# 영역 기반 정책 방화벽 설계 이해

## 목차

# 소개

이 문서에서는 Cisco IOS® 방화벽 기능 집합인 ZFW(Zone-based Policy Firewall)의 컨피그레이션 모델에 대해 설명합니다.

# 사전 요구 사항

## 요구 사항

이 문서에 대한 특정 요건이 없습니다.

## 사용되는 구성 요소

이 문서는 특정 소프트웨어 및 하드웨어 버전으로 한정되지 않습니다.

이 문서의 정보는 특정 랩 환경의 디바이스를 토대로 작성되었습니다. 이 문서에 사용된 모든 디바이스는 초기화된(기본) 컨피그레이션으로 시작되었습니다. 현재 네트워크가 작동 중인 경우 모든 명령의 잠재적인 영향을 미리 숙지하시기 바랍니다.

## 표기 규칙

문서 규칙에 대한 자세한 내용은 Cisco 기술 팁 표기 규칙을 참고하십시오.

# 배경 정보

이 새로운 컨피그레이션 모델은 다중 인터페이스 라우터에 대한 직관적인 정책, 방화벽 정책 애플리케이션의 세분화된 강화, 원하는 트래픽을 허용하기 위해 명시적 정책이 적용될 때까지 방화벽 보안 영역 간의 트래픽을 금지하는 기본 거부 정책을 제공합니다.

Cisco IOS Software Release 12.4(6)T 이전에 구현된 거의 모든 기존 Cisco IOS Firewall 기능은 새로운 영역 기반 정책 검사 인터페이스에서 지원됩니다.

- 상태 저장 패킷 검사
- VRF 인식 Cisco IOS 방화벽
- URL 필터링
- DoS(서비스 거부) 완화

Cisco IOS Software 릴리스 12.4(9)T에는 클래스별 세션/연결 및 처리량 제한은 물론 애플리케이션 검사 및 제어에 대한 ZFW 지원이 추가되었습니다.

- HTTP
- POP3(Post Office Protocol), IMAP(Internet Mail Access Protocol), SMTP/ESMTP(Simple Mail Transfer Protocol) 향상
- Sun RPC(원격 프로시저 호출)
- 인스턴트 메시징(IM) 애플리케이션:마이크로소프트 메신저야후! 메신저AOL 인스턴트 메신저
- P2P(Peer-to-Peer) 파일 공유:비트토렌트카자그누텔라당나귀

Cisco IOS Software 릴리스 12.4(11)T에는 더 쉬운 DoS 보호 조정을 위해 통계가 추가되었습니다.

Cisco IOS Classic Firewall의 일부 기능은 Cisco IOS Software Release 12.4(15)T의 ZFW에서 아직 지원되지 않습니다.

- 인증 프록시
- 스테이트풀 방화벽 장애 조치
- 통합 방화벽 MIB
- IPv6 상태 기반 검사
- TCP 비순차적 지원

ZFW는 일반적으로 대부분의 방화벽 검사 활동에서 Cisco IOS 성능을 개선합니다. Cisco IOS ZFW나 Classic Firewall에는 멀티캐스트 트래픽에 대한 스테이트풀 검사 지원이 포함되지 않습니다.

# 영역 기반 정책 개요

Cisco IOS Classic Firewall 상태 기반 검사(이전의 CBAC(Context-Based Access Control))는 상태 기반 검사 정책이 인터페이스에 적용되는 인터페이스 기반 컨피그레이션 모델을 사용했습니다. 모든 트래픽이 해당 인터페이스를 통과하면서 동일한 검사 정책을 수신했습니다. 이 컨피그레이션 모델은 방화벽 정책의 세분화를 제한했으며, 특히 여러 인터페이스 간에 방화벽 정책을 적용해야 하는 시나리오에서 방화벽 정책의 올바른 적용에 혼선을 초래했습니다.

영역 기반 정책 방화벽(ZFW(Zone-Policy Firewall)이라고도 함)은 기존의 인터페이스 기반 모델에서 보다 유연하고 이해하기 쉬운 영역 기반 모델로 방화벽 구성을 변경합니다. 인터페이스가 영역에 할당되고 검사 정책이 영역 사이를 이동하는 트래픽에 적용됩니다. 영역 간 정책은 상당한 유연성과 세분성을 제공하므로 동일한 라우터 인터페이스에 연결된 여러 호스트 그룹에 서로 다른 검사 정책을 적용할 수 있습니다.

방화벽 정책은 계층 구조를 사용하여 네트워크 프로토콜 및 검사를 적용할 수 있는 호스트 그룹에 대한 검사를 정의하는 CPL(Cisco Policy Language)로 구성됩니다.

# 영역 기반 정책 컨피그레이션 모델

ZFW는 Cisco IOS Classic Firewall에 비해 Cisco IOS Firewall 검사 구성 방식을 완전히 변경합니다.

방화벽 컨피그레이션의 첫 번째 주요 변경 사항은 영역 기반 컨피그레이션의 도입입니다. Cisco IOS Firewall은 영역 컨피그레이션 모델을 구현하는 최초의 Cisco IOS Software 위협 방어 기능입니다. 다른 기능은 시간이 지남에 따라 영역 모델을 채택할 수 있습니다. ip inspect 명령 집합을 사용하는 Cisco IOS Classic Firewall CBAC(stateful inspection) 인터페이스 기반 컨피그레이션 모델은 일정 기간 유지됩니다. 그러나 클래식 CLI(Command Line Interface)로 구성할 수 있는 새로운 기능은 거의 없습니다. ZFW는 상태 저장 검사 또는 CBAC 명령을 사용하지 않습니다. 두 컨피그레이션 모델은 라우터에서 동시에 사용할 수 있지만 인터페이스에서 결합되지는 않습니다. 인터페이스를 보안 영역 멤버로 구성할 수 없으며 동시에 ip inspect에 대해 구성할 수 없습니다.

영역은 네트워크의 보안 경계를 설정합니다. 영역은 트래픽이 네트워크의 다른 영역으로 이동할 때 정책 제한이 적용되는 경계를 정의합니다. 영역 간의 ZFW 기본 정책은 모두 거부입니다. 정책을 명시적으로 구성하지 않으면 영역 간에 이동하는 모든 트래픽이 차단됩니다. 이는 ACL(Access Control List)로 명시적으로 차단될 때까지 트래픽이 암시적으로 허용되는 상태 기반 검사 모델과는 크게 다릅니다.

두 번째 주요 변경 사항은 CPL이라고 하는 새로운 컨피그레이션 정책 언어의 도입입니다. Cisco IOS 소프트웨어 QoS(Modular Quality-of-Service) CLI(MQC)에 익숙한 사용자는 클래스 맵의 QoS 사용과 형식이 유사하다는 것을 인식하여 정책 맵에 적용된 작업의 영향을 받는 트래픽을 지정할 수 있습니다.

# 영역 기반 정책 방화벽 애플리케이션에 대한 규칙

영역의 라우터 네트워크 인터페이스 멤버십은 영역 멤버 인터페이스 간에 이동하는 트래픽과 마찬가지로 인터페이스 동작을 제어하는 여러 규칙이 적용됩니다.

- 영역에 인터페이스를 할당하려면 먼저 영역을 구성해야 합니다.
- 인터페이스는 하나의 보안 영역에만 할당할 수 있습니다.
- 지정된 인터페이스를 오가는 모든 트래픽은 인터페이스가 영역에 할당될 때 암시적으로 차단됩니다. 단, 동일한 영역에 있는 다른 인터페이스를 오가는 트래픽과 라우터의 어떤 인터페이스로의 트래픽은 제외됩니다.
- 트래픽은 동일한 영역의 멤버인 인터페이스 간에 기본적으로 흐르도록 암시적으로 허용됩니다.
- 영역 멤버 인터페이스를 오가는 트래픽을 허용하려면 해당 영역과 다른 영역 간에 트래픽을 허용하거나 검사하는 정책을 구성해야 합니다.
- 자체 영역은 기본 거부 모든 정책에 대한 유일한 예외입니다. 모든 라우터 인터페이스에 대한 모든 트래픽은 트래픽이 명시적으로 거부될 때까지 허용됩니다.
- 트래픽은 영역 멤버 인터페이스와 영역 멤버가 아닌 인터페이스 간에 이동할 수 없습니다. 통과, 검사 및 삭제 작업은 두 영역 사이에만 적용할 수 있습니다.
- 영역에 할당되지 않은 인터페이스는 기존 라우터 포트로 작동하며, 기존의 상태 기반 검사/CBAC 컨피그레이션을 계속 사용할 수 있습니다.
- 상자의 인터페이스가 영역/방화벽 정책의 일부가 아니어야 하는 경우. 이 인터페이스를 영역에 배치하고 트래픽 흐름이 필요한 다른 영역과 해당 영역 사이에 모든 정책 통과(일종의 더미 정책)를 구성해야 합니다.
- 이전 동작에서는 트래픽이 라우터의 모든 인터페이스 간에 이동할 경우 모든 인터페이스가 영역 지정 모델의 일부여야 합니다(각 인터페이스는 한 영역 또는 다른 영역의 멤버여야 함).
- 이전 동작의 유일한 예외인 deny by default 접근 방식은 기본적으로 허용된 라우터를 오가는 트래픽입니다. 이러한 트래픽을 제한하도록 명시적 정책을 구성할 수 있습니다.

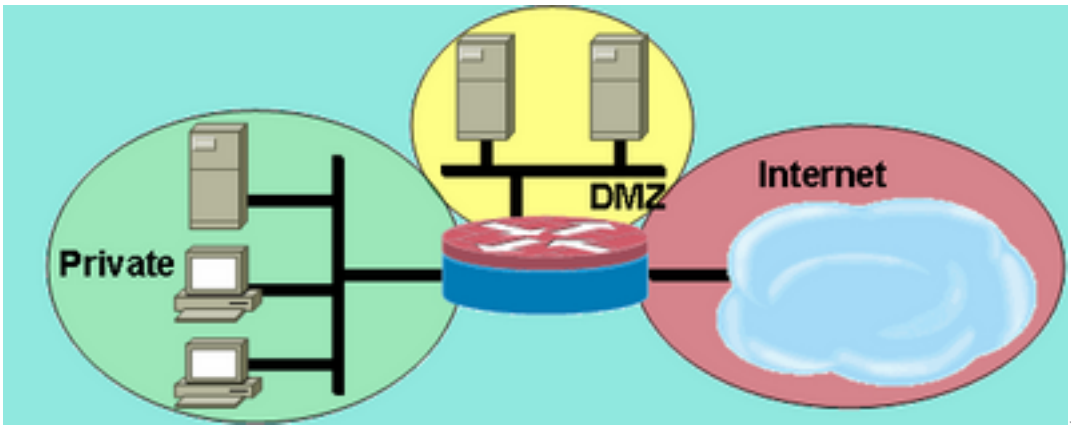
# 영역 기반 정책 네트워크 보안 설계

동일한 영역에 할당된 모든 인터페이스가 유사한 보안 수준으로 보호되도록 네트워크 내 상대 보안 영역별로 보안 영역을 구성해야 합니다. 예를 들어, 3개의 인터페이스가 있는 액세스 라우터를 가정해 보겠습니다.

- 공용 인터넷에 연결된 하나의 인터페이스
- 공용 인터넷에서 액세스할 수 없는 전용 LAN에 연결된 하나의 인터페이스
- 공용 인터넷에서 웹 서버, DNS(Domain Name System) 서버 및 이메일 서버에 액세스할 수 있어야 하는 인터넷 서비스 DMZ(Demilitarized Zone)에 연결된 하나의 인터페이스

공용 인터넷에서 DMZ의 특정 호스트로의 다양한 액세스와 보호된 LAN의 호스트에 대한 다양한 애플리케이션 사용 정책을 허용할 수 있지만 이 네트워크의 각 인터페이스는 자체 영역에 할당됩니다(그림 1 참조).

**그림 1: 기본 보안 영역 토폴로지**




*기본 보안 영역 토폴로지*

이 예에서는 각 영역에 인터페이스가 하나만 있습니다. 프라이빗 영역에 추가 인터페이스가 추가된 경우, 영역의 새 인터페이스에 연결된 호스트는 동일한 영역에 있는 현재 인터페이스의 모든 호스트에 트래픽을 전달할 수 있습니다. 또한 다른 영역의 호스트에 대한 호스트 트래픽도 현재 정책의 영향을 받습니다.

일반적으로 예제 네트워크에는 세 가지 주요 정책이 있습니다.

- 인터넷에 대한 개인 영역 연결
- DMZ 호스트에 대한 전용 영역 연결
- DMZ 호스트에 대한 인터넷 영역 연결

DMZ는 공용 인터넷에 노출되므로 DMZ 호스트는 하나 이상의 DMZ 호스트를 손상시키는데 성공할 수 있는 악의적인 사용자로부터 원하지 않는 활동을 당할 수 있습니다. DMZ 호스트가 전용 영역 호스트 또는 인터넷 영역 호스트에 도달하도록 액세스 정책을 제공하지 않으면 DMZ 호스트를 공격한 개인은 DMZ 호스트를 사용하여 사설 또는 인터넷 호스트에 대한 추가 공격을 수행할 수 없습니다. ZFW는 금지되는 기본 보안 상태를 적용합니다. 따라서 DMZ 호스트가 다른 네트워크에 대한 액세스를 특별히 제공하지 않는 한, 다른 네트워크는 DMZ 호스트의 모든 연결로부터 보호됩니다. 마찬가지로, 인터넷 호스트가 사설 영역 호스트에 액세스하기 위한 액세스는 제공되지 않으므로, 사설 영역 호스트는 인터넷 호스트에서 원치 않는 액세스로부터 안전합니다.

# 영역 기반 정책 방화벽과 함께 IPSec VPN 사용

최근 IPSec VPN이 개선되어 VPN 연결을 위한 방화벽 정책 구성이 간소화되었습니다. IPSec VTI(Virtual Tunnel Interface) 및 GRE+IPSec을 사용하면 터널 인터페이스를 지정된 보안 영역에 배치하여 특정 보안 영역에 대한 VPN 사이트 간 및 클라이언트 연결을 제한할 수 있습니다. 연결이 특정 정책에 의해 제한되어야 하는 경우 VPN DMZ에서 연결을 격리할 수 있습니다. 또는 VPN 연결을

암시적으로 신뢰할 수 있는 경우 VPN 연결을 신뢰할 수 있는 내부 네트워크와 동일한 보안 영역에 배치할 수 있습니다.

비 VTI IPSec이 적용되는 경우, VPN 연결 방화벽 정책에서는 보안을 유지하기 위해 면밀한 검토가 필요합니다. 영역 정책은 보안 호스트가 라우터에 대한 VPN 클라이언트 암호화 연결과 다른 영역에 있는 경우 원격 사이트 호스트 또는 VPN 클라이언트에 대해 IP 주소로 액세스를 허용해야 합니다. 액세스 정책이 제대로 구성되지 않으면 보호해야 하는 호스트가 원치 않는 잠재적으로 적대적인 호스트에 노출될 수 있습니다. 컨셉과 컨피그레이션에 대한 자세한 내용은 Using VPN with Zone-Based Policy Firewall을 참조하십시오.

# CPL(Cisco Policy Language) 컨피그레이션

이 절차를 사용하여 ZFW를 구성할 수 있습니다. 단계의 순서는 중요하지 않지만 일부 이벤트는 순서대로 완료해야 합니다. 예를 들어 정책 맵에 클래스 맵을 할당하려면 먼저 클래스 맵을 구성해야 합니다. 마찬가지로, 정책을 구성할 때까지 영역 쌍에 정책 맵을 할당할 수 없습니다. 구성하지 않은 컨피그레이션의 다른 부분에 의존하는 섹션을 구성하려고 하면 라우터가 오류 메시지로 응답합니다.

1. 영역을 정의합니다.
2. 영역 쌍을 정의합니다.
3. 영역 쌍을 통과할 때 정책이 적용되어야 하는 트래픽을 설명하는 클래스 맵을 정의합니다.
4. 정책 맵을 정의하여 클래스 맵 트래픽에 작업을 적용합니다.
5. 영역 쌍에 정책 맵을 적용합니다.
6. 영역에 인터페이스를 할당합니다.

## 영역 기반 정책 방화벽 클래스 맵 구성

클래스 맵은 방화벽에서 정책 애플리케이션에 대해 선택하는 트래픽을 정의합니다. 레이어 4 클래스 맵은 여기에 나열된 기준에 따라 트래픽을 정렬합니다. 이러한 기준은 클래스 맵에서 match 명령으로 지정합니다.

- Access-group — 표준, 확장 또는 명명된 ACL은 소스 및 대상 IP 주소와 소스 및 대상 포트를 기반으로 트래픽을 필터링할 수 있습니다.
- 프로토콜 — Layer 4 프로토콜(TCP, UDP, ICMP) 및 애플리케이션 서비스(예: HTTP, SMTP, DNS 등)를 지정합니다. 포트-애플리케이션 매핑에 알려진 잘 알려진 서비스나 사용자 정의 서비스도 지정할 수 있습니다.
- 클래스 맵 — 추가 일치 기준을 제공하는 하위 클래스 맵은 다른 클래스 맵 내에 중첩될 수 있습니다.
- Not — not 기준은 지정된 서비스(프로토콜), 액세스 그룹 또는 하위 클래스 맵과 일치하지 않는 모든 트래픽이 클래스 맵에 대해 선택되도록 지정합니다.

### "일치" 기준 결합: "Match-Any" 대 "Match-All"

클래스 맵은 match-any 또는 match-all 연산자를 적용하여 일치 기준을 적용하는 방법을 결정할 수 있습니다. match-any가 지정된 경우 트래픽은 클래스 맵의 일치 기준 중 하나만 충족해야 합니다. match-all이 지정된 경우 트래픽이 모든 클래스 맵 기준과 일치해야 특정 클래스에 속할 수 있습니다.

트래픽이 여러 기준을 충족하는 경우 일치 기준을 더 구체화에서 덜 구체화되는 순서로 적용해야

합니다. 예를 들어 다음 class-map을 고려하십시오.

```
class-map type inspect match-any my-test-cmap
 match protocol http
 match protocol tcp
```

HTTP 트래픽은 HTTP 검사의 서비스별 기능에 의해 트래픽이 처리되는지 확인하려면 먼저 일치 프로토콜 http를 만나야 합니다. 일치 라인이 반전되어 트래픽에서 match protocol TCP 문을 확인한 후 이를 match protocol http와 비교하는 경우, 트래픽은 TCP 트래픽으로 분류되고 Firewall TCP Inspection 구성 요소의 기능을 기반으로 검사됩니다. 이는 FTP, TFTP와 같은 특정 서비스와 H.323, SIP, Skinny, RTSP 등과 같은 여러 멀티미디어 및 음성 신호 서비스에 대한 문제입니다. 이러한 서비스의 더욱 복잡한 활동을 인식하려면 추가적인 검사 기능이 필요합니다.

## 일치 기준으로 ACL 적용

클래스 맵은 정책 애플리케이션의 일치 기준 중 하나로 ACL을 적용할 수 있습니다. class-map only matches 기준이 ACL이고 inspect 작업을 적용하는 정책 맵과 연결된 경우 라우터는 ACL에서 허용하는 모든 트래픽에 대해 기본 TCP 또는 UDP 검사를 적용합니다. 단, ZFW에서 애플리케이션 인식 검사를 제공하는 경우는 예외입니다. 여기에는 FTP, SIP, Skinny(SCCP), H.323, Sun RPC 및 TFTP가 포함됩니다(이에 제한되지 않음). 애플리케이션별 검사가 사용 가능하고 ACL에서 기본 또는 제어 채널을 허용하는 경우, ACL에서 트래픽을 허용하는지 여부에 관계없이 기본/제어와 연결된 보조 또는 미디어 채널이 허용됩니다.

클래스 맵이 ACL 101만 일치 기준으로 적용하는 경우 ACL 101이 다음과 같이 표시됩니다.

```
access-list 101 permit ip any any
```

모든 트래픽은 지정된 영역 쌍에 적용된 서비스 정책 방향으로 허용되며, 이에 해당하는 반환 트래픽은 반대 방향으로 허용됩니다. 따라서 ACL은 트래픽을 원하는 특정 유형으로 제한하기 위해 제한을 적용해야 합니다. PAM 목록에는 HTTP, NetBIOS, H.323, DNS 등의 애플리케이션 서비스가 포함됩니다. 그러나 PAM이 특정 포트의 특정 애플리케이션 사용에 대해 알고 있음에도 불구하고 방화벽은 애플리케이션 트래픽의 잘 알려진 요구 사항을 수용하기에 충분한 애플리케이션별 기능만 적용합니다. 따라서 텔넷, SSH 및 기타 단일 채널 애플리케이션과 같은 간단한 애플리케이션 트래픽은 TCP로 검사되고 해당 통계는 show 명령 출력에서 함께 결합됩니다. 네트워크 활동에 대한 애플리케이션별 가시성이 필요한 경우 애플리케이션 이름별로 서비스에 대한 검사를 구성해야 합니다(match protocol HTTP, match protocol telnet 등 구성).

이 컨피그레이션의 show policy-map type inspect zone-pair 명령 출력에서 사용 가능한 통계를 페이지 아래쪽에 표시된 더 명확한 방화벽 정책과 비교합니다. 이 컨피그레이션은 Cisco IP Phone은 물론 HTTP, FTP, NetBIOS, SSH, DNS 등 다양한 트래픽을 사용하는 여러 워크스테이션에서 트래픽을 검사하는 데 사용됩니다.

```
class-map type inspect match-all all-private
 match access-group 101
!
policy-map type inspect priv-pub-pmap
 class type inspect all-private
  inspect
 class class-default
!
zone security private
zone security public
zone-pair security priv-pub source private destination public
 service-policy type inspect priv-pub-pmap
```

```
!
interface FastEthernet4
 ip address 172.16.108.44 255.255.255.0
 zone-member security public
!
interface Vlan1
 ip address 192.168.108.1 255.255.255.0
 zone-member security private
!
access-list 101 permit ip 192.168.108.0 0.0.0.255 any
```

이 컨피그레이션은 프라이빗 영역에서 시작되는 모든 트래픽을 정의하고 수용하기는 쉽지만(트래픽이 표준 PAM 인식 목적지 포트를 준수하는 한), 서비스 활동에 대한 가시성은 제한적이며, 특정 유형의 트래픽에 대해 ZFW의 대역폭 및 세션 제한을 적용할 기회를 제공하지 않습니다. 이 show policy-map type inspect zone-pair priv-pub 명령 출력은 영역 쌍 간에 허용 IP[서브넷] any ACL만 사용하는 이전의 간단한 컨피그레이션의 결과입니다. 보다시피 대부분의 워크스테이션 트래픽은 기본 TCP 또는 UDP 통계에서 계산됩니다.

```
stg-871-L#show policy-map type insp zone-pair priv-pub
Zone-pair: priv-pub

  Service-policy inspect : priv-pub-pmap

    Class-map: all-private (match-all)
      Match: access-group 101
      Inspect
        Packet inspection statistics [process switch:fast switch]
        tcp packets: [413:51589]
        udp packets: [74:28]
        icmp packets: [0:8]
        ftp packets: [23:0]
        tftp packets: [3:0]
        tftp-data packets: [6:28]
        skinny packets: [238:0]

        Session creations since subsystem startup or last reset 39
        Current session counts (estab/half-open/terminating) [3:0:0]
        Maxever session counts (estab/half-open/terminating) [3:4:1]
        Last session created 00:00:20
        Last statistic reset never
        Last session creation rate 2
        Maxever session creation rate 7
        Last half-open session total 0

    Class-map: class-default (match-any)
      Match: any
      Drop (default action)
        0 packets, 0 bytes
```

반면, 애플리케이션별 클래스를 추가하는 유사한 컨피그레이션은 더욱 세분화된 애플리케이션 통계 및 제어를 제공하며, 정책 맵에서 ACL과 일치하는 마지막 기회 클래스 맵을 정의할 때 첫 번째 예에 나타난 것과 동일한 범위의 서비스를 계속 수용합니다.

```
class-map type inspect match-all all-private
 match access-group 101
class-map type inspect match-all private-ftp
 match protocol ftp
 match access-group 101
class-map type inspect match-any netbios
 match protocol msrpc
 match protocol netbios-dgm
```

```
 match protocol netbios-ns
 match protocol netbios-ssn
class-map type inspect match-all private-netbios
 match class-map netbios
 match access-group 101
class-map type inspect match-all private-ssh
 match protocol ssh
 match access-group 101
class-map type inspect match-all private-http
 match protocol http
 match access-group 101
!
policy-map type inspect priv-pub-pmap
 class type inspect private-http
  inspect
 class type inspect private-ftp
  inspect
 class type inspect private-ssh
  inspect
 class type inspect private-netbios
  inspect
 class type inspect all-private
  inspect
 class class-default!
zone security private
zone security public
zone-pair security priv-pub source private destination public
 service-policy type inspect priv-pub-pmap
!
interface FastEthernet4
 ip address 172.16.108.44 255.255.255.0
 zone-member security public
!
interface Vlan1
 ip address 192.168.108.1 255.255.255.0
 zone-member security private
!
access-list 101 permit ip 192.168.108.0 0.0.0.255 any
```

보다 구체적인 컨피그레이션에서는 show policy-map type inspect zone-pair priv-pub 명령에 대해 다음과 같은 실질적인 세분화된 출력을 제공합니다.

```
stg-871-L#sh policy-map type insp zone-pair priv-pub
 Zone-pair: priv-pub

   Service-policy inspect : priv-pub-pmap

     Class-map: private-http (match-all)
       Match: protocol http
       Match: access-group 101
       Inspect
         Packet inspection statistics [process switch:fast switch]
         tcp packets: [0:2193]

         Session creations since subsystem startup or last reset 731
         Current session counts (estab/half-open/terminating) [0:0:0]
         Maxever session counts (estab/half-open/terminating) [0:3:0]
         Last session created 00:29:25
         Last statistic reset never
         Last session creation rate 0
         Maxever session creation rate 4
         Last half-open session total 0
```

```
Class-map: private-ftp (match-all)
  Match: protocol ftp
  Inspect
    Packet inspection statistics [process switch:fast switch]
    tcp packets: [86:167400]
    ftp packets: [43:0]

    Session creations since subsystem startup or last reset 7
    Current session counts (estab/half-open/terminating) [0:0:0]
    Maxever session counts (estab/half-open/terminating) [2:1:1]
    Last session created 00:42:49
    Last statistic reset never
    Last session creation rate 0
    Maxever session creation rate 4
    Last half-open session total 0

Class-map: private-ssh (match-all)
  Match: protocol ssh
  Inspect
    Packet inspection statistics [process switch:fast switch]
    tcp packets: [0:62]

    Session creations since subsystem startup or last reset 4
    Current session counts (estab/half-open/terminating) [0:0:0]
    Maxever session counts (estab/half-open/terminating) [1:1:1]
    Last session created 00:34:18
    Last statistic reset never
    Last session creation rate 0
    Maxever session creation rate 2
    Last half-open session total 0

Class-map: private-netbios (match-all)
  Match: access-group 101
  Match: class-map match-any netbios
    Match: protocol msrpc
      0 packets, 0 bytes
      30 second rate 0 bps
    Match: protocol netbios-dgm
      0 packets, 0 bytes
      30 second rate 0 bps
    Match: protocol netbios-ns
      0 packets, 0 bytes
      30 second rate 0 bps
    Match: protocol netbios-ssn
      2 packets, 56 bytes
      30 second rate 0 bps
  Inspect
    Packet inspection statistics [process switch:fast switch]
    tcp packets: [0:236]

    Session creations since subsystem startup or last reset 2
    Current session counts (estab/half-open/terminating) [0:0:0]
    Maxever session counts (estab/half-open/terminating) [1:1:1]
    Last session created 00:31:32
    Last statistic reset never
    Last session creation rate 0
    Maxever session creation rate 1
    Last half-open session total 0

Class-map: all-private (match-all)
  Match: access-group 101
  Inspect
    Packet inspection statistics [process switch:fast switch]
    tcp packets: [51725:158156]
```

```
        udp packets: [8800:70]
        tftp packets: [8:0]
        tftp-data packets: [15:70]
        skinny packets: [33791:0]

        Session creations since subsystem startup or last reset 2759
        Current session counts (estab/half-open/terminating) [2:0:0]
        Maxever session counts (estab/half-open/terminating) [2:6:1]
        Last session created 00:22:21
        Last statistic reset never
        Last session creation rate 0
        Maxever session creation rate 12
        Last half-open session total 0

    Class-map: class-default (match-any)
      Match: any
      Drop (default action)
        4 packets, 112 bytes
```

앞에서 언급한 바와 같이 더욱 세분화된 클래스 맵 및 정책 맵 컨피그레이션을 사용할 경우 세션 및 속도 값에 클래스별 제한을 적용할 수 있는 또 다른 추가적인 이점이 있습니다. 그리고, 특히 각 클래스 검사 동작을 조정하기 위해 parameter-map을 적용하여 검사 파라미터를 조정한다.

## 영역 기반 정책 방화벽 정책 맵 구성

policy-map은 보안 영역 쌍에 적용되는 서비스 정책을 정의하기 위해 하나 이상의 클래스 맵에 방화벽 정책 작업을 적용합니다. inspect-type policy-map을 만들면 class class-default라는 기본 클래스가 클래스 끝에 적용됩니다. class-default 기본 정책 작업은 drop이지만 pass로 변경할 수 있습니다. log 옵션은 drop 작업과 함께 추가할 수 있습니다. 클래스 class-default에는 검사를 적용할 수 없습니다.

### 영역 기반 정책 방화벽 작업

ZFW는 한 영역에서 다른 영역으로 이동하는 트래픽에 대해 세 가지 작업을 제공합니다.

- Drop — 모든 inspect-type policy-map을 종료하는 class-default 클래스에 의해 적용되는 모든 트래픽의 기본 작업입니다. 정책 맵 내의 다른 클래스 맵도 원하지 않는 트래픽을 삭제하도록 구성할 수 있습니다. 거부된 트래픽을 전송한 호스트에 ICMP "host unreachable" 메시지를 전송하는 경우 ACL 동작과는 반대로, 삭제 작업에 의해 처리되는 트래픽은 ZFW에 의해 자동으로 삭제(즉, 삭제에 대한 알림이 관련 최종 호스트로 전송되지 않음)됩니다. 현재는 자동 삭제 동작을 변경할 수 있는 옵션이 없습니다. 방화벽에 의해 트래픽이 삭제되었다는 syslog 알림에 대해 drop과 함께 log 옵션을 추가할 수 있습니다.
- Pass — 이 작업을 통해 라우터가 한 영역에서 다른 영역으로 트래픽을 전달할 수 있습니다. pass 작업은 트래픽 내 연결 또는 세션의 상태를 추적하지 않습니다. Pass(통과)는 한 방향의 트래픽만 허용합니다. 반환 트래픽이 반대 방향으로 통과하도록 하려면 병렬 정책을 적용해야 합니다. pass 작업은 IPSec ESP, IPSec AH, ISAKMP 등의 프로토콜과 예측 가능한 동작이 포함된 기타 본질적으로 안전한 프로토콜에 유용합니다. 그러나 대부분의 애플리케이션 트래픽은 검사 작업을 통해 ZFW에서 더 잘 처리됩니다.
- Inspect — inspect 작업은 상태 기반 트래픽 제어를 제공합니다. 예를 들어, 프라이빗 영역에서 이전 예 네트워크의 인터넷 영역으로의 트래픽을 검사할 경우 라우터는 TCP 및 UDP(User Datagram Protocol) 트래픽에 대한 연결 또는 세션 정보를 유지 관리합니다. 따라서 라우터는 프라이빗 영역 연결 요청에 대한 응답으로 인터넷 영역 호스트에서 전송된 반환 트래픽을 허용합니다. 또한 inspect는 취약하거나 민감한 애플리케이션 트래픽을 전달할 수 있는 특정 서비스 프로토콜에 대한 애플리케이션 검사 및 제어를 제공할 수 있습니다. Audit-trail을 매개변수 맵과

함께 적용하여 연결/세션 시작, 중지, 기간, 전송된 데이터 볼륨, 소스 및 목적지 주소를 기록할 수 있습니다.

작업은 정책 맵의 클래스 맵과 연결됩니다.

```
configure terminal
 policy-map type inspect z1-z2-pmap
  class type inspect service-cmap
   inspect|drop|allow [service-parameter-map]
```

매개변수 맵은 지정된 클래스 맵 검사 정책에 대한 연결 매개변수를 수정하는 옵션을 제공합니다.

## 영역 정책 방화벽 매개변수 맵 구성

매개변수 맵은 DoS 보호, TCP 연결/UDP 세션 타이머, 감사 추적 로깅 설정과 같은 매개변수에 대해 ZFW에 대한 검사 동작을 지정합니다. 매개변수 맵은 HTTP 객체, POP3 및 IMAP 인증 요구 사항 및 기타 애플리케이션별 정보와 같은 애플리케이션별 동작을 정의하기 위해 레이어 7 클래스 및 정책 맵에도 적용됩니다.

ZFW에 대한 검사 매개변수 맵은 다른 ZFW 클래스 및 정책 개체와 마찬가지로 type inspect로 구성됩니다.

```
stg-871-L(config)#parameter-map type inspect z1-z2-pmap stg-871-L(config-profile)#?
parameter-map commands:
  alert           Turn on/off alert
  audit-trail     Turn on/off audit trail
  dns-timeout     Specify timeout for DNS
  exit            Exit from parameter-map
  icmp            Config timeout values for icmp
  max-incomplete  Specify maximum number of incomplete connections before
                  clamping
  no              Negate or set default values of a command
  one-minute      Specify one-minute-sample watermarks for clamping
  sessions        Maximum number of inspect sessions
  tcp             Config timeout values for tcp connections
  udp             Config timeout values for udp flows
```

특정 유형의 매개변수 맵은 레이어 7 애플리케이션 검사 정책에 의해 적용되는 매개변수를 지정합니다. Regex-type parameter-maps는 정규식으로 트래픽을 필터링하는 HTTP 애플리케이션 검사에 사용할 정규식을 정의합니다.

```
parameter-map type regex [parameter-map-name]
```

Protocol-info-type 매개 변수 맵은 IM 애플리케이션 검사에 사용할 서버 이름을 정의합니다.

```
parameter-map type protocol-info [parameter-map-name]
```

HTTP 및 IM 애플리케이션 검사에 대한 전체 컨피그레이션 세부 정보는 이 문서의 각 애플리케이션 검사 섹션에 나와 있습니다.

## 영역 기반 정책 방화벽 정책에 대한 로깅 적용

ZFW는 기본적으로 삭제되거나 구성된 방화벽 정책 작업에 의해 검사되는 트래픽에 대한 로깅 옵션

을 제공합니다. 감사 추적 로깅은 ZFW에서 검사하는 트래픽에 사용할 수 있습니다. 감사 추적은 감사 추적이 매개변수 맵에 정의되고 inspect 작업이 있는 매개변수 맵이 정책 맵에 적용될 때 적용됩니다.

```
configure terminal
 policy-map type inspect z1-z2-pmap
  class type inspect service-cmap
   inspect|drop|allow [parameter-map-name (optional)]
```

삭제 로깅은 ZFW가 삭제하는 트래픽에 사용할 수 있습니다. 삭제 로깅은 정책 맵에서 삭제 작업을 사용하여 로그를 추가하는 경우에 의해 구성됩니다.

```
configure terminal
 policy-map type inspect z1-z2-pmap
  class type inspect service-cmap
   inspect|drop|allow [service-parameter-map]
```

### 영역 정책 방화벽 클래스 맵 및 정책 맵 수정

ZFW는 현재 정책 맵, 클래스 맵, 매개변수 맵과 같은 다양한 ZFW 구조를 수정할 수 있는 편집기를 통합하지 않습니다. 클래스 맵 또는 작업 응용 프로그램의 match 문을 정책 맵에 포함된 다양한 클래스 맵으로 다시 정렬하려면 다음 단계를 완료해야 합니다.

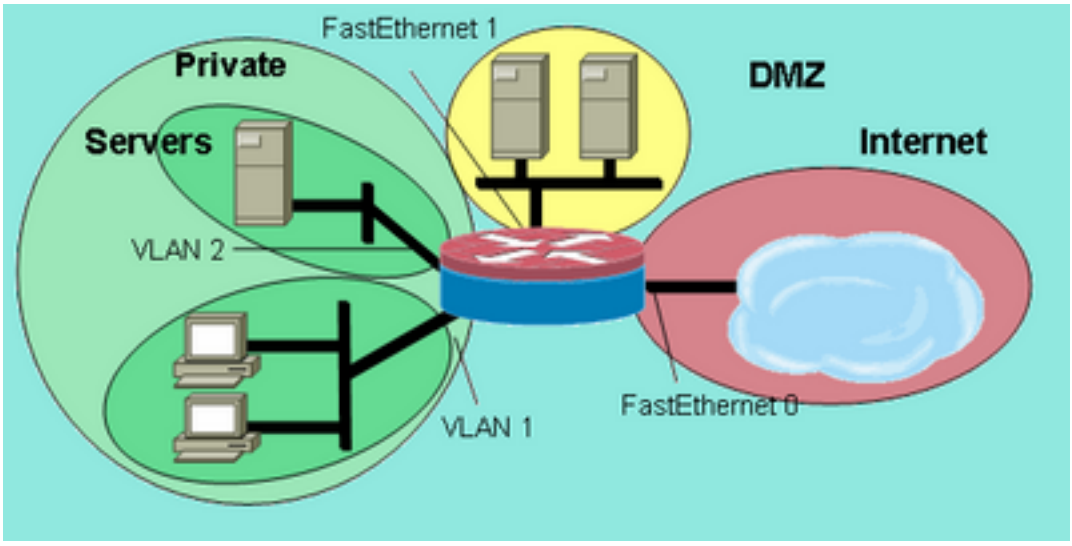
1. Microsoft Windows 메모장과 같은 텍스트 편집기나 Linux/Unix 플랫폼의 vi와 같은 편집기에 현재 구조를 복사합니다.
2. 라우터 컨피그레이션에서 현재 구조를 제거합니다.
3. 텍스트 편집기에서 구조를 편집합니다.
4. 라우터 CLI에 구조를 다시 복사합니다.

## 구성 예

이 컨피그레이션 예에서는 Cisco 1811 Integrated Services Router를 사용합니다. IP 연결, VLAN 컨피그레이션, 두 프라이빗 이더넷 LAN 세그먼트 간의 투명 브리징을 포함한 기본 컨피그레이션은 부록 A에서 확인할 수 있습니다. 라우터는 5개의 영역으로 구분됩니다.

- 공용 인터넷은 FastEthernet 0(Internet zone)에 연결됩니다
- 두 개의 인터넷 서버가 FastEthernet 1(DMZ 영역)에 연결됨
- 이더넷 스위치는 두 개의 VLAN으로 구성됩니다.워크스테이션은 VLAN1(클라이언트 영역)에 연결됩니다.서버는 VLAN2(서버 영역)에 연결됩니다.클라이언트와 서버 영역이 동일한 서브넷에 있습니다. 영역 사이에 투명 방화벽이 적용되므로, 두 인터페이스의 영역 간 정책은 클라이언트와 서버 영역 간의 트래픽에만 영향을 미칠 수 있습니다.
- VLAN1 및 VLAN2 인터페이스는 브리지 가상 인터페이스(BVI1)를 통해 다른 네트워크와 통신합니다. 이 인터페이스는 전용 영역에 할당됩니다. (그림 2 참조)

**그림 2: 영역 토폴로지 세부 정보**

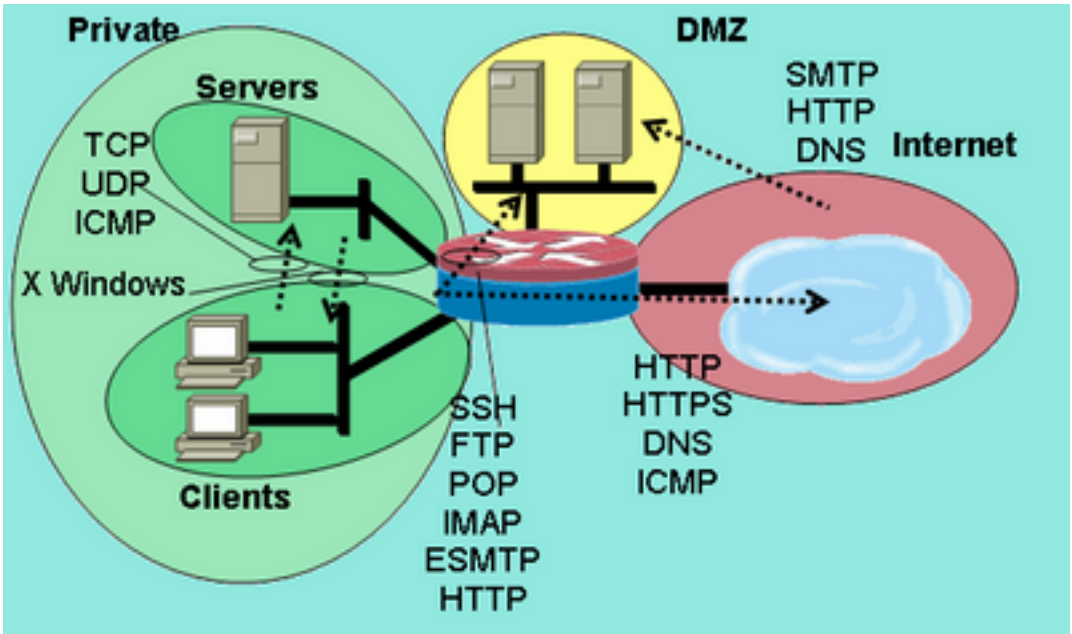



*영역 토폴로지 세부 정보*

이러한 정책은 이전에 정의된 네트워크 영역과 함께 적용됩니다.

- 인터넷 영역의 호스트는 DMZ에 있는 한 서버의 DNS, SMTP 및 SSH 서비스에 연결할 수 있습니다. 다른 서버는 SMTP, HTTP 및 HTTPS 서비스를 제공합니다. 방화벽 정책은 각 호스트에서 사용 가능한 특정 서비스에 대한 액세스를 제한합니다.
- DMZ 호스트는 다른 영역의 호스트에 연결할 수 없습니다.
- 클라이언트 영역의 호스트는 모든 TCP, UDP 및 ICMP 서비스의 서버 영역에 있는 호스트에 연결할 수 있습니다.
- 서버 영역의 호스트는 클라이언트 영역의 호스트에 연결할 수 없습니다. 단, UNIX 기반 애플리케이션 서버는 포트 6900~6910의 클라이언트 영역에 있는 데스크톱 PC의 X Windows 서버에 대한 X Windows 클라이언트 세션을 열 수 있습니다.
- 프라이빗 영역(클라이언트와 서버의 조합)의 모든 호스트는 SSH, FTP, POP, IMAP, ESMTP 및 HTTP 서비스의 DMZ와 HTTP, HTTPS, DNS 서비스 및 ICMP의 인터넷 영역에 있는 호스트에 액세스할 수 있습니다. 또한 지원되는 IM 및 P2P 애플리케이션이 포트 80에서 전달되지 않도록 하기 위해 프라이빗 영역에서 인터넷 영역으로의 HTTP 연결에 애플리케이션 검사가 적용됩니다. (그림 3 참조)

**그림 3: 컨피그레이션 예에서 적용할 영역 쌍 서비스 권한**




*컨피그레이션 예에서 적용할 영역 쌍 서비스 권한*

이러한 방화벽 정책은 복잡성 순서로 구성됩니다.

1. 클라이언트-서버 TCP/UDP/ICMP 검사
2. 사설 DMZ SSH/FTP/POP/IMAP/ESMTP/HTTP 검사
3. 인터넷 -DMZ SMTP/HTTP/DNS 검사가 호스트 주소로 제한됨
4. PAM(Port-Application Mapping) 지정 서비스를 통한 서버-클라이언트 X Windows 검사
5. HTTP 애플리케이션 검사를 사용하는 사설 인터넷 HTTP/HTTPS/DNS/ICMP

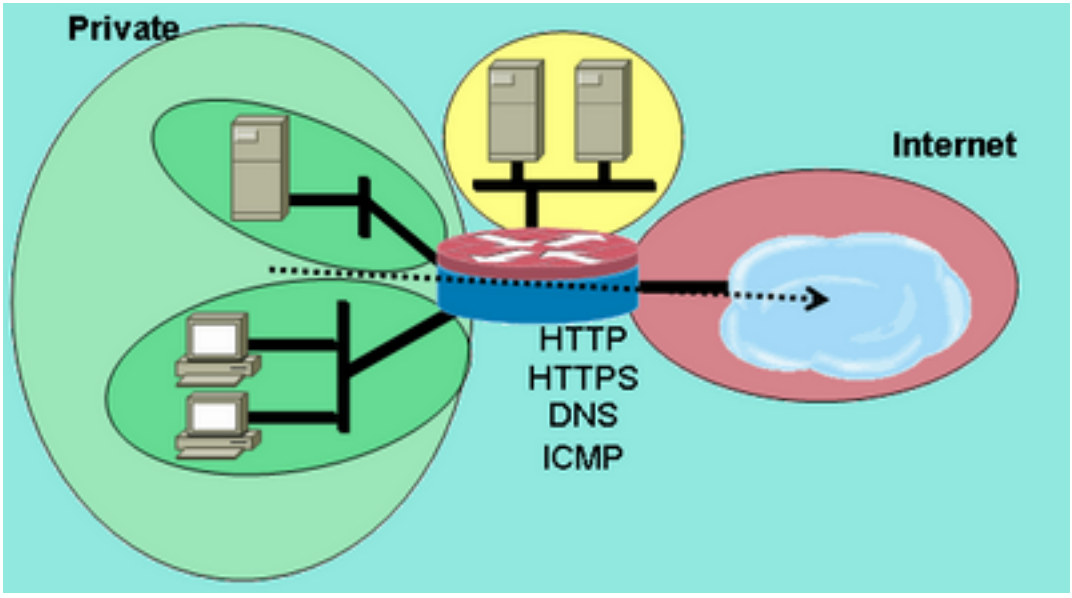
컨피그레이션의 일부를 서로 다른 시간에 서로 다른 네트워크 세그먼트에 적용하므로, 네트워크 세그먼트가 영역에 위치할 때 다른 세그먼트와의 연결이 끊어지는 것을 기억해야 합니다. 예를 들어, 프라이빗 영역이 구성된 경우, 프라이빗 영역의 호스트는 해당 정책이 정의될 때까지 DMZ 및 인터넷 영역에 대한 연결이 끊어집니다.

## 스테이트풀 인스펙션 라우팅 방화벽

### 개인 인터넷 정책 구성

그림 4는 사설 인터넷 정책의 컨피그레이션을 보여줍니다.

**그림 4: 개인 영역에서 인터넷 영역으로의 서비스 검사**




*개인 영역에서 인터넷 영역으로의 서비스 검사*

프라이빗 인터넷 정책은 HTTP, HTTPS, DNS에 레이어 4 검사를 적용하고 프라이빗 영역에서 인터넷 영역으로 ICMP에 대한 레이어 4 검사를 적용합니다. 이렇게 하면 프라이빗 영역에서 인터넷 영역으로의 연결이 허용되며 반환 트래픽이 허용됩니다. 레이어 7 검사는 애플리케이션 제어를 강화하고, 보안을 강화하며, 수정이 필요한 애플리케이션을 지원할 수 있다는 장점이 있습니다. 그러나 레이어 7 검사는 앞서 언급했듯이 영역 간에는 검사를 위해 구성되지 않은 레이어 7 프로토콜이 허용되지 않으므로 네트워크 활동을 더 정확하게 이해해야 합니다.

1. 앞서 설명한 정책을 기반으로 영역 간에 허용할 트래픽을 설명하는 클래스 맵을 정의합니다.
```
configure terminal
 class-map type inspect match-any internet-traffic-class
  match protocol http
  match protocol https
  match protocol dns
  match protocol icmp
```
2. 방금 정의한 클래스 맵에서 트래픽을 검사하도록 정책 맵을 구성합니다.

```
configure terminal
 policy-map type inspect private-internet-policy
  class type inspect internet-traffic-class
   inspect
```

3. 사설 및 인터넷 영역을 구성하고 해당 영역에 라우터 인터페이스를 할당합니다.

```
configure terminal
 zone security private
 zone security internet
 int bvi1
  zone-member security private
 int fastethernet 0
  zone-member security internet
```

영역 쌍을 구성하고 적절한 정책 맵을 적용합니다.

**참고:** 다음과 같이 인터넷 영역으로 이동하는 전용 영역에서 제공된 연결을 검사하려면 현재 전용 인터넷 영역 쌍만 구성해야 합니다.

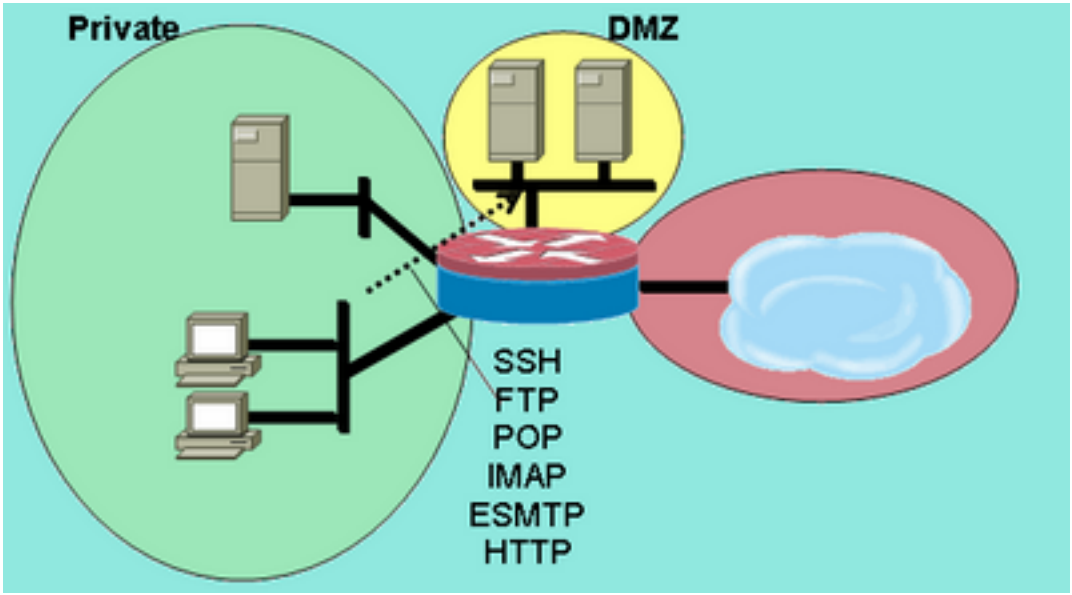
```
configure terminal
 zone-pair security private-internet source private destination internet
  service-policy type inspect private-internet-policy
```

이렇게 하면 클라이언트 영역에서 서버 영역으로 HTTP, HTTPS, DNS 및 ICMP 연결을 허용하고 HTTP 트래픽에 애플리케이션 검사를 적용하여 원치 않는 트래픽이 HTTP의 서비스 포트인 TCP 80을 통과하지 못하도록 하는 프라이빗 인터넷 영역 쌍에 대한 레이어 7 검사 정책의 구성이 완료됩니다.

### 프라이빗 DMZ 정책 구성

그림 5는 프라이빗 DMZ 정책의 컨피그레이션을 보여줍니다.

### 그림 5: 개인 영역에서 DMZ 영역으로 서비스 검사




*개인 영역에서 DMZ 영역으로 서비스 검사*

프라이빗 DMZ 정책은 영역 간 네트워크 트래픽을 더 잘 이해해야 하므로 복잡성이 추가됩니다. 이 정책은 프라이빗 영역의 레이어 7 검사를 DMZ에 적용합니다. 이렇게 하면 전용 영역에서 DMZ로의 연결이 허용되며 반환 트래픽이 허용됩니다. 레이어 7 검사는 애플리케이션 제어를 강화하고, 보안을 강화하며, 수정이 필요한 애플리케이션을 지원할 수 있다는 장점이 있습니다. 그러나 레이어 7

검사는 앞서 언급했듯이 영역 간에는 검사를 위해 구성되지 않은 레이어 7 프로토콜이 허용되지 않으므로 네트워크 활동을 더 정확하게 이해해야 합니다.

1. 앞서 설명한 정책을 기반으로 영역 간에 허용할 트래픽을 설명하는 클래스 맵을 정의합니다.
```
configure terminal
 class-map type inspect match-any L7-inspect-class
  match protocol ssh
  match protocol ftp
  match protocol pop
  match protocol imap
  match protocol esmtp
  match protocol http
```
2. 방금 정의한 클래스 맵의 트래픽을 검사하도록 정책 맵을 구성합니다.
```
configure terminal
 policy-map type inspect private-dmz-policy
  class type inspect L7-inspect-class
   inspect
```
3. 프라이빗 및 DMZ 영역을 구성하고 해당 영역에 라우터 인터페이스를 할당합니다.
```
configure terminal
 zone security private
 zone security dmz
 int bvi1
  zone-member security private
 int fastethernet 1
  zone-member security dmz
```
4. 영역 쌍을 구성하고 적절한 정책 맵을 적용합니다.

   **참고:** DMZ로 이동하는 전용 영역에서 제공된 연결을 검사하려면 다음과 같이 현재 전용 DMZ 영역 쌍만 구성해야 합니다.

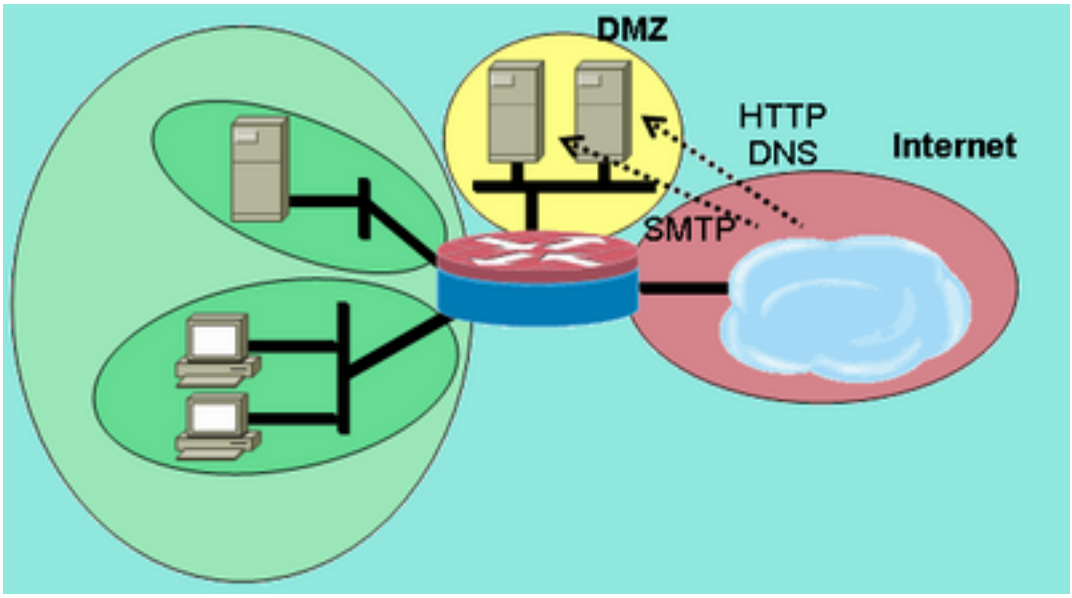
```
configure terminal
 zone-pair security private-dmz source private destination dmz
  service-policy type inspect private-dmz-policy
```

클라이언트 영역에서 서버 영역으로 모든 TCP, UDP 및 ICMP 연결을 허용하기 위해 사설 DMZ에서 레이어 7 검사 정책의 컨피그레이션을 완료합니다. 정책은 하위 채널에 대한 픽스업을 적용하지 않지만 대부분의 애플리케이션 연결을 수용하는 간단한 정책의 예를 제공합니다.

### 인터넷 DMZ 정책 구성

그림 6은 인터넷 DMZ 정책의 컨피그레이션을 보여줍니다.

### 그림 6: 인터넷 영역에서 DMZ 영역으로 서비스 검사



*인터넷 영역에서 DMZ 영역으로 서비스 검사*

이 정책은 인터넷 영역의 레이어 7 검사를 DMZ에 적용합니다. 이렇게 하면 인터넷 영역에서 DMZ로의 연결이 허용되며 DMZ 호스트에서 연결이 시작된 인터넷 호스트로의 반환 트래픽이 허용됩니다. 인터넷 DMZ 정책은 레이어 7 검사를 ACL에 의해 정의된 주소 그룹과 결합하여 특정 호스트, 호스트 그룹 또는 서브넷의 특정 서비스에 대한 액세스를 제한합니다. 이를 위해 IP 주소를 지정하기 위해 ACL을 참조하는 다른 클래스 맵 내의 서비스를 지정하는 클래스 맵을 중첩합니다.

1. 앞서 설명한 정책을 기반으로 영역 간에 허용할 트래픽을 설명하는 클래스 맵 및 ACL을 정의합니다.서로 다른 두 서버에 액세스하기 위해 서로 다른 액세스 정책이 적용되므로 서비스에 대해 여러 클래스 맵을 사용해야 합니다. 인터넷 호스트는 172.16.2.2에 대한 DNS 및 HTTP 연결이 허용되고 SMTP 연결은 172.16.2.3에 허용됩니다. 클래스 맵의 차이점을 확인하십시오. 서비스를 지정하는 클래스 맵은 match-any 키워드를 사용하여 나열된 서비스를 허용합니다. ACL을 서비스 클래스 맵과 연결하는 클래스 맵에서는 match-all 키워드를 사용하여 클래스 맵의 두 조건이 모두 충족되어야 트래픽이 허용되도록 합니다.
```
configure terminal
 access-list 110 permit ip any host 172.16.2.2
 access-list 111 permit ip any host 172.16.2.3
 class-map type inspect match-any dns-http-class
  match protocol dns
  match protocol http
 class-map type inspect match-any smtp-class
  match protocol smtp
 class-map type inspect match-all dns-http-acl-class
  match access-group 110
  match class-map dns-http-class
 class-map type inspect match-all smtp-acl-class
  match access-group 111
  match class-map smtp-class
```
2. 방금 정의한 클래스 맵의 트래픽을 검사하도록 정책 맵을 구성합니다.
```
configure terminal
 policy-map type inspect internet-dmz-policy
  class type inspect dns-http-acl-class
   inspect
  class type inspect smtp-acl-class
   inspect
```
3. 인터넷 및 DMZ 영역을 구성하고 해당 영역에 라우터 인터페이스를 할당합니다. 이전 섹션에서 설정한 경우 DMZ 컨피그레이션을 건너뜁니다.
```
configure terminal
zone security internet
zone security dmz
```

```
int fastethernet 0
 zone-member security internet
int fastethernet 1
 zone-member security dmz
```

4. 영역 쌍을 구성하고 적절한 정책 맵을 적용합니다.**참고:** 현재 인터넷 DMZ 영역 쌍만 구성하면 DMZ 영역으로 이동하는 인터넷 영역에서 제공된 연결을 검사할 수 있습니다(다음 참조).

```
configure terminal
 zone-pair security internet-dmz source internet destination dmz
  service-policy type inspect internet-dmz-policy
```

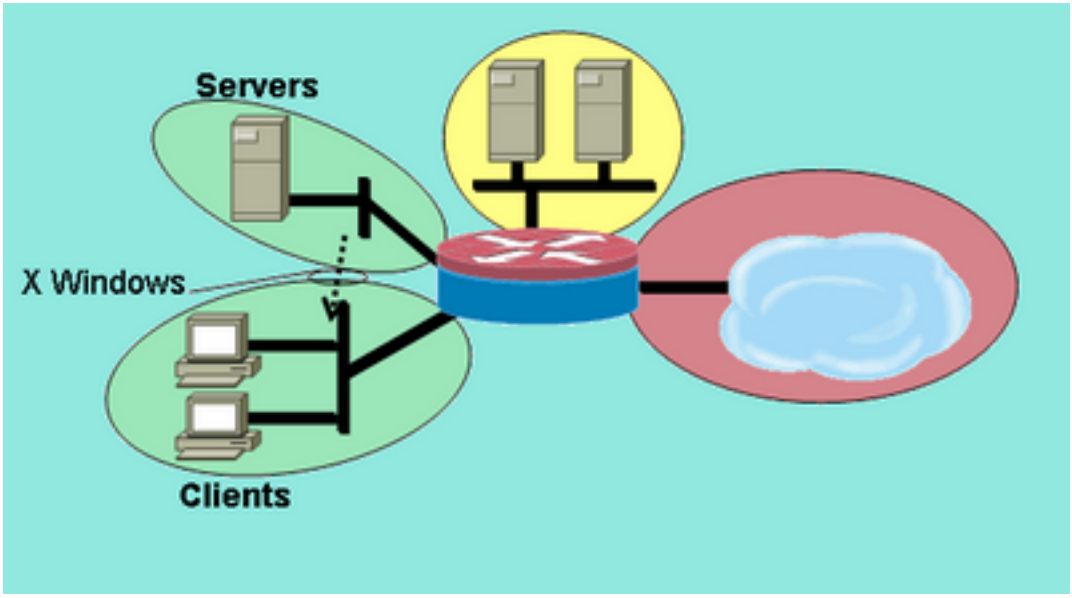
인터넷 DMZ 영역 쌍에 대한 주소별 레이어 7 검사 정책의 컨피그레이션을 완료합니다.

## 스테이트풀 인스펙션 투명 방화벽

### 서버-클라이언트 정책 구성

다음 그림에서는 서버-클라이언트 정책의 컨피그레이션을 보여줍니다.

**그림 7: 서버 영역에서 클라이언트 영역으로 서비스 검사**




*서버 영역에서 클라이언트 영역으로 서비스 검사*

서버-클라이언트 정책은 사용자 정의 서비스와 함께 검사를 적용합니다. 레이어 7 검사는 서버 영역에서 클라이언트 영역으로 적용됩니다. 이렇게 하면 서버 영역에서 클라이언트 영역까지 특정 포트 범위에 대한 X Windows 연결이 허용되고 반환 트래픽이 허용됩니다. X Windows는 PAM에서 기본적으로 지원되는 프로토콜이 아니므로 ZFW가 적절한 트래픽을 인식하고 검사할 수 있도록 PAM에서 사용자 구성 서비스를 정의해야 합니다.

IEEE 브리지 그룹에 2개 이상의 라우터 인터페이스가 구성되어 IRB(Integrated Routing and Bridging)를 제공하여 브리지 그룹의 인터페이스 간에 브리징을 제공하고 BVI(Bridge Virtual Interface)를 통해 다른 서브넷으로 라우팅합니다. 투명 방화벽 정책은 BVI를 통해 브리지 그룹을 나가는 트래픽이 아닌 "브리지를 통과하는" 트래픽에 방화벽 검사를 적용합니다. 검사 정책은 브리지 그룹을 가로지르는 트래픽에만 적용됩니다. 따라서 이 시나리오에서는 프라이빗 영역 내에 중첩된 클라이언트와 서버 영역 사이에서 이동하는 트래픽에만 검사가 적용됩니다. 프라이빗 영역과 퍼블릭 및 DMZ 영역 사이에 적용되는 정책은 트래픽이 BVI를 통해 브리지 그룹을 나갈 때만 적용됩니다. 클라이언트 또는 서버 영역에서 BVI를 통해 트래픽이 나갈 경우 투명 방화벽 정책이 호출되지 않습니다.

1. X Windows용 사용자 정의 항목으로 PAM을 구성합니다.X Windows 클라이언트(애플리케이

션이 호스팅되는 경우)는 포트 6900에서 시작되는 범위에서 클라이언트에 정보를 표시하기 위한 연결을 엽니다(사용자가 작업하는 경우).각 추가 연결에서는 연속된 포트를 사용하므로 클라이언트가 하나의 호스트에 10개의 서로 다른 세션을 표시할 경우 서버는 포트 6900-6909를 사용합니다. 따라서 6900~6909의 포트 범위를 검사할 경우 6909를 초과하는 포트에 열린 연결은 실패합니다.

```
configure terminal
 ip port-map user-Xwindows port tcp from 6900 to 6910
```

2. PAM 문서를 검토하여 추가 PAM 질문을 해결하거나 PAM과 Cisco IOS Firewall 상태 기반 검사 간의 상호 운용성에 대한 자세한 내용은 세부적인 프로토콜 검사 문서를 확인하십시오.
3. 앞서 설명한 정책을 기반으로 영역 간에 허용할 트래픽을 설명하는 클래스 맵을 정의합니다.

```
configure terminal
 class-map type inspect match-any Xwindows-class
  match protocol user-Xwindows
```

4. 방금 정의한 클래스 맵의 트래픽을 검사하도록 정책 맵을 구성합니다.

```
configure terminal
 policy-map type inspect servers-clients-policy
  class type inspect Xwindows-class
   inspect
```

5. 클라이언트 및 서버 영역을 구성하고 해당 영역에 라우터 인터페이스를 할당합니다.Clients-Servers Policy Configuration 섹션에서 이러한 영역을 구성하고 인터페이스를 할당한 경우 영역 쌍 정의로 건너뛸 수 있습니다. 완전성을 위해 브리징 IRB 컨피그레이션이 제공됩니다.

```
configure terminal
bridge irb
bridge 1 protocol ieee
bridge 1 route ip
zone security clients
zone security servers
 int vlan 1
 bridge-group 1
 zone-member security clients
int vlan 2
 bridge-group 1
 zone-member security servers
```

6. 영역 쌍을 구성하고 적절한 정책 맵을 적용합니다.**참고:** 현재 서버-클라이언트 영역 쌍만 구성하면 다음과 같이 클라이언트 영역으로 이동하는 서버 영역에서 시작된 연결을 검사할 수 있습니다.

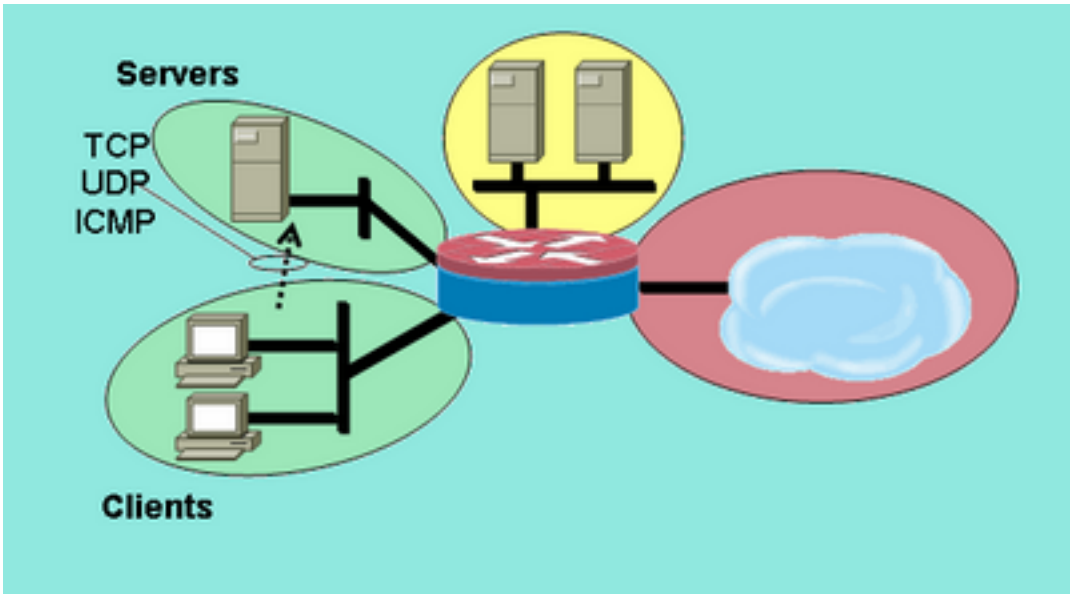
```
configure terminal
 zone-pair security servers-clients source servers destination clients
  service-policy type inspect servers-clients-policy
```

이렇게 하면 서버 영역에서 클라이언트 영역으로 X Windows 연결을 허용하기 위해 servers-clients 영역 쌍에서 사용자 정의 검사 정책의 컨피그레이션이 완료됩니다.

### 클라이언트-서버 정책 구성

그림 8은 클라이언트-서버 정책의 컨피그레이션을 보여줍니다.

**그림 8: 클라이언트 영역에서 서버 영역으로 서비스 검사**



*클라이언트 영역에서 서버 영역으로 서비스 검사*

클라이언트-서버 정책은 다른 것보다 덜 복잡합니다. 레이어 4 검사는 클라이언트 영역에서 서버 영역으로 적용됩니다. 클라이언트 영역에서 서버 영역으로 연결을 허용하고 반환 트래픽을 허용합니다. 레이어 4 검사는 대부분의 애플리케이션 트래픽을 허용하는 데 몇 가지 규칙만 필요하다는 점에서 방화벽 컨피그레이션이 간소화되는 이점을 제공합니다. 그러나 레이어 4 검사에는 두 가지 주요 단점이 있습니다.

- FTP 또는 미디어 서비스와 같은 애플리케이션은 서버에서 클라이언트로 추가 하위 채널을 협상하는 경우가 많습니다. 이 기능은 일반적으로 제어 채널 대화 상자를 모니터링하고 하위 채널을 허용하는 서비스 픽스업에 포함됩니다. 이 기능은 레이어 4 검사에서 사용할 수 없습니다.
- 레이어 4 검사는 거의 모든 애플리케이션 레이어 트래픽을 허용합니다. 방화벽을 통해 소수의 애플리케이션만 허용되도록 네트워크 사용을 제어해야 하는 경우, 방화벽을 통해 허용되는 서비스를 제한하려면 아웃바운드 트래픽에 대해 ACL을 구성해야 합니다.

두 라우터 인터페이스는 IEEE 브리지 그룹에 구성되어 있으므로 이 방화벽 정책은 투명 방화벽 검사를 적용합니다. 이 정책은 IEEE IP 브리지 그룹의 두 인터페이스에 적용됩니다. 검사 정책은 브리지 그룹을 가로지르는 트래픽에만 적용됩니다. 클라이언트 및 서버 영역이 개인 영역 내에 중첩된 이유를 설명합니다.

1. 앞서 설명한 정책을 기반으로 영역 간에 허용할 트래픽을 설명하는 클래스 맵을 정의합니다.
```
configure terminal
 class-map type inspect match-any L4-inspect-class
 match protocol tcp
 match protocol udp
 match protocol icmp
```
2. 방금 정의한 클래스 맵의 트래픽을 검사하도록 정책 맵을 구성합니다.
```
configure terminal
 policy-map type inspect clients-servers-policy
 class type inspect L4-inspect-class
  inspect
```
3. 클라이언트 및 서버 영역을 구성하고 해당 영역에 라우터 인터페이스를 할당합니다.
```
configure terminal
 zone security clients
 zone security servers
 interface vlan 1
  zone-member security clients
 interface vlan 2
  zone-member security servers
```
4. 영역 쌍을 구성하고 적절한 정책 맵을 적용합니다.**참고:** 현재 클라이언트-서버 영역-쌍을 구성

하기만 하면 다음과 같이 서버 영역으로 이동하는 클라이언트 영역에서 제공된 연결을 검사할 수 있습니다.

```
configure terminal
 zone-pair security clients-servers source clients destination servers
  service-policy type inspect clients-servers-policy
```

클라이언트 영역에서 서버 영역으로 모든 TCP, UDP 및 ICMP 연결을 허용하기 위해 클라이언트-서버 영역 쌍에 대한 레이어 4 검사 정책 컨피그레이션을 완료합니다. 정책은 하위 채널에 대한 수정을 적용하지 않지만 대부분의 애플리케이션 연결을 수용하기 위한 간단한 정책의 예를 제공합니다.

## 영역 기반 정책 방화벽에 대한 속도 정책

데이터 네트워크는 특정 유형의 네트워크 트래픽의 전송 속도를 제한하고, 우선순위가 낮은 트래픽의 영향을 업무상 필수적인 트래픽으로 제한하는 기능을 통해 자주 이점을 제공합니다. Cisco IOS 소프트웨어는 트래픽 명목상 속도 및 버스트를 제한하는 트래픽 폴리싱과 함께 이 기능을 제공합니다. Cisco IOS Software는 Cisco IOS 릴리스 12.1(5)T부터 트래픽 폴리싱을 지원했습니다.

Cisco IOS Software Release 12.4(9)T는 특정 클래스 맵이 방화벽을 한 보안 영역에서 다른 보안 영역으로 이동할 때 해당 클래스 맵의 정의와 일치하는 트래픽을 폴리싱하는 기능을 추가할 때 속도 제한을 통해 ZFW를 보강합니다. 이를 통해 특정 트래픽을 설명하고 방화벽 정책을 적용하며 트래픽 대역폭 소비를 폴리싱할 수 있는 편리한 단일 컨피그레이션 포인트를 제공합니다. ZFW는 정책 적합성을 위해 전송되는 작업과 정책 위반을 위해 삭제되는 작업만 제공한다는 점에서 인터페이스 기반과 다릅니다. ZFW에서 DSCP에 대한 트래픽을 표시할 수 없습니다.

ZFW는 대역폭 사용을 바이트/초로만 지정할 수 있으며 패킷/초는 제공되지 않습니다. ZFW는 인터페이스 기반 또는 인터페이스 기반 없이 적용할 수 있습니다. 따라서 추가 기능이 필요한 경우 인터페이스 기반으로 이러한 기능을 적용할 수 있습니다. 인터페이스 기반을 방화벽과 함께 사용하는 경우 정책이 충돌하지 않는지 확인합니다.

### ZFW 정책 구성

ZFW 폴리싱은 정책 맵 클래스 맵의 트래픽을 초당 8,000비트에서 2,000,000,000비트 사이의 사용자 정의 속도 값으로 제한하며, 구성 가능한 버스트 값은 1,000~512,000,000바이트 범위입니다.

ZFW 폴리싱은 정책 맵의 추가 컨피그레이션 행에 의해 구성되며, 이는 정책 작업 후에 적용됩니다.

```
policy-map type inspect private-allowed-policy
 class type inspect http-class
  inspect
  police rate [bps rate value <8000-2000000000>] burst [value in bytes <1000-512000000>]
```

### 세션 제어

또한 ZFW 정책은 클래스 맵과 일치하는 정책 맵에서 트래픽에 대한 세션 수를 제한하기 위해 세션 제어를 도입했습니다. 이는 현재 클래스 맵당 DoS 보호 정책을 적용하는 기능에 추가됩니다. 이를 통해 Zone-Pair를 통과하는 지정된 클래스 맵과 일치하는 세션 수를 세부적으로 제어할 수 있습니다. 여러 정책 맵 또는 영역 쌍에서 동일한 클래스 맵을 사용하는 경우 다양한 클래스 맵 애플리케이션에 서로 다른 세션 제한을 적용할 수 있습니다.

세션 제어는 원하는 세션 볼륨을 포함하는 매개변수 맵이 구성된 경우 적용되며, 매개변수 맵은 정책 맵 아래의 클래스 맵에 적용된 검사 작업에 추가됩니다.

```
parameter-map type inspect my-parameters
 sessions maximum [1-2147483647]

policy-map type inspect private-allowed-policy
 class type inspect http-class
  inspect my-parameters
```

매개변수 맵은 inspect 작업에만 적용할 수 있으며 pass 또는 drop 작업에서는 사용할 수 없습니다.

ZFW 세션 제어 및 폴리싱 활동은 다음 명령을 사용하여 볼 수 있습니다.

```
show policy-map type inspect zone-pair
```

## 애플리케이션 검사

애플리케이션 검사에는 ZFW에 대한 추가 기능이 도입되었습니다. 애플리케이션 검사 정책은 OSI 모델의 레이어 7에 적용됩니다. 여기서 사용자 애플리케이션은 유용한 기능을 제공할 수 있는 메시지를 보내고 받습니다. 일부 애플리케이션은 원치 않거나 취약한 기능을 제공할 수 있으므로 이러한 기능과 연결된 메시지를 필터링하여 애플리케이션 서비스에 대한 작업을 제한해야 합니다.

Cisco IOS Software ZFW는 다음 애플리케이션 서비스에 대한 애플리케이션 검사 및 제어 기능을 제공합니다.

- HTTP
- SMTP
- POP3
- IMAP
- Sun RPC
- P2P 애플리케이션 트래픽
- IM 애플리케이션

AIC(Application Inspection and Control)는 서비스마다 기능이 다릅니다. HTTP 검사는 여러 유형의 애플리케이션 활동에 대한 세분화된 필터링을 제공하며, 애플리케이션 동작 표준을 준수하도록 하고 서비스를 통해 전송되는 콘텐츠 유형을 제한하기 위해 전송 크기, 웹 주소 길이 및 브라우저 활동을 제한하는 기능을 제공합니다. SMTP용 AIC는 콘텐츠 길이를 제한하고 프로토콜 규정준수를 적용할 수 있습니다. POP3 및 IMAP 검사는 사용자가 보안 인증 메커니즘을 사용하여 사용자 자격 증명이 손상되지 않도록 하는 데 도움이 됩니다.

애플리케이션 검사는 애플리케이션별 클래스 맵 및 정책 맵의 추가 집합으로 구성되며, 이는 검사 정책 맵에서 애플리케이션 서비스 정책을 정의할 때 현재 검사 클래스 맵 및 정책 맵에 적용됩니다.

## HTTP 애플리케이션 검사

애플리케이션 검사는 HTTP 트래픽에 적용되어 IM, P2P 파일 공유, TCP 80을 통해 방화벽 애플리케이션을 리디렉션할 수 있는 터널링 애플리케이션과 같은 다른 애플리케이션에 대해 HTTP 서비스 포트의 원치 않는 사용을 제어할 수 있습니다.

허용된 HTTP 트래픽을 위반하는 트래픽을 설명하기 위해 애플리케이션 검사 클래스 맵을 구성합니다.

```
! configure the actions that are not permitted
```

```
class-map type inspect http match-any http-aic-cmap
 match request port-misuse any
 match req-resp protocol-violation
! define actions to be applied to unwanted traffic
policy-map type inspect http http-aic-pmap
 class type insp http http-aic-cmap
  reset
  log
! define class-map for stateful http inspection
class-map type inspect match-any http-cmap
 match protocol http
! define class-map for stateful inspection for other traffic
class-map type inspect match-any other-traffic-cmap
 match protocol smtp
 match protocol dns
 match protocol ftp
! define policy-map, associate class-maps and actions
policy-map type inspect priv-pub-pmap
 class type inspect http-cmap
  inspect
  service-policy http http-aic-pmap
 class type inspect other-traffic-cmap
  inspect
```

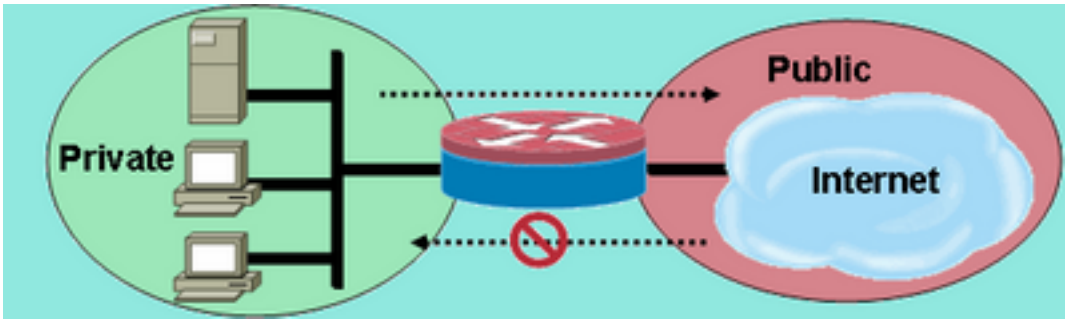
### HTTP 애플리케이션 검사 개선

Cisco IOS Software Release 12.4(9)T에서는 ZFW HTTP 검사 기능이 개선되었습니다. Cisco IOS Firewall은 Cisco IOS Software 릴리스 12.3(14)T에서 HTTP 애플리케이션 검사를 도입했습니다. Cisco IOS Software 릴리스 12.4(9)T는 다음을 추가할 때 현재 기능을 보강합니다.

- 헤더 이름 및 헤더 값을 기반으로 요청과 응답을 허용, 거부 및 모니터링하는 기능. 이는 취약한 헤더 필드를 전달하는 요청과 응답을 차단하는 데 유용합니다.
- HTTP 요청 및 응답 헤더(예: 최대 URL 길이, 최대 헤더 길이, 최대 헤더 수, 최대 헤더 라인 길이)에서 서로 다른 요소의 크기를 제한하는 기능은 버퍼 오버플로를 방지하는 데 유용합니다.
- 동일한 유형의 여러 헤더를 전달하는 요청과 응답을 차단할 수 있는 기능 예를 들어, 두 개의 content-length 헤더가 있는 요청.
- 비 ASCII 헤더로 요청 및 응답을 차단하는 기능. 이 기능은 웹 서버에 웜과 기타 악성 콘텐츠를 전달하기 위해 이진 및 기타 비 ASCII 문자를 사용하는 다양한 공격을 방지하는 데 유용합니다.
- HTTP 메서드를 사용자 지정 범주로 그룹화하는 기능과 각 그룹을 차단/허용/모니터링하는 유연성을 제공합니다. HTTP RFC는 제한된 HTTP 메서드 집합을 허용합니다. 일부 표준 방법은 웹 서버에서 취약성을 악용하는 데 사용될 수 있으므로 안전하지 않은 것으로 간주됩니다. 비표준 방식의 상당수는 보안 기록이 좋지 않다.
- 사용자가 구성한 정규식을 기반으로 특정 URI를 차단하는 메서드입니다. 이 기능은 사용자에게 사용자 지정 URI 및 쿼리를 차단하는 기능을 제공합니다.
- 사용자 지정 가능한 문자열로 헤더 유형(특히 서버 헤더 유형)을 스푸핑하는 기능. 이는 공격자가 웹 서버 응답을 분석하고 가능한 많은 정보를 학습한 다음 특정 웹 서버의 취약점을 악용하는 공격을 실행하는 경우에 유용합니다.
- 하나 이상의 HTTP 매개 변수 값이 사용자가 정규식으로 입력한 값과 일치하는 경우 HTTP 연결을 차단하거나 HTTP 연결에 대해 알림을 보내는 기능. 가능한 HTTP 값 컨텍스트 중 일부는 헤더, 본문, 사용자 이름, 비밀번호, 사용자 에이전트, 요청 라인, 상태 라인 및 디코딩된 CGI 변수를 포함합니다.

HTTP 애플리케이션 검사 개선의 컨피그레이션 예는 그림 9와 같이 단순한 네트워크를 전제로 합니다.

**그림 9: 애플리케이션 검사에서 간단한 네트워크 가정**



*애플리케이션 검사에서 간단한 네트워크 가정*

방화벽은 트래픽을 두 클래스로 그룹화합니다.

- HTTP 트래픽
- 기타 모든 단일 채널 TCP, UDP 및 ICMP 트래픽

HTTP는 웹 트래픽에 대한 특정 검사를 허용하도록 분리됩니다. 이렇게 하면 이 문서의 첫 번째 섹션에서 폴리싱을 구성하고 두 번째 섹션에서 HTTP 애플리케이션 검사를 구성할 수 있습니다. 이 문서의 세 번째 섹션에서 P2P 및 IM 트래픽에 대한 특정 클래스 맵 및 정책 맵을 구성할 수 있습니다. 전용 영역에서 공용 영역으로의 연결이 허용됩니다. 공용 영역에서 전용 영역으로의 연결은 제공되지 않습니다.

초기 정책을 구현하는 전체 컨피그레이션은 부록 C를 참조하십시오.

### HTTP 애플리케이션 검사 개선 사항 구성

HTTP 애플리케이션 검사(및 기타 애플리케이션 검사 정책)에는 기본 레이어 4 구성보다 복잡한 구성이 필요합니다. 제어하려는 특정 트래픽을 인식하고 바람직하지 않은 바람직한 트래픽에 원하는 작업을 적용하려면 레이어 7 트래픽 분류 및 정책을 구성해야 합니다.

HTTP 애플리케이션 검사(다른 애플리케이션 검사 유형과 유사)는 HTTP 트래픽에만 적용할 수 있습니다. 따라서 특정 HTTP 트래픽에 대해 Layer 7 클래스 맵과 정책 맵을 정의한 다음 HTTP에 대해 Layer 4 클래스 맵을 정의하고 Layer-7 정책을 Layer-4 정책 맵의 HTTP 검사에 적용해야 합니다.

```
!configure the layer-7 traffic characteristics:
class-map type inspect http match-any http-l7-cmap
 match  req-resp protocol-violation
 match  request body length gt 4096
!
!configure the action to be applied to the traffic
!matching the specific characteristics:
policy-map type inspect http http-l7-pmap
 class type inspect http http-l7-cmap
  reset
  log
!
!define the layer-4 inspection policy
class-map type inspect match-all http-l4-cmap
 match protocol http
!
!associate layer-4 class and layer-7 policy-map
!in the layer-4 policy-map:
policy-map type inspect private-allowed-policy
 class type inspect http-l4-cmap
  inspect
  service-policy http http-l7-pmap
```

이러한 모든 HTTP 애플리케이션 검사 트래픽 특성은 Layer 7 클래스 맵에서 정의됩니다.

- 헤더 검사 명령은 헤더가 구성된 정규식과 일치하는 요청 또는 응답을 허용/거부/모니터링하는 기능을 제공합니다. 클래스 맵 기준과 일치하는 요청 또는 응답에 허용 또는 재설정 작업을 적용할 수 있습니다. 로그 작업을 추가하면 syslog 메시지가 생성됩니다.
  APPFW-6-HTTP_HDR_REGEX_MATCHED
  명령 사용:

```
match {request|response|req-resp} header regex <parameter-map-name>
```

샘플 활용 사례

- 헤더에 비 ASCII 문자가 포함된 요청 또는 응답을 차단하도록 http appfw 정책을 구성합니다.

```
parameter-map type regex non_ascii_regex
   pattern “[^\x00-\x80]”
class-map type inspect http non_ascii_cm
   match req-resp header regex non_ascii_regex
policy-map type inspect http non_ascii_pm
   class type inspect http non_ascii_cm
      reset
```

헤더 길이 검사 — 이 명령은 요청 또는 응답 헤더의 길이를 확인하고 길이가 구성된 임계값을 초과하는 경우 작업을 적용합니다. 작업이 허용되거나 재설정됩니다. 로그 작업을 추가하면 syslog 메시지가 생성됩니다.

```
APPFW-4- HTTP_HEADER_LENGTH
```

명령 사용:

```
match {request|response|req-resp} header length gt <bytes>
```

샘플 활용 사례

헤더 길이가 4096바이트보다 큰 요청 및 응답을 차단하도록 http appfw 정책을 구성합니다.

```
class-map type inspect http hdr_len_cm
   match req-resp header length gt 4096

policy-map type inspect http hdr_len_pm
   class type inspect http hdr_len_cm
      reset
```

헤더 카운트 검사 — 이 명령은 요청/응답의 헤더 라인(필드) 수를 확인하고, 카운트가 구성된 임계값을 초과하면 작업을 적용합니다. 작업이 허용되거나 재설정됩니다. 로그 작업을 추가하면 syslog 메시지가 생성됩니다.

```
APPFW-6- HTTP_HEADER_COUNT
```

명령 사용:

```
match {request|response|req-resp} header count gt <number>
```

샘플 활용 사례

16개가 넘는 헤더 필드가 있는 요청을 차단하도록 http appfw 정책을 구성합니다.

```
class-map type inspect http hdr_cnt_cm
   match request header count gt 16

policy-map type inspect http hdr_cnt_pm
   class type inspect http hdr_cnt_cm
      reset
```

헤더 필드 검사 — 이 명령은 특정 HTTP 헤더 필드 및 값을 포함하는 요청/응답을 허용/거부/모니터링하는 기능을 제공합니다. 클래스 맵 기준과 일치하는 요청 또는 응답에 허용 또는 재설정 작업을 적용할 수 있습니다. log 작업을 추가하면 syslog 메시지가 생성됩니다.

```
APPFW-6- HTTP_HDR_FIELD_REGEX_MATCHED
```

명령 사용:

```
match {request|response|req-resp} header <header-name>
```

샘플 활용 사례

스파이웨어/애드웨어를 차단하려면 HTTP 애플리케이션 검사 정책을 구성합니다.

```
parameter-map type regex ref_regex
   pattern “\.delfinproject\.com”
   pattern “\.looksmart\.com”

parameter-map type regex host_regex
   pattern “secure\.keenvalue\.com”
   pattern “\.looksmart\.com”

parameter-map type regex usragnt_regex
   pattern “Peer Points Manager”

class-map type inspect http spy_adwr_cm
   match request header refer regex ref_regex
   match request header host regex host_regex
   match request header user-agent regex usragnt_regex

policy-map type inspect http spy_adwr_pm
   class type inspect http spy_adwr_cm
      reset
```

헤더 필드 길이 검사 — 이 명령은 헤더 필드 줄의 길이를 제한하는 기능을 제공합니다. 클래스 맵 기준과 일치하는 요청 또는 응답에 허용 또는 재설정 작업을 적용할 수 있습니다. log 작업을 추가하면 syslog 메시지가 생성됩니다.

```
APPFW-6- HTTP_HDR_FIELD_LENGTH
```

명령 사용:

```
match {request|response|req-resp} header <header-name> length gt <bytes>
```

샘플 활용 사례

쿠키 및 사용자 에이전트 필드 길이가 각각 256 및 128을 초과하는 요청을 차단하도록 http appfw 정책을 구성합니다.

```
class-map type inspect http hdrline_len_cm
   match request header cookie length gt 256
   match request header user-agnet length gt 128

policy-map type inspect http hdrline_len_pm
   class type inspect http hdrline_len_cm
      reset
```

헤더 필드 반복 검사 — 이 명령은 요청 또는 응답에서 헤더 필드가 반복되었는지 확인합니다. 클래스 맵 기준과 일치하는 요청 또는 응답에 허용 또는 재설정 작업을 적용할 수 있습니다. 활성화되면 log 작업으로 syslog 메시지가 생성됩니다.

```
APPFW-6- HTTP_REPEATED_HDR_FIELDS
```

명령 사용:

```
match {request|response|req-resp} header <header-name>
```

샘플 활용 사례

여러 content-length 헤더 라인이 있는 요청 또는 응답을 차단하도록 http appfw 정책을 구성합니다. 세션 밀수를 방지하는 데 사용되는 가장 유용한 기능 중 하나입니다.

```
class-map type inspect http multi_occrns_cm
   match req-resp header content-length count gt 1

policy-map type inspect http multi_occrns_pm
   class type inspect http multi_occrns_cm
      reset
```

- 메서드 검사 — HTTP RFC는 제한된 HTTP 메서드 집합을 허용합니다. 그러나 일부 표준 방법을 사용하여 웹 서버에서 취약성을 악용할 수 있으므로 일부 표준 방법도 안전하지 않은 것으로 간주됩니다. 많은 비표준 방식이 악의적인 활동에 자주 사용됩니다. 이는 방법들을 다양한 카테고리들로 그룹화하고 사용자가 각각의 카테고리에 대한 액션을 선택하도록 하는 필요성을 필요로 한다. 이 명령은 안전한 메서드, 안전하지 않은 메서드, webdav 메서드, RFC 메서드 및 확장 메서드와 같은 다양한 범주로 메서드를 그룹화할 수 있는 유연한 방법을 제공합니다. 클래스 맵 기준과 일치하는 요청 또는 응답에 허용 또는 재설정 작업을 적용할 수 있습니다. 로그 작업을 추가하면 syslog 메시지가 생성됩니다.

```
APPFW-6-HTTP_METHOD
```

명령 사용:

```
match request method <method>
```

샘플 활용 사례

HTTP 메서드를 세 가지 범주로 그룹화하는 http appfw 정책을 구성합니다. 안전하고 안전하지 않으며 webdav 다음 표에 나와 있습니다. 다음과 같은 작업을 구성합니다.

- 안전한 모든 방법은 로그 없이 허용됩니다.
- 안전하지 않은 모든 메서드는 로그에 허용됩니다.
- 모든 webdav 메서드는 로그로 차단됩니다.

**안전 안전하지 않음 웹데브**

GET, HEAD, 옵션 포스트, 풋, 연결, 추적 BCOPY, BDELETE, BMOVE

```
http policy:

class-map type inspect http safe_methods_cm
   match request method get
   match request method head
   match request method option

class-map type inspect http unsafe_methods_cm
   match request method post
   match request method put
   match request method connect
   match request method trace

class-map type inspect http webdav_methods_cm
   match request method bcopy
   match request method bdelete
   match request method bmove

policy-map type inspect http methods_pm
   class type inspect http safe_methods_cm
      allow
   class type inspect http unsafe_methods_cm
      allow log
      class type inspect http webdav_methods_cm
      reset log
```

URI 검사— 이 명령은 구성된 일반 검사와 URI가 일치하는 요청을 허용/거부/모니터링할 수 있는 기능을 제공합니다. 이를 통해 사용자는 맞춤형 URL 및 쿼리를 차단할 수 있습니다. 클래스 맵 기준과 일치하는 요청 또는 응답에 허용 또는 재설정 작업을 적용할 수 있습니다. 로그 작업을 추가하면 syslog 메시지가 생성됩니다.

```
APPFW-6- HTTP_URI_REGEX_MATCHED
```

명령 사용:

```
match request uri regex <parameter-map-name>
```

샘플 활용 사례

URI가 다음 정규식과 일치하는 요청을 차단하도록 http appfw 정책을 구성합니다.

- .*cmd.exe
- .*섹스
- .*도박

```
parameter-map type regex uri_regex_cm
   pattern “.*cmd.exe”
   pattern “.*sex”
   pattern “.*gambling”

class-map type inspect http uri_check_cm
   match request uri regex uri_regex_cm

policy-map type inspect http uri_check_pm
   class type inspect http uri_check_cm
      reset
```

- URI 길이 검사 — 이 명령은 요청에서 전송된 URI의 길이를 확인하고, 길이가 구성된 임계값을 초과할 경우 구성된 작업을 적용합니다. 클래스 맵 기준과 일치하는 요청 또는 응답에 허용 또는 재설정 작업을 적용할 수 있습니다. 로그 작업을 추가하면 syslog 메시지가 생성됩니다.
  APPFW-6- HTTP_URI_LENGTH
  명령 사용:
  match request uri length gt <bytes>
  샘플 활용 사례

요청의 URI 길이가 3076바이트를 초과할 때마다 경보를 발생시키도록 http appfw 정책을 구성합니다.

```
class-map type inspect http uri_len_cm
   match request uri length gt 3076

policy-map type inspect http uri_len_pm
   class type inspect http uri_len_cm
      log
```

인수 검사 — 이 명령은 구성된 일반 검사와 일치하는 인수(매개변수)를 가진 요청을 허용, 거부 또는 모니터링할 수 있는 기능을 제공합니다. 클래스 맵 기준과 일치하는 요청 또는 응답에 허용 또는 재설정 작업을 적용할 수 있습니다. 로그 작업을 추가하면 syslog 메시지가 생성됩니다.

```
APPFW-6- HTTP_ARG_REGEX_MATCHED
```

명령 사용:

```
match request arg regex <parameter-map-name>
```

인수가 다음 정규식과 일치하는 요청을 차단하도록 http appfw 정책을 구성합니다.

- .*고려
- 공격

```
parameter-map type regex arg_regex_cm
   pattern “.*codered”
   pattern “.*attack”

class-map type inspect http arg_check_cm
   match request arg regex arg_regex_cm

policy-map type inspect http arg_check_pm
   class type inspect http arg_check_cm
      reset
```

- 인수 길이 검사 — 이 명령은 요청에서 전송된 인수의 길이를 확인하고, 길이가 구성된 임계값을 초과하면 구성된 작업을 적용합니다. 클래스 맵 기준과 일치하는 요청 또는 응답에 허용 또는 재설정 작업을 적용할 수 있습니다. 로그 작업을 추가하면 syslog 메시지가 생성됩니다.
  APPFW-6- HTTP_ARG_LENGTH
  명령 사용:
  match request arg length gt <bytes>
  샘플 활용 사례

요청의 인수 길이가 512바이트를 초과할 때마다 경보를 울리도록 http appfw 정책을 구성합니다.

```
class-map type inspect http arg_len_cm
   match request arg length gt 512
```

```
policy-map type inspect http arg_len_pm
   class type inspect http arg_len_cm
      log
```

- 본문 검사 — 이 CLI를 사용하면 요청 또는 응답의 본문과 일치시킬 정규식 목록을 지정할 수 있습니다. 클래스 맵 기준과 일치하는 요청 또는 응답에 허용 또는 재설정 작업을 적용할 수 있습니다. 로그 작업을 추가하면 syslog 메시지가 생성됩니다.
  `APPFW-6- HTTP_BODY_REGEX_MATCHED`
  명령 사용:
  `match {request|response|reg-resp} body regex <parameter-map-name>`
  샘플 활용 사례

본문에 `.*[Aa][Tt][Tt][Aa][Cc][Kk]` 포함된 응답을 차단하도록 http appfw를 구성합니다.

```
parameter-map type regex body_regex
   pattern ".*[Aa][Tt][Tt][Aa][Cc][Kk]"

class-map type inspect http body_match_cm
   match response body regex body_regex

policy-map type inspect http body_match_pm
   class type inspect http body_match_cm
      reset
```

본문(콘텐츠) 길이 검사 — 이 명령은 요청 또는 응답을 통해 전송되는 메시지의 크기를 확인합니다. 클래스 맵 기준과 일치하는 요청 또는 응답에 허용 또는 재설정 작업을 적용할 수 있습니다. 로그 작업을 추가하면 syslog 메시지가 생성됩니다.

`APPFW-4- HTTP_CONTENT_LENGTH`
명령 사용:

```
match {request|response|req-resp} body length lt <bytes> gt <bytes>
```

샘플 활용 사례

요청 또는 응답에서 10KB 이상의 메시지를 전달하는 http 세션을 차단하도록 http appfw 정책을 구성합니다.

```
class-map type inspect http cont_len_cm
   match req-resp header content-length gt 10240

policy-map type inspect http cont_len_pm
   class type inspect http cont_len_cm
      reset
```

상태 라인 검사 — 이 명령을 사용하면 사용자가 응답의 상태 라인과 일치시킬 정규식 목록을 지정할 수 있습니다. 클래스 맵 기준과 일치하는 요청 또는 응답에 허용 또는 재설정 작업을 적용할 수 있습니다. 로그 작업을 추가하면 syslog 메시지가 생성됩니다.

`APPFW-6-HTTP_STLINE_REGEX_MATCHED`
명령 사용:

```
match response status-line regex <class-map-name>
```

샘플 활용 사례

금지된 페이지에 액세스하려고 할 때마다 경보를 로깅하도록 http appfw를 구성합니다. 금지된 페이지에는 일반적으로 403 상태 코드가 포함되어 있으며, 상태 줄은 `HTTP/1.0 403  \r\n` 같습니다.

```
parameter-map type regex status_line_regex
   pattern “[Hh][Tt][Tt][Pp][/][0-9][.][0-9][ \t]+403”

class-map type inspect http status_line_cm
   match response status-line regex status_line_regex

policy-map type inspect http status_line_pm
   class type inspect http status_line_cm
      log
```

- Content-type inspection — 이 명령은 메시지 헤더의 content-type이 지원되는 콘텐츠 유형 목록에 있는지 확인합니다. 또한 헤더의 content-type이 메시지 데이터 또는 엔티티 본문 부분의 내용과 일치하는지 확인합니다. 키워드 불일치가 구성된 경우 명령은 요청 메시지의 허용 필드 값에 대해 응답 메시지의 콘텐츠 유형을 확인합니다. 클래스 맵 기준과 일치하는 요청 또는 응답에 허용 또는 재설정 작업을 적용할 수 있습니다. 로그 작업을 추가하면 적절한 syslog 메시지가 생성됩니다.

  ```
  APPFW-4- HTTP_CONT_TYPE_VIOLATION
  APPFW-4- HTTP_CONT_TYPE_MISMATCH
  APPFW-4- HTTP_CONT_TYPE_UNKNOWN
  ```

  명령 사용:

  ```
  match {request|response|req-resp} header content-type [mismatch|unknown|violation]
  ```

  샘플 활용 사례알 수 없는 content-type이 있는 요청과 응답을 전달하는 http 세션을 차단하도록 http appfw 정책을 구성합니다.

  ```
  class-map type inspect http cont_type_cm
     match req-resp header content-type unknown

  policy-map type inspect http cont_type_pm
     class type inspect http cont_type_cm
        reset
  ```

포트 오용 검사 — 이 명령은 IM, P2P, 터널링 등의 다른 애플리케이션에서 http 포트(80)가 오용되는 것을 방지하기 위해 사용됩니다. 허용 또는 재설정 작업을 클래스 맵 기준과 일치하는 요청 또는 응답에 적용할 수 있습니다. 로그 작업을 추가하면 적절한 syslog 메시지가 생성됩니다.

```
APPFW-4- HTTP_PORT_MISUSE_TYPE_IM
APPFW-4-HTTP_PORT_MISUSE_TYPE_P2P
APPFW-4-HTTP_PORT_MISUSE_TYPE_TUNNEL
```

명령 사용:

```
match request port-misuse {im|p2p|tunneling|any}
```

샘플 활용 사례

IM 애플리케이션에 잘못 사용되는 http 세션을 차단하도록 http appfw 정책을 구성합니다.

```
class-map type inspect http port_misuse_cm
   match request port-misuse im

policy-map type inspect http port_misuse_pm
   class type inspect http port_misuse_cm
      reset
```

- Strict-http 검사 — 이 명령은 HTTP 요청 및 응답에 대한 엄격한 프로토콜 적합성 검사를 활성

화합니다. 클래스 맵 기준과 일치하는 요청 또는 응답에 허용 또는 재설정 작업을 적용할 수 있습니다. 로그 작업을 추가하면 syslog 메시지가 생성됩니다.
APPFW-4- HTTP_PROTOCOL_VIOLATION
명령 사용:
match req-resp protocol-violation
샘플 활용 사례RFC 2616을 위반하는 요청 또는 응답을 차단하도록 http appfw 정책을 구성합니다.

```
class-map type inspect http proto-viol_cm
   match req-resp protocol-violation

policy-map type inspect http proto-viol_pm
   class type inspect http proto-viol_cm
      reset
```

- Transfer- Encoding Inspection — 이 명령은 전송 인코딩 유형이 구성된 유형과 일치하는 요청/응답을 허용, 거부 또는 모니터링하는 기능을 제공합니다. 클래스 맵 기준과 일치하는 요청 또는 응답에 허용 또는 재설정 작업을 적용할 수 있습니다. 로그 작업을 추가하면 syslog 메시지가 생성됩니다.
APPFW-6- HTTP_TRANSFER_ENCODING
명령 사용:
match {request|response|req-resp} header transfer-encoding
{regex <parameter-map-name> |gzip|deflate|chunked|identity|all}
샘플 활용 사례압축 유형 인코딩이 있는 요청 또는 응답을 차단하도록 http appfw 정책을 구성합니다.

```
class-map type inspect http trans_encoding_cm
   match req-resp header transfer-encoding type compress

policy-map type inspect http trans_encoding_pm
   class type inspect http trans_encoding_cm
      reset
```

- Java 애플릿 검사 — 이 명령은 응답에 Java 애플릿이 있는지 확인하고 애플릿이 탐지되면 구성된 작업을 적용합니다. 클래스 맵 기준과 일치하는 요청 또는 응답에 허용 또는 재설정 작업을 적용할 수 있습니다. 로그 작업을 추가하면 syslog 메시지가 생성됩니다.
APPFW-4- HTTP_JAVA_APPLET
명령 사용:
match response body java-applet
샘플 활용 사례Java 애플릿을 차단하도록 http appfw 정책을 구성합니다.

```
class-map type inspect http java_applet_cm
   match response body java-applet

policy-map type inspect http java_applet_pm
   class type inspect http java_applet_cm
      reset
```

## 인스턴트 메시징 및 피어 투 피어 애플리케이션 제어를 위한 ZFW 지원

### Cisco IOS Software 릴리스 12.4(9)T에는 IM 및 P2P 애플리케이션에 대한 ZFW 지원이 도입되었습니다.

Cisco IOS Software는 먼저 Cisco IOS Software 릴리스 12.4(4)T에서 IM 애플리케이션 제어에 대한 지원을 제공했습니다. ZFW의 초기 릴리스는 ZFW 인터페이스에서 IM 애플리케이션을 지원하지 않았습니다. IM 애플리케이션 제어가 필요한 경우, 사용자는 ZFW 컨피그레이션 인터페이스로 마이그레이션할 수 없습니다. Cisco IOS Software 릴리스 12.4(9)T에는 Yahoo! Messenger(YM), MSN Messenger(MSN) 및 AIM(AOL Instant Messenger)입니다. Cisco IOS Software Release 12.4(9)T는 P2P 파일 공유 애플리케이션을 위한 기본 Cisco IOS 방화벽 지원을 제공하는 Cisco

IOS 소프트웨어의 첫 번째 버전입니다.

IM 및 P2P 검사 모두 애플리케이션 트래픽에 대한 레이어 4 및 레이어 7 정책을 제공합니다. 즉, ZFW는 트래픽을 허용하거나 거부하기 위한 기본 상태 저장 검사를 제공할 수 있으며, 다양한 프로토콜의 특정 활동에 대한 세분화된 레이어 7 제어를 제공하여 특정 애플리케이션 활동은 허용하고 다른 활동은 거부할 수 있습니다.

### P2P 애플리케이션 검사 및 제어

SDM 2.2는 방화벽 컨피그레이션 섹션에 P2P 애플리케이션 제어를 도입했습니다. SDM은 NBAR(Network-Based Application Recognition) 및 QoS 정책을 적용하여 P2P 애플리케이션 활동을 탐지하여 라인 속도 0으로 관리하고 모든 P2P 트래픽을 차단했습니다. 이로 인해 Cisco IOS Firewall CLI에서 P2P를 지원해야 하는 CLI 사용자가 필요한 NBAR/QoS 컨피그레이션을 알지 못하면 CLI에서 P2P 차단을 구성할 수 없다는 문제가 제기되었습니다. Cisco IOS Software Release 12.4(9)T는 NBAR를 활용하여 P2P 애플리케이션 활동을 탐지할 수 있도록 ZFW CLI에 기본 P2P 제어를 도입합니다. 이 소프트웨어 릴리스는 여러 P2P 애플리케이션 프로토콜을 지원합니다.

- 비트토렌트
- 당나귀
- 패스트트랙
- 그누텔라
- 카자 / 카자A2
- WinMX

P2P 애플리케이션은 "포트 호핑(port-hopping)" 동작 및 탐지를 피하기 위한 기타 요령과 프로토콜의 동작을 수정하는 P2P 애플리케이션의 빈번한 변경 및 업데이트로 인해 발생하는 문제를 탐지하기가 특히 어렵습니다. ZFW는 기본 방화벽 상태 기반 검사를 NBAR의 트래픽 인식 기능과 결합하여 ZFW의 CPL 컨피그레이션 인터페이스에서 P2P 애플리케이션 제어를 제공합니다. NBAR는 두 가지 뛰어난 이점을 제공합니다.

- 탐지하기 어려운 복잡한 동작에도 애플리케이션을 인식하는 휴리스틱 기반 애플리케이션 인식 옵션
- 프로토콜 업데이트 및 수정 사항을 지속적으로 파악할 수 있는 업데이트 메커니즘을 제공하는 확장 가능한 인프라

### P2P 검사 구성

앞에서 언급한 것처럼 P2P 검사 및 제어는 레이어 4 상태 기반 검사 및 레이어 7 애플리케이션 제어를 모두 제공합니다. 네이티브 애플리케이션 서비스 포트에 대한 검사가 적합한 경우 레이어 4 검사는 다른 애플리케이션 서비스와 유사하게 구성됩니다.

```
class-map type inspect match-any my-p2p-class
match protocol [bittorrent | edonkey | fasttrack | gnutella | kazaa | kazaa2 | winmx ]
[signature (optional)]
!
policy-map type inspect private-allowed-policy
 class type inspect my-p2p-class
  [drop | inspect | pass]
```

match protocol [service-name]의 추가 서명 옵션을 확인합니다. match protocol 문의 끝에 signature 옵션이 추가되면 트래픽에 NBAR 휴리스틱이 적용되어 특정 P2P 애플리케이션 활동을 나타내는 트래픽의 텔테일을 검색합니다. 여기에는 포트 호핑 및 트래픽 탐지를 방지하기 위한 기타 애플리케이션 동작 변경이 포함됩니다. 이러한 수준의 트래픽 검사는 CPU 사용률을 높이고 네트워크 처

리량 기능을 줄인 가격에 제공됩니다. signature 옵션이 적용되지 않을 경우 NBAR 기반 휴리스틱 분석을 적용하여 포트 호핑 동작을 탐지하지 않으며 CPU 사용률이 동일한 범위에 영향을 미치지 않습니다.

네이티브 서비스 검사에서는 애플리케이션이 비표준 소스 및 대상 포트로 "홉"하는 경우 또는 애플리케이션이 인식되지 않는 포트 번호에서 작업을 시작하도록 업데이트되는 경우 P2P 애플리케이션에 대한 제어를 유지할 수 없다는 단점이 있습니다.

| **애플리케이션** | **네이티브 포트(12.4(15)T PAM 목록에서 인식됨)** |
|---|---|
| 비트토렌트 | TCP 6881-6889 |
| 전자 키 | TCP 4662 |
| 빠른 경로 | TCP 1214 |
| 그누텔라 | TCP 6346-6349 TCP 6355,5634 UDP 6346-6348 |
| 카자흐스탄2 | PAM에 의존 |
| Winmx | TCP 6699 |

P2P 트래픽을 허용(검사)하려면 추가 컨피그레이션을 제공해야 합니다. 일부 애플리케이션에서는 여러 P2P 네트워크를 사용하거나, 애플리케이션이 작동하도록 방화벽 컨피그레이션에 수용하는 데 필요한 특정 동작을 구현할 수 있습니다.

- BitTorrent 클라이언트는 일반적으로 일부 비표준 포트에서 실행되는 http를 통해 "추적기"(피어 디렉토리 서버)와 통신합니다. 이는 일반적으로 TCP 6969이지만 Torrent 전용 추적기 포트를 확인해야 합니다. BitTorrent를 허용하려는 경우 추가 포트를 수용하는 가장 좋은 방법은 HTTP를 match 프로토콜 중 하나로 구성하고 ip port-map 명령을 사용하여 HTTP에 TCP 6969를 추가하는 것입니다.
  ```
  ip port-map http port tcp 6969
  ```

  http 및 bittorrent를 클래스 맵에 적용된 일치 기준으로 정의해야 합니다.
- eDonkey는 eDonkey 및 Gnutella로 탐지되는 연결을 시작하는 것으로 나타납니다.
- KaZaA 검사는 NBAR 시그니처 탐지에 전적으로 의존합니다.

레이어 7(애플리케이션) 검사는 레이어 4 검사를 보완하여 파일 검색, 파일 전송, 텍스트 채팅 기능을 선택적으로 차단하거나 허용하는 등 서비스별 작업을 인식하고 적용할 수 있습니다. 서비스별 기능은 서비스별로 다릅니다.

P2P 애플리케이션 검사는 HTTP 애플리케이션 검사와 유사합니다.

```
!configure the layer-7 traffic characteristics:
class-map type inspect [p2p protocol] match-any p2p-l7-cmap
 match action
!
!configure the action to be applied to the traffic
!matching the specific characteristics:
policy-map type inspect [p2p protocol] p2p-l7-pmap
 class type inspect p2p p2p-l7-cmap
  [ reset | allow ]
  log
!
!define the layer-4 inspection policy
class-map type inspect match-all p2p-l4-cmap
 match protocol [p2p protocol]
!
!associate layer-4 class and layer-7 policy-map
!in the layer-4 policy-map:
policy-map type inspect private-allowed-policy
```

```
 class type inspect p2p-l4-cmap
  [ inspect | drop | pass ]
  service-policy p2p p2p-l7-pmap
```

P2P Application Inspection은 Layer 4 Inspection에서 지원하는 애플리케이션 하위 집합에 대해 애플리케이션별 기능을 제공합니다.

- 전자 키
- 빠른 경로
- 그누텔라
- 카자흐스탄2

이러한 각 애플리케이션은 다양한 애플리케이션별 일치 기준 옵션을 제공합니다.

전자 키

```
router(config)#class-map type inspect edonkey match-any edonkey-l7-cmap
router(config-cmap)#match ?
  file-transfer     Match file transfer stream
  flow              Flow based QoS parameters
  search-file-name  Match file name
  text-chat         Match text-chat
```

빠른 경로

```
router(config)#class-map type inspect fasttrack match-any ftrak-l7-cmap
router(config-cmap)#match ?
  file-transfer  File transfer stream
  flow           Flow based QoS parameters
```

그누텔라

```
router(config)#class-map type inspect gnutella match-any gtella-l7-cmap
router(config-cmap)#
```

카자흐스탄2

```
router(config)#class-map type inspect kazaa2 match-any kazaa2-l7-cmap
router(config-cmap)#match ?
  file-transfer  Match file transfer stream
  flow           Flow based QoS parameters
```

새로운 P2P 프로토콜 정의 또는 현재 P2P 프로토콜에 대한 업데이트는 NBAR의 동적 pdlm 업데이트 기능으로 로드할 수 있습니다. 새 PDLM을 로드하기 위한 컨피그레이션 명령입니다.

```
ip nbar pdlm <file-location>
```

새 프로토콜은 클래스 유형 inspect에 대한 match protocol 명령에서 사용할 수 있습니다. 새로운 P2P 프로토콜에 서비스(하위 프로토콜)가 있는 경우, 새로운 Layer 7 검사 클래스 맵 유형 및 Layer 7 일치 기준을 사용할 수 있게 됩니다.

### IM 애플리케이션 검사 및 제어

Cisco IOS Software 릴리스 12.4(4)T에는 IM Application Inspection and Control이 도입되었습니다.

IM 지원은 12.4(6)T에서 ZFW와 함께 도입되지 않았으므로 사용자는 ZFW와 레거시 방화벽 기능이 지정된 인터페이스에 공존할 수 없으므로 동일한 방화벽 정책에서 IM 제어와 ZFW를 적용할 수 없습니다.

Cisco IOS Software 릴리스 12.4(9)T는 다음 IM 서비스에 대한 상태 저장 검사 및 애플리케이션 제어를 지원합니다.

- AOL 인스턴트 메신저
- MSN 메신저
- 야후! 메신저

IM 검사는 대부분의 서비스마다 약간씩 다르며, IM 검사는 지정된 각 서비스에 대한 특정 호스트 그룹에 대한 액세스를 제어합니다. IM 서비스는 일반적으로 비교적 영구적인 디렉토리 서버 그룹에 의존하는데, 이 디렉토리 서버는 IM 서비스에 액세스하기 위해 클라이언트가 연락할 수 있어야 합니다. IM 애플리케이션은 프로토콜 또는 서비스 관점에서 제어하기가 매우 어려운 경향이 있습니다. 이러한 애플리케이션을 제어하는 가장 효과적인 방법은 고정 IM 서버에 대한 액세스를 제한하는 것입니다.

### IM 검사 구성

IM 검사 및 제어는 레이어 4 상태 기반 검사를 모두 제공합니다.

레이어 7 애플리케이션 제어.

레이어 4 검사는 다른 애플리케이션 서비스와 유사하게 구성됩니다.

```
class-map type inspect match-any my-im-class
match protocol [aol | msnmsgr | ymsgr ]
!
policy-map type inspect private-allowed-policy
 class type inspect my-im-class
  [drop | inspect | pass
```

IM 애플리케이션은 기능을 유지하기 위해 여러 포트의 서버에 연결할 수 있습니다. inspect 작업으로 지정된 IM 서비스를 허용하려면 IM 서비스 서버에 대한 허용된 액세스를 정의하는 server-list가 필요하지 않습니다. 그러나 AOL Instant Messenger와 같은 특정 IM 서비스를 지정하는 클래스 맵을 구성하고 관련 정책 맵에서 삭제 작업을 적용하면 IM 클라이언트에서 인터넷에 연결할 수 있는 다른 포트를 찾으려고 시도할 수 있습니다. 지정된 서비스에 대한 연결을 허용하지 않거나 IM 서비스 기능을 텍스트 채팅으로 제한하려면 ZFW가 IM 애플리케이션과 연결된 트래픽을 식별할 수 있도록 서버 목록을 정의해야 합니다.

```
!configure the server-list parameter-map:
parameter-map type protocol-info <name>
  server name <name>
  server ip a.b.c.d
  server ip range a.b.c.d a.b.c.d
```

예를 들어 Yahoo IM 서버 목록은 다음과 같이 정의됩니다.

```
parameter-map type protocol-info ymsgr-pmap
  server name scs.msg.yahoo.com
  server name scsd.msg.yahoo.com
  server ip 10.0.77.88
  server ip range 172.16.0.77 172.16.0.99
```

프로토콜 정의에 server-list를 적용해야 합니다.

```
class-map type inspect match-any ym-l4-cmap
 match protocol ymsgr ymsgr-pmap
```

이름 확인을 활성화하려면 ip domain lookup 및 ip name-server ip.ad.re.ss 명령을 구성해야 합니다 .

IM 서버 이름은 상당히 동적입니다. 구성된 IM 서버 목록이 완전하고 정확한지 정기적으로 확인해야 합니다.

Layer 7(애플리케이션) 검사는 Layer 4 검사를 강화하여 서비스별 작업(예: 텍스트 채팅 기능을 선택적으로 차단 또는 허용하고 다른 서비스 기능을 거부함)을 인식 및 적용할 수 있습니다.

IM Application Inspection은 현재 텍스트 채팅 활동과 다른 모든 애플리케이션 서비스를 구별할 수 있는 기능을 제공합니다. IM 활동을 텍스트 채팅으로 제한하려면 레이어 7 정책을 구성합니다.

```
class-map type inspect ymsgr match-any ymsgr-text-cmap
  match service text-chat

class-map type inspect ymsgr match-any ymsgr-default-cmap
  match service any

policy-map type inspect im ymsgr-l7-pmap
  class type inspect im ymsgr-text-cmap
    allow
    [log]
  class type inspect im ymsgr-text-cmap
    reset
    [log]
```

Yahoo!에 레이어 7 정책 적용 이전에 구성된 메신저 정책:

```
class-map type inspect match-any my-im-class
match protocol ymsgr
!
policy-map type inspect private-allowed-policy
 class type inspect my-im-class
  inspect
  service-policy im ymsgr-l7-pmap
```

## URL 필터

ZFW는 웹 콘텐츠에 대한 액세스를 라우터에 정의된 화이트리스트 또는 블랙리스트에서 지정한 것으로 제한하거나 특정 도메인에 대한 액세스를 확인하기 위해 도메인 이름을 URL 필터링 서버로 전달하는 URL 필터링 기능을 제공합니다. 애플리케이션 검사와 마찬가지로 Cisco IOS Software 릴리스 12.4(6)T에서 12.4(15)T로의 ZFW URL 필터링이 추가 정책 작업으로 적용됩니다.

서버 기반 URL 필터링의 경우 url 필터 서버 컨피그레이션을 설명하는 매개변수 맵을 정의해야 합니다.

```
parameter-map type urlfilter websense-parmap
 server vendor [n2h2 | websense] 10.1.1.1
```

고정 화이트리스트 또는 블랙리스트를 선호할 경우, 구체적으로 허용되거나 거부된 도메인 또는 하위 도메인 목록을 정의할 수 있으며, 목록과 일치하지 않는 트래픽에는 반대 작업이 적용됩니다.

```
parameter-map type urlfilter websense-parmap
 exclusive-domain deny .disallowed.com
 exclusive-domain permit .cisco.com
```
단독 도메인 정의에서 거부 옵션으로 URL 차단 목록을 정의하는 경우 다른 모든 도메인이 허용됩니다. "허용" 정의가 정의된 경우 IP 액세스 제어 목록의 기능과 마찬가지로 허용된 모든 도메인을 명시적으로 지정해야 합니다.

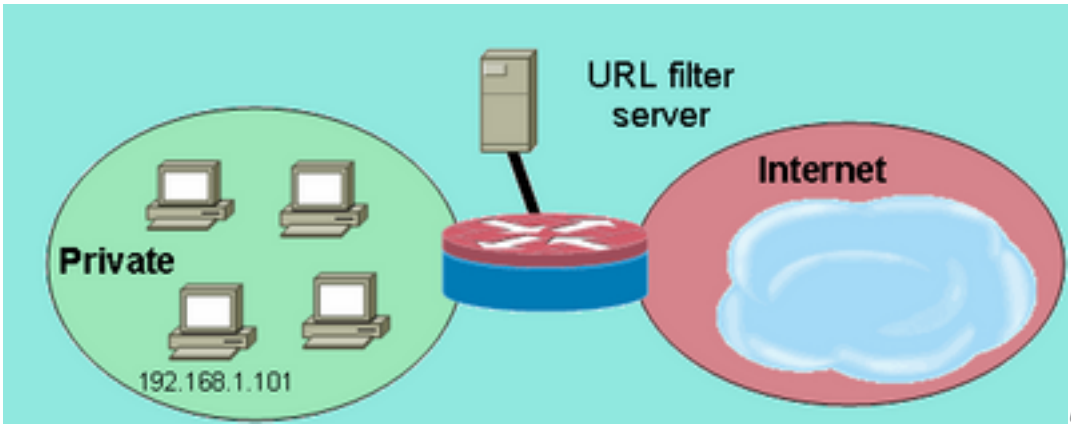
HTTP 트래픽과 일치하는 클래스 맵을 설정합니다.

```
class-map type inspect match-any http-cmap
 match protocol http
```
클래스 맵을 inspect 및 urlfilter 작업과 연결하는 정책 맵을 정의합니다.

```
policy-map type inspect http-filter-pmap
 class type inspect http-cmap
  inspect
  urlfilter websense-parmap
```
URL 필터링 서버와 통신하기 위한 최소 요구 사항을 구성합니다. 추가 URL 필터링 동작을 정의하는 데 몇 가지 옵션을 사용할 수 있습니다.

일부 네트워크 구축에서는 일부 호스트 또는 서브넷에 대해 URL 필터링을 적용하고 다른 호스트에 대해서는 URL 필터링을 우회하려고 합니다. 예를 들어, 그림 9에서 전용 영역의 모든 호스트는 특정 호스트 192.168.1.101을 제외하고 URL 필터 서버에서 HTTP 트래픽을 확인해야 합니다.

**그림 10: URL 필터링 예제 토폴로지**



*URL 필터링 예제 토폴로지*

이 작업은 두 개의 서로 다른 클래스 맵 맵을 정의하는 경우 수행할 수 있습니다.

- URL 필터링을 수신하는 대규모 호스트 그룹에 대한 HTTP 트래픽만 매칭하는 하나의 클래스 맵.
- URL 필터링을 수신하지 않는 소규모 호스트 그룹에 대한 하나의 클래스 맵. 두 번째 클래스 맵은 HTTP 트래픽은 물론 URL 필터링 정책에서 제외된 호스트 목록과 매칭합니다.

두 클래스 맵은 모두 정책 맵에서 구성되지만 하나의 클래스 맵에서만 urlfilter 작업을 수신합니다.

```
class-map type inspect match-any http-cmap
 match protocol http
class-map type inspect match-all http-no-urlf-cmap
 match protocol http
```

```
 match access-group 101
!
policy-map type inspect http-filter-pmap
 class type inspect http-no-urlf-cmap
  inspect
 class type inspect http-cmap
  inspect
  urlfilter websense-parmap
!
access-list 101 permit ip 192.168.1.101 any
```

## 라우터에 대한 액세스 제어

대부분의 네트워크 보안 엔지니어는 라우터의 관리 인터페이스(예: SSH, 텔넷, HTTP, HTTPS, SNMP 등)를 공용 인터넷에 노출하면 불편하며, 특정 상황에서는 라우터에 대한 LAN 액세스도 제어해야 합니다. Cisco IOS Software는 NFP(Network Foundation Protection) 기능 제품군, 관리 인터페이스에 대한 다양한 액세스 제어 메커니즘 및 ZFW의 자체 영역을 포함하여 다양한 인터페이스에 대한 액세스를 제한하는 다양한 옵션을 제공합니다. VTY 액세스 제어, 관리 플레인 보호, SNMP 액세스 제어 등의 다른 기능을 검토하여 특정 애플리케이션에 가장 적합한 라우터 제어 기능의 조합을 결정해야 합니다.

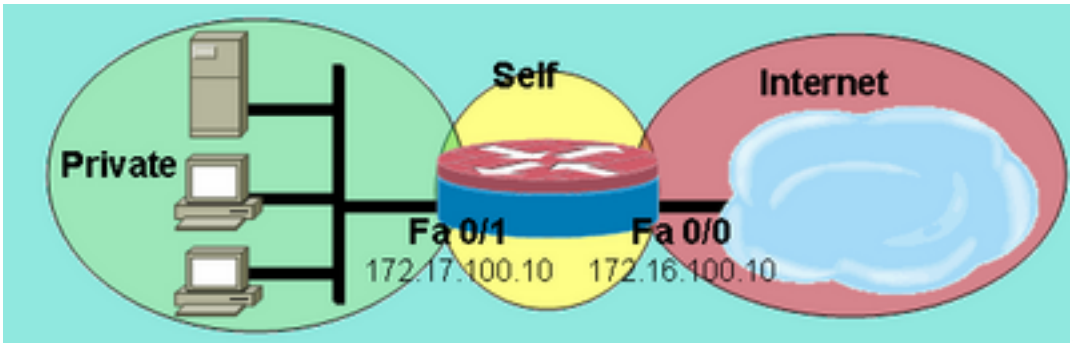
일반적으로 NFP 기능 패밀리는 라우터 자체로 향하는 트래픽을 제어하는 데 가장 적합합니다. NFP 기능을 사용하는 라우터 보호에 대한 자세한 내용은 Cisco IOS Software의 컨트롤 플레인 보안 개요를 참조하십시오.

ZFW를 적용하여 라우터 자체의 IP 주소에서 들어오고 나가는 트래픽을 제어하려는 경우 방화벽 기본 정책 및 기능이 전송 트래픽에 사용할 수 있는 것과 다르다는 점을 이해해야 합니다. 트랜짓 트래픽은 소스 및 목적지 IP 주소가 라우터 인터페이스에 적용된 IP 주소와 일치하지 않는 네트워크 트래픽으로 정의되며, 이러한 트래픽은 라우터가 ICMP TTL 만료 또는 네트워크/호스트 도달 불가 메시지와 같은 네트워크 제어 메시지를 전송하지 않도록 합니다.

ZFW는 영역 간에 이동하는 트래픽에 기본 거부 정책을 적용합니다. 단, 일반 규칙에서 언급한 것처럼 라우터 인터페이스의 주소로 직접 이동하는 모든 영역의 트래픽은 암시적으로 허용됩니다. 이렇게 하면 영역 방화벽 컨피그레이션이 라우터에 적용될 때 라우터의 관리 인터페이스에 대한 연결이 유지됩니다. 동일한 모든 거부 정책이 라우터와의 연결에 직접 영향을 미치는 경우, 라우터에 영역을 구성하기 전에 완전한 관리 정책 컨피그레이션을 적용해야 합니다. 정책이 잘못 구현되거나 잘못된 순서로 적용된 경우 관리 연결이 중단될 수 있습니다.

인터페이스가 영역 멤버로 구성된 경우 인터페이스에 연결된 호스트는 영역에 포함됩니다. 그러나 라우터 인터페이스의 IP 주소를 오가는 트래픽은 영역 정책에 의해 제어되지 않습니다(그림 10의 참고 사항에 설명된 경우는 예외). 그 대신, 라우터의 모든 IP 인터페이스는 ZFW가 구성될 때 자동으로 자체 영역에 포함됩니다. 라우터의 여러 영역에서 라우터 인터페이스로 이동하는 IP 트래픽을 제어하려면 영역과 라우터 자체 영역 간의 트래픽을 차단 또는 허용/검사하도록 정책을 적용해야 하며, 그 반대의 경우도 마찬가지입니다(그림 11 참조).

**그림 11: 네트워크 영역과 라우터 자체 영역 간에 정책 적용**



*네트워크 영역과 라우터 자체 영역 간에 정책 적용*

라우터가 모든 영역과 자체 영역 간에 기본 허용 정책을 제공하지만, 어떤 영역에서든 자체 영역으로 정책이 구성되고 라우터의 사용자 구성 가능한 인터페이스 연결 영역에 대한 자체 정책이 구성되지 않은 경우, 라우터에서 시작된 모든 트래픽은 라우터를 반환할 때 연결된 영역과 자체 영역 간 정책을 발견하고 차단됩니다. 따라서 라우터에서 시작된 트래픽이 자체 영역으로 돌아갈 수 있도록 검사해야 합니다.

> **참고:** Cisco IOS Software는 syslog, tftp, 텔넷 및 기타 컨트롤 플레인 서비스와 같은 트래픽에 대해 인터페이스 "가장 가까운" 대상 호스트와 연결된 IP 주소를 항상 사용하며 이 트래픽을 자체 영역 방화벽 정책으로 전환합니다. 그러나 서비스에서 특정 인터페이스를 logging source-interface [type number] , ip tftp source-interface [type number] 및 ip telnet source-interface [type number] 명령을 포함하는(이에 제한되지 않음) 소스 인터페이스로 정의하는 경우 트래픽은 자체 영역에 적용됩니다.

> **참고**:일부 서비스(특히 라우터의 VoIP(voice-over-IP) 서비스)는 보안 영역에 할당할 수 없는 임시 인터페이스 또는 구성 불가능한 인터페이스를 사용합니다. 이러한 서비스는 트래픽이 구성된 보안 영역과 연결될 수 없는 경우 제대로 작동할 수 없습니다.

### 자체 영역 정책 제한

자체 영역 정책은 통과 트래픽 영역 쌍에 사용할 수 있는 정책에 비해 기능이 제한적입니다.

- 기존의 스테이트풀 검사와 마찬가지로 라우터에서 생성되는 트래픽은 TCP, UDP, ICMP 및 H.323에 대한 복합 프로토콜 검사로 제한됩니다.
- 자체 영역 정책에는 애플리케이션 검사를 사용할 수 없습니다.
- 자체 영역 정책에서는 세션 및 속도 제한을 구성할 수 없습니다.

### 자체 영역 정책 컨피그레이션

대부분의 경우 라우터 관리 서비스에 대한 바람직한 액세스 정책은 다음과 같습니다.

- 모든 텔넷 연결을 거부합니다. 텔넷의 일반 텍스트 프로토콜은 사용자 자격 증명 및 기타 중요한 정보를 쉽게 노출하기 때문입니다.
- 모든 영역의 모든 사용자로부터의 SSH 연결을 허용합니다. SSH는 사용자 자격 증명 및 세션 데이터를 암호화하여, 패킷 캡처 툴을 사용하는 악의적인 사용자로부터 사용자 활동을 스누핑하고 사용자 자격 증명 또는 라우터 컨피그레이션과 같은 중요한 정보를 손상시킵니다. SSH 버전 2는 더 강력한 보호 기능을 제공하며 SSH 버전 1의 고유한 특정 취약성을 해결합니다.
- 전용 영역이 신뢰할 수 있는 경우 전용 영역에서 라우터로 HTTP 연결을 허용합니다. 그렇지 않으면 비공개 영역에 악의적인 사용자가 정보를 손상시킬 가능성이 있는 경우 HTTP는 관리 트

래픽을 보호하기 위해 암호화를 사용하지 않으며 사용자 자격 증명 또는 컨피그레이션과 같은 중요한 정보를 공개할 수 있습니다.
- 모든 영역에서 HTTPS 연결을 허용합니다. SSH와 유사하게 HTTPS는 세션 데이터 및 사용자 자격 증명을 암호화합니다.
- 특정 호스트 또는 서브넷에 대한 SNMP 액세스를 제한합니다. SNMP를 사용하여 라우터 컨피그레이션을 수정하고 컨피그레이션 정보를 표시할 수 있습니다. SNMP는 다양한 커뮤니티에 대한 액세스 제어로 구성해야 합니다.
- 공용 인터넷에서 전용 영역 주소로의 ICMP 요청을 차단합니다(전용 영역 주소를 라우팅할 수 있다고 가정). 필요한 경우 네트워크 트러블슈팅을 위해 하나 이상의 공개 주소를 ICMP 트래픽에 노출할 수 있습니다. 라우터 리소스를 마비시키거나 네트워크 토폴로지 및 아키텍처를 재확인하는 데 여러 ICMP 공격을 사용할 수 있습니다.

라우터는 제어해야 하는 각 영역에 대해 두 개의 영역 쌍을 추가하여 이러한 유형의 정책을 적용할 수 있습니다. 라우터 자체 영역으로 인바운드 또는 아웃바운드 트래픽의 각 영역 쌍은 반대 방향에서 트래픽이 발생하지 않는 한 반대 방향에서 각 정책과 일치해야 합니다. 모든 트래픽을 설명하는 인바운드 및 아웃바운드 영역 쌍에 대해 각각 하나의 정책 맵을 적용하거나 영역 쌍당 특정 정책 맵을 적용할 수 있습니다. 정책 맵당 특정 영역 쌍의 컨피그레이션은 각 정책 맵과 일치하는 활동을 볼 수 있는 세분성을 제공합니다.

172.17.100.11에 SNMP 관리 스테이션이 있고 172.17.100.17에 TFTP 서버가 있는 네트워크 예시, 이 출력은 전체 관리 인터페이스 액세스 정책의 예를 제공합니다.

```
class-map type inspect match-any self_service-cmap
 match protocol tcp
 match protocol udp
 match protocol icmp
 match protocol h323
!
class-map type inspect match-all to-self-cmap
 match class-map self_service-cmap
 match access-group 120
!
class-map type inspect match-all from-self-cmap
 match class-map self_service-cmap
!
class-map type inspect match-all tftp-in-cmap
 match access-group 121
!
class-map type inspect match-all tftp-out-cmap
 match access-group 122
!
policy-map type inspect to-self-pmap
 class type inspect to-self-cmap
  inspect
 class type inspect tftp-in-cmap
  pass
!
policy-map type inspect from-self-pmap
 class type inspect from-self-cmap
  inspect
 class type inspect tftp-out-cmap
  pass
!
zone security private
zone security internet
zone-pair security priv-self source private destination self
 service-policy type inspect to-self-pmap
zone-pair security net-self source internet destination self
```

```
 service-policy type inspect to-self-pmap
zone-pair security self-priv source self destination private
 service-policy type inspect from-self-pmap
zone-pair security self-net source self destination internet
 service-policy type inspect from-self-pmap

!
interface FastEthernet 0/0
 ip address 172.16.100.10
 zone-member security internet
!
interface FastEthernet 0/1
 ip address 172.17.100.10
 zone-member security private
!
access-list 120 permit icmp 172.17.100.0 0.0.0.255 any
access-list 120 permit icmp any host 172.17.100.10 echo
access-list 120 deny icmp any any
access-list 120 permit tcp 172.17.100.0 0.0.0.255 host 172.17.100.10 eq www
access-list 120 permit tcp any any eq 443
access-list 120 permit tcp any any eq 22
access-list 120 permit udp any host 172.17.100.10 eq snmp
access-list 121 permit udp host 172.17.100.17 host 172.17.100.10
access-list 122 permit udp host 172.17.100.10 host 172.17.100.17
```

불행히도 자체 영역 정책은 TFTP 전송을 검사하는 기능을 제공하지 않습니다. 따라서 TFTP가 방화벽을 통과해야 하는 경우 방화벽은 TFTP 서버를 오가는 모든 트래픽을 전달해야 합니다.

라우터가 IPSec VPN 연결을 종료하는 경우 IPSec ESP, IPSec AH, ISAKMP 및 NAT-T IPSec(UDP 4500)을 전달하는 정책도 정의해야 합니다. 이는 사용하는 서비스에 따라 필요한 항목에 따라 달라집니다. 이 다음 정책은 위의 정책 외에도 적용할 수 있습니다. VPN 트래픽에 대한 클래스 맵이 통과 작업과 함께 삽입된 정책 맵의 변경 사항을 확인합니다. 보안 정책에서 지정된 엔드포인트에서 암호화된 트래픽을 허용해야 한다고 명시하지 않는 한, 일반적으로 암호화된 트래픽은 신뢰할 수 있습니다.

```
class-map type inspect match-all crypto-cmap
 match access-group 123
!
policy-map type inspect to-self-pmap
 class type inspect crypto-cmap
  pass
 class type inspect to-self-cmap
  inspect
 class type inspect tftp-in-cmap
  pass
!
policy-map type inspect from-self-pmap
 class type inspect crypto-cmap
  pass
 class type inspect from-self-cmap
  inspect
 class type inspect tftp-out-cmap
  pass
!
access-list 123 permit esp any any
access-list 123 permit udp any any eq 4500
access-list 123 permit ah any any
access-list 123 permit udp any any eq 500
```

# 영역 기반 방화벽 및 광역 애플리케이션 서비스

Cisco WAAS(Wide Area Application Services) 릴리스 정보(소프트웨어 버전 4.0.13) - 소프트웨어 버전 4.0.13의 새로운 기능에서 구성 예 및 사용 지침을 제공하는 애플리케이션 정보를 참조하십시오

# show 및 debug 명령을 사용하여 영역 기반 정책 방화벽 모니터링

ZFW는 정책 컨피그레이션을 보고 방화벽 활동을 모니터링하기 위해 새로운 명령을 도입했습니다.

영역 설명 및 지정된 영역에 포함된 인터페이스를 표시합니다.

```
show zone security [<zone-name>]
```

영역 이름이 포함되지 않은 경우 명령은 구성된 모든 영역의 정보를 표시합니다.

```
Router#show zone security z1
zone z1
  Description: this is test zone1
  Member Interfaces:
    Ethernet0/0
```

영역 쌍에 연결된 소스 영역, 대상 영역 및 정책을 표시합니다.

```
show zone-pair security [source <source-zone-name>] [destination <destination-zone-name>]
```

소스 또는 대상을 지정하지 않으면 소스, 대상 및 연결된 정책이 있는 모든 영역 쌍이 표시됩니다. 소스/대상 영역만 언급하면 이 영역을 소스/대상으로 포함하는 모든 영역 쌍이 표시됩니다.

```
Router#show zone-pair security
zone-pair name zp
  Source-Zone z1  Destination-Zone z2
  service-policy p1
```

지정된 정책 맵을 표시합니다.

```
show policy-map type inspect [<policy-map-name> [class <class-map-name]]
```

정책 맵의 이름을 지정하지 않으면 inspect 유형의 모든 정책 맵이 표시됩니다(하위 유형을 포함하는 레이어 7 정책 맵과 함께).

```
Router#show policy-map type inspect p1
 Policy Map type inspect p1
   Class c1
    Inspect
```

지정된 영역 쌍에 현재 있는 런타임 검사 유형 정책 맵 통계를 표시합니다.

```
show policy-map type inspect zone-pair [zone-pair-name] [sessions]
```

영역 쌍 이름이 언급되지 않으면 모든 영역 쌍의 정책 맵이 표시됩니다.

sessions 옵션은 지정된 영역 쌍에서 정책 맵 애플리케이션에 의해 생성된 검사 세션을 표시합니다.

```
Router#show policy-map type inspect zone-pair zp
 Zone-pair: zp

  Service-policy : p1

    Class-map: c1 (match-all)
      Match: protocol tcp
      Inspect
        Session creations since subsystem startup or last reset 0
        Current session counts (estab/half-open/terminating) [0:0:0]
        Maxever session counts (estab/half-open/terminating) [0:0:0]
        Last session created never
        Last statistic reset never
        Last session creation rate 0
        Last half-open session total 0

    Class-map: c2 (match-all)
      Match: protocol udp
      Pass
        0 packets, 0 bytes

    Class-map: class-default (match-any)
      Match: any
      Drop
        0 packets, 0 bytes
```

urlfilter 키워드는 지정된 정책 맵과 관련된 urlfilter 관련 통계(또는 영역 쌍 이름이 지정되지 않은 경우 모든 대상의 정책 맵)를 표시합니다.

```
show policy-map type inspect zone-pair [zone-pair-name] [urlfilter [cache]]
```

cache 키워드가 urlfilter와 함께 지정되면 IP 주소의 urlfilter 캐시가 표시됩니다.

inspect policy-maps에 대한 show policy-map 명령 요약:

```
show policy-map type inspect inspect { <policy name> [class <class name>] |
                  zone-pair [<zone-pair name>] [sessions | urlfilter cache] }
```

# 영역 기반 정책 방화벽 서비스 거부 보호 조정

ZFW는 네트워크 엔지니어에게 네트워크 활동의 급격한 변화를 알리고, 원치 않는 활동을 완화함으로써 네트워크 활동 변경의 영향을 줄일 수 있도록 DoS 보호 기능을 제공합니다. ZFW는 모든 정책 맵의 클래스 맵에 대해 별도의 카운터를 유지 관리합니다. 따라서 하나의 클래스 맵이 서로 다른 두 영역 쌍의 정책 맵에 사용되는 경우 서로 다른 두 DoS 보호 카운터 집합이 적용됩니다.

ZFW는 12.4(11)T 이전 Cisco IOS Software 릴리스에서 DoS 공격 완화 기능을 기본값으로 제공합니다. Cisco IOS Software Release 12.4(11)T로 기본 DoS 보호 동작이 변경되었습니다.

TCP SYN DoS 공격에 대한 자세한 내용은 TCP SYN DoS 공격에 대한 보호 전략 정의를 참조하십시오.

# 부록

## 부록 A: 기본 설정

```
ip subnet-zero
ip cef
!
bridge irb
!
interface FastEthernet0
 ip address 172.16.1.88 255.255.255.0
 duplex auto
 speed auto
!
interface FastEthernet1
 ip address 172.16.2.1 255.255.255.0
 duplex auto
 speed auto
!
interface FastEthernet2
 switchport access vlan 2
!
interface FastEthernet3
 switchport access vlan 2
!
interface FastEthernet4
 switchport access vlan 1
!
interface FastEthernet5
 switchport access vlan 1
!
interface FastEthernet6
 switchport access vlan 1
!
interface FastEthernet7
 switchport access vlan 1
!
interface Vlan1
 no ip address
 bridge-group 1
!
interface Vlan2
 no ip address
 bridge-group 1
!
interface BVI1
 ip address 192.168.1.254 255.255.255.0
 ip route-cache flow
!
ip classless
ip route 0.0.0.0 0.0.0.0 172.16.1.1
!
bridge 1 protocol ieee
bridge 1 route ip
!
end
```

## 부록 B: 최종(전체) 컨피그레이션

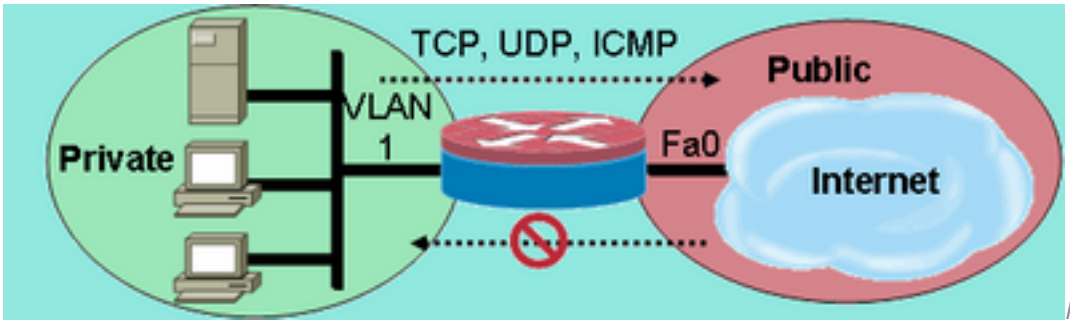
```
ip subnet-zero
ip cef
!
ip port-map user-Xwindows port tcp from 6900 to 6910
!
class-map type inspect match-any L4-inspect-class
 match protocol tcp
 match protocol udp
 match protocol icmp
class-map type inspect match-any L7-inspect-class
  match protocol ssh
  match protocol ftp
  match protocol pop
  match protocol imap
  match protocol esmtp
  match protocol http
class-map type inspect match-any dns-http-class
 match protocol dns
 match protocol http
class-map type inspect match-any smtp-class
 match protocol smtp
class-map type inspect match-all dns-http-acl-class
 match access-group 110
 match class-map dns-http-class
class-map type inspect match-all smtp-acl-class
 match access-group 111
 match class-map smtp-class
class-map type inspect match-any Xwindows-class
 match protocol user-Xwindows
class-map type inspect match-any internet-traffic-class
 match protocol http
 match protocol https
 match protocol dns
 match protocol icmp
class-map type inspect http match-any bad-http-class
 match  port-misuse all
 match  strict-http
!
policy-map type inspect clients-servers-policy
 class type inspect L4-inspect-class
  inspect
policy-map type inspect private-dmz-policy
  class type inspect L7-inspect-class
   inspect
policy-map type inspect internet-dmz-policy
 class type inspect dns-http-acl-class
  inspect
 class type inspect smtp-acl-class
  inspect
policy-map type inspect servers-clients-policy
 class type inspect Xwindows-class
  inspect
policy-map type inspect private-internet-policy
  class type inspect internet-traffic-class
   inspect
  class type inspect bad-http-class
   drop
!
zone security clients
zone security servers
zone security private
zone security internet
zone security dmz
zone-pair security private-internet source private destination internet
```

```
  service-policy type inspect private-internet-policy
zone-pair security servers-clients source servers destination clients
  service-policy type inspect servers-clients-policy
zone-pair security clients-servers source clients destination servers
  service-policy type inspect clients-servers-policy
zone-pair security private-dmz source private destination dmz
  service-policy type inspect private-dmz-policy
zone-pair security internet-dmz source internet destination dmz
  service-policy type inspect internet-dmz-policy
!
bridge irb
!
interface FastEthernet0
 ip address 172.16.1.88 255.255.255.0
 zone-member internet
!
interface FastEthernet1
 ip address 172.16.2.1 255.255.255.0
 zone-member dmz
!
interface FastEthernet2
 switchport access vlan 2
!
interface FastEthernet3
 switchport access vlan 2
!
interface FastEthernet4
 switchport access vlan 1
!
interface FastEthernet5
 switchport access vlan 1
!
interface FastEthernet6
 switchport access vlan 1
!
interface FastEthernet7
 switchport access vlan 1
!
interface Vlan1
 no ip address
 zone-member clients
 bridge-group 1
!
interface Vlan2
 no ip address
 zone-member servers
 bridge-group 1
!
interface BVI1
 ip address 192.168.1.254 255.255.255.0
 zone-member private
!
ip classless
ip route 0.0.0.0 0.0.0.0 172.16.1.1
!
access-list 110 permit ip any host 172.16.2.2
access-list 111 permit ip any host 172.16.2.3
!
bridge 1 protocol ieee
bridge 1 route ip
!
End
```

## 부록 C: 2개 영역에 대한 기본 영역 정책 방화벽 컨피그레이션

이 예에서는 Cisco IOS Software ZFW의 개선 사항을 테스트할 수 있는 간단한 컨피그레이션을 제공합니다. 이 컨피그레이션은 1811 라우터에 구성된 대로 두 개의 영역에 대한 모델 컨피그레이션입니다. 프라이빗 영역은 라우터의 고정 스위치 포트에 적용되므로 스위치 포트의 모든 호스트는 VLAN 1에 연결됩니다. 퍼블릭 영역은 FastEthernet 0에 적용됩니다(그림 12 참조).

**그림 12: FastEthernet 0에 적용되는 공용 영역**




*FastEthernet 0에 적용되는 공용 영역*

```
class-map type inspect match-any private-allowed-class
 match protocol tcp
 match protocol udp
 match protocol icmp
class-map type inspect match-all http-class
 match protocol http
!
policy-map type inspect private-allowed-policy
 class type inspect http-class
  inspect my-parameters
 class type inspect private-allowed-class
  inspect
!
zone security private
zone security public
zone-pair security priv-pub source private destination public
 service-policy type inspect private-allowed-policy
!
interface fastethernet 0
 zone-member security public
!
interface VLAN 1
 zone-member security private
```

# 관련 정보

- 기술 지원 및 문서 – Cisco Systems

이 번역에 관하여

Cisco는 전 세계 사용자에게 다양한 언어로 지원 콘텐츠를 제공하기 위해 기계 번역 기술과 수작업 번역을 병행하여 이 문서를 번역했습니다. 아무리 품질이 높은 기계 번역이라도 전문 번역가의 번역 결과물만큼 정확하지는 않습니다. Cisco Systems, Inc.는 이 같은 번역에 대해 어떠한 책임도 지지 않으며 항상 원본 영문 문서(링크 제공됨)를 참조할 것을 권장합니다.