4b	-	D	-
bgroup3	0.0.0.0/0	Null	0014.f6e6.2d4c	-	D	-
serial1/0	2.2.2.2/30	Untrust	N/A	-	U	-
tun.1	unnumbered	Untrust	loopback.1	-	R	-
loopback.1	3.3.3.1/28	Untrust	N/A	-	U	-
vlan1	0.0.0.0/0	VLAN	0014.f6e6.2d4f	1	D	-
null	0.0.0.0/0	Null	N/A	-	U	0

You can also check individual interfaces by appending the command with the interface name.

```
ssg20-> get interface serial1/0
```

Interface serial1/0:

```
description serial1/0
number 14, if_info 1232, if_index 0, encap ppp, mode route
link up, link protocol up, phy-link up
speed T1, clocking external, holdtime: up 0 ms, down 0 ms
t1-options:
  BERT time period: 10 seconds, algorithm: pseudo-2e15-o151, error rate
  10e-0
  buildout: 0-132 feet
  byte encoding: nx64, line encoding: b8zs
  data inversion: disabled
  loopback: none, fcs: 16, framing: sf
  remote loopback respond: disabled
  idle cycle flag: flags
  start/end flag: filler
  timeslots: 1-3,22-24
vsys Root, zone Untrust, vr trust-vr
```

```
admin mtu 1500, operating mtu 1500, default mtu 1500
*ip 2.2.2.2/30
gateway 2.2.2.1
*manage ip 2.2.2.2
route-deny disable
pmtu-v4 disabled
ping enabled, telnet disabled, SSH disabled, SNMP disabled
web enabled, ident-reset disabled, SSL disabled
DNS Proxy disabled, webauth disabled, webauth-ip 0.0.0.0
OSPF disabled BGP disabled RIP disabled RIPng disabled mtrace
disabled
PIM: not configured IGMP not configured
bandwidth: physical 1544kbps, configured egress [gbw 0kbps mbw 0kbps]
            configured ingress mbw 0kbps, current bw 0kbps
            total allocated gbw 0kbps
Number of SW session: 4058, hw sess err cnt 0
```

**ssg20-> get interface tunnel.1**

```
Interface tunnel.1:
description tunnel.1
number 20, if_info 1768, if_index 1, mode route
link ready
vsys Root, zone Untrust, vr trust-vr
admin mtu 1500, operating mtu 1500, default mtu 1500
*ip 0.0.0.0/0 unnumbered, source interface loopback.1
*manage ip 0.0.0.0
bound vpn:
    TestVPN

Next-Hop Tunnel Binding table
Flag Status Next-Hop(IP)    tunnel-id  VPN
      R      4.4.4.1           0x00000001 TestVPN

pmtu-v4 disabled
ping disabled, telnet disabled, SSH disabled, SNMP disabled
web disabled, ident-reset disabled, SSL disabled
DNS Proxy disabled
OSPF disabled BGP disabled RIP disabled RIPng disabled mtrace
disabled
PIM: not configured IGMP not configured
bandwidth: physical 0kbps, configured egress [gbw 0kbps mbw 0kbps]
            configured ingress mbw 0kbps, current bw 0kbps
            total allocated gbw 0kbps
Number of SW session: 4060, hw sess err cnt 0
```

**ssg20-> get interface loopback.1**

```
Interface loopback.1:
description loopback.1
number 126, if_info 11096, if_index 1, mode route
link up
Loopback interface has 0 members:

vsys Root, zone Untrust, vr trust-vr
admin mtu 1500, operating mtu 1500, default mtu 1500
*ip 3.3.3.1/28
*manage ip 3.3.3.1
pmtu-v4 disabled
ping enabled, telnet disabled, SSH disabled, SNMP disabled
web enabled, ident-reset disabled, SSL disabled
```

```

DNS Proxy disabled
OSPF disabled BGP disabled RIP disabled RIPng disabled mtrace
disabled
PIM: not configured IGMP not configured
Number of SW session: 4059, hw sess err cnt 0

```

To verify PPP configuration, use command: **get ppp profile <profile name>**.

```

ssg20-> get ppp profile "ISP"
PPP Profile: ISP
Authentication:
Type: pap
Local name: username
Secret: OGQoFDv8NewjatsJ3rCLoX42V/npLlw3Vw==
Netmask: 255.255.255.255
Static-ip: Enabled
Passive: Disabled
Involved interface( 1 ): serial1/0

```

```

ssg20-> get int serial1/0 ppp
LCP : OPENED
IPCP : OPENED
Keep alive interval : 10
Keep alive down counter : 3
Binding profile : ISP

```

You can also check the event log for PPP related entries.

```

ssg20-> get event include ppp
Date      Time      Module Level  Type Description
2006-11-28 11:00:41 system notif 00572 PPP protocol on interface serial2/0
is
                                         UP, local IP: 208.223.208.210, peer
IP:
                                         208.223.208.209 .
2006-11-28 11:00:41 system notif 00572 PPP LCP on interface serial2/0 is
UP.
Total entries matched = 2

```

To verify VPN configuration, use command: **get sa**. Status should show “A/-“ (or “A/U” if VPN monitoring is enabled).

```

ssg20-> get sa
total configured sa: 1
HEX ID   Gateway      Port Algorithm      SPI      Life:sec kb Sta  PID
vsys
00000001< 4.4.4.1 500 esp:3des/sha1 018e258b 3597 unlim A/- -1 0
00000001> 4.4.4.1 500 esp:3des/sha1 e1db5910 3597 unlim A/- -1 0

ssg20-> get sa id 0x1
index 0, name TestVPN, peer gateway ip 4.4.4.1. vsys<Root>
auto key. tunnel if binding node, tunnel mode, policy id in:<-1> out:<-1>
vpngrp:<-1>. sa_list_nxt:<-1>.

```

```
tunnel id 1, peer id 0, NSRP Local.      site-to-site. Local interface is
loopback.1 <3.3.3.1>.
  esp, group 2, 3des encryption, sha1 authentication
  autokey, IN active, OUT active
  monitor<0>, latency: 0, availability: 0
  DF bit: clear
  app_sa_flags: 0x2063
  proxy id: local 0.0.0.0/0.0.0.0, remote 0.0.0.0/0.0.0.0, proto 0, port 0
  ike activity timestamp: 171684
nat-traversal map not available
incoming: SPI 018e258b, flag 00004000, tunnel info 40000001, pipeline
  life 3600 sec, 3581 remain, 0 kb, 0 bytes remain
  anti-replay off, idle timeout value <0>, idled 18 seconds
  next pak sequence number: 0x0
outgoing: SPI e1db5910, flag 00000000, tunnel info 40000001, pipeline
  life 3600 sec, 3581 remain, 0 kb, 0 bytes remain
  anti-replay off, idle timeout value <0>, idled 18 seconds
  next pak sequence number: 0x4
```

Verify that hosts across VPN are reachable by pinging from the Trust side to a host on the remote peer side.

```
ssg20-> ping 10.10.10.10 from bgroup0
Type escape sequence to abort

Sending 5, 100-byte ICMP Echos to 10.10.10.10, timeout is 1 seconds from
bgroup0
!!!!
Success Rate is 100 percent (5/5), round-trip time min/avg/max=29/29/31 ms
```

## Debug Commands

Below are some debug commands to run if any issues are encountered during connection or traffic flow:

- **get interface <interface> frame** (frame relay status and statistics)
- **debug ppp all** (debugging failure to authenticate or other PPP issues)
- **debug fr all** (debugging frame relay issues)
- **debug hdlc all** (debugging Cisco HDLC issues)
- **debug flow basic** (debugging traffic flow, policies, etc. Recommend using flow filters)
- **debug ike all** (debugging VPN connection issues)
- **debug mip all** (MIP debugging)
- **debug driver [tx|txdump|rx|rxdump]** (get information on packets xmit/recv'd)

assumes no responsibility for any inaccuracies in this document or for any obligation to update information in this document. Juniper Networks reserves the right to change, modify, transfer, or otherwise revise this publication without notice.