01 용어 정리

용어 정리

이 가이드는 Check Point 보안 게이트웨이의 거의 모든 기능을 훑기 때문에 용어도 폭넓게 나옵니다. 여기서는 뒤 장들을 읽을 때 바탕이 되는 핵심 용어 만 골라 흐름에 따라 풀어 둡니다.

솔루션의 세 기둥

가장 먼저 세 가지 기본 구성요소입니다. **Security Gateway** 는 조직의 보안 정책을 집행하고 LAN의 진입점이 되는 전용 서버 로, Management Server가 관리합니다. **Security Management Server** 는 객체와 정책을 관리·저장하고 게이트웨이에 배포 하는 서버이며, **SmartConsole** 은 정책을 짜고 제품·이벤트를 모니터링하며 장비를 프로비저닝하는 GUI 애플리케이션 입니다. 여러 게이트웨이를 묶은 **Cluster**, 그리고 Maestro·Chassis 같은 **Scalable Platform Security Group** 도 게이트웨이 자리에 올 수 있습니다.

정책의 뼈대는 **Rule Base** 입니다. 보안 정책을 이루는 규칙 묶음 전체 를 뜻하고, 그 안의 조건·동작 한 줄이 **Rule** 입니다. **Software Blade** 는 특정 보안 기능 모듈 로, 게이트웨이에서는 트래픽의 특정 측면을 검사하고 Management Server에서는 관리 기능을 제공합니다(Software Blade 총람에서 종류별로 다룸).

운영·통신의 기본 용어

장비 사이 통신의 토대는 **SIC(Secure Internal Communication)** 입니다. Check Point 장비들이 SSL로 서로를 인증하는 고유 메커니즘 으로, Management Server의 **ICA(Internal Certificate Authority)** 가 발급한 인증서에 기반합니다.

운영체제는 **Gaia** 입니다. Check Point의 보안 운영체제 로, 웹 인터페이스인 **Gaia Portal**, 기본 명령줄 셸인 **Gaia Clish**(역할 기반 제한 셸), 그리고 전체 root 권한을 주는 **Expert Mode** 를 함께 제공합니다. 소프트웨어 보수에는 **Hotfix**(특정 동작을 고치거나 기능을 더하는 패키지)와 이를 묶은 **Jumbo Hotfix Accumulator(JHA)** 가 쓰입니다.

성능·가상화 기술

성능을 끌어올리는 기술이 여럿 등장합니다. **SecureXL** 은 게이트웨이를 지나는 IPv4·IPv6 트래픽을 가속 하고, **CoreXL** 은 여러 CPU 코어에 방화벽 인스턴스를 복제해 병렬 처리 합니다. CoreXL의 일부인 **CoreXL SND**(Secure Network Distributer)는 들어오는 트래픽을 받아 각 방화벽 인스턴스로 분배하며, 그 복제된 방화벽 하나하나가 **CoreXL Firewall Instance** 입니다(성능·가속·가상화).

가상화 쪽에는 **VSX(Virtual System Extension)** 가 있습니다. 한 하드웨어 위에 여러 가상 방화벽(Virtual System)을 올리는 솔루션으로, 각 Virtual System은 물리 게이트웨이와 같은 기능을 합니다. 이 가상 시스템들을 호스팅하는 물리 서버가 **VSX Gateway** 이며, 그 안의 첫 시스템이 VS0입니다.

배포·관리 형태

게이트웨이를 어디에 두고 어떻게 관리하느냐도 용어로 구분됩니다. **Standalone** 은 게이트웨이와 Management Server를 한 장비에 설치 한 구성, **Distributed Deployment** 는 둘을 다른 장비에 나눠 설치 한 구성입니다. **DAIP Gateway** 는 외부 인터페이스 IP를 ISP가 동적으로 배정 하는 게이트웨이를 말합니다.

여러 도메인을 한 번에 관리할 때는 **Multi-Domain Server(MDS)** 가 가상 관리 서버(Domain Management Server)들을 호스팅 하고, **Multi-Domain Log Server(MDLS)** 가 각 도메인의 로그를 모읍니다. 로그 전용 서버는 **Log Server** 입니다.

이 밖에 정책에서 자주 쓰는 객체로 **Network Object**(토폴로지의 여러 부분 — 컴퓨터·IP·프로토콜 등을 나타내는 논리 객체), **Updatable Object**(Microsoft 365·AWS·Geo 위치처럼 외부 서비스를 나타내는 객체), **Dynamic Object**(IP를 미리 알 수 없어 실시간으로 해석하는 객체)가 있습니다.

02 Quantum Security Gateway 솔루션

Quantum Security Gateway 솔루션

이 가이드 전체를 떠받치는 그림 한 장이 있습니다. Check Point 방화벽 솔루션이 어떤 부품으로 이뤄지고 누가 무엇을 하는지 — 이 한 장을 잡으면 뒤 장들이 모두 이 위에 얹히는 디테일입니다.

!Check Point 방화벽 솔루션 구성 *① SmartConsole ② Security Management Server ③ 인터넷·외부 네트워크 ④ Security Gateway·Cluster·Scalable Platform Security Group ⑤ 내부 네트워크*

세 가지 핵심 구성요소

Check Point 방화벽 솔루션은 세 부품 으로 돌아갑니다.

Security Gateway(또는 Cluster·Scalable Platform Security Group)는 조직의 보안 정책을 실제로 집행하는 엔진 입니다. LAN으로 들어오는 진입점 역할을 하며, Management Server의 관리를 받습니다. 트래픽을 검사하고 정책에 따라 허용·차단하는 일이 모두 여기서 일어납니다.

Security Management Server 는 보안 정책을 관리·저장하고, 그것을 게이트웨이에 배포 하는 서버입니다. 정책은 여기서 만들어져 게이트웨이로 내려갑니다.

SmartConsole 은 이 모든 것을 다루는 GUI 애플리케이션 입니다. 보안 정책을 구성하고, 제품과 이벤트를 모니터링하며, 업데이트를 설치하고, 새 장비를 프로비저닝하며, 멀티 도메인 환경까지 관리합니다.

어디서 더 보나

이 가이드는 게이트웨이 자체에 초점을 맞추므로, 곁가지 주제는 전용 가이드로 넘깁니다. Cluster는 R82 ClusterXL 관리자 가이드, Scalable Platform(Maestro·Chassis)은 해당 Scalable Platforms 가이드, Management Server와 SmartConsole은 R82 Security Management 관리자 가이드 에서 자세히 다룹니다.

다음 장부터는 이 게이트웨이가 어떤 보안 정책을 집행하는지, 어떤 Software Blade들을 얹을 수 있는지, 그리고 성능·가상화 기능과 여러 운영 기능으로 이어집니다.

03 보안 정책(Security Policy)

보안 정책(Security Policy)

Security Policy 는 네트워크 트래픽을 통제하고 데이터 보호·접근 규칙을 패킷 검사로 강제하는 규칙·설정의 묶음 입니다. Check Point은 한 종류가 아니라 목적이 다른 여러 정책 종류 를 제공합니다. 이 장은 그 종류들이 각각 무엇을 하고 SmartConsole 어디에서 다루는지를 한눈에 정리합니다.

Access Control Policy — 접근 통제의 본체

가장 핵심인 **Access Control Policy** 는 지정한 출발지에서 목적지로, 지정한 프로토콜을 통한 접근을 통제 합니다. 세 개의 Rule Base로 이뤄집니다.

Access Control Rule Base 는 단순·세밀한 규칙을 통합(unified)해 접근을 제어 합니다. 게이트웨이에 Identity Awareness를 켜면 Access Role 객체를 출발지·목적지로 써서 개인·그룹별 규칙 을 쉽게 만들 수 있습니다. 규칙은 이름·출발지·목적지·VPN·서비스/애플리케이션·동작(Accept/Drop/Reject 등)·시간·추적(Track)·설치 대상으로 구성됩니다. SmartConsole의 **Security Policies** > **Access Control** > **Policy** 에서 다룹니다.

NAT Rule Base 는 네트워크 주소 변환(NAT)을 위한 자동·수동 규칙 을 담으며, **Security Policies** > **Access Control** > **NAT** 에서 다룹니다. **Desktop Rule Base** 는 Remote Access Client에 적용되는 Desktop Security 정책 으로, 클라이언트가 VPN Site 업데이트 때 내려받아 자신의 접근을 통제합니다(Inbound·Outbound 두 Rule Base). 쓰려면 게이트웨이에서 IPsec VPN·Policy Server 블레이드를 켜고 정책 패키지에서 Desktop Security를 활성화해야 합니다.

그 밖의 정책 종류들

Threat Prevention Policy 는 봇·바이러스를 어떻게 검사할지 정하며, Malware 데이터베이스와 네트워크 객체를 쓰는 Rule Base가 핵심입니다(**Security Policies** > **Threat Prevention** > **Policy**). 프로파일은 Basic·Optimized·Strict 또는 사용자 정의 중에서 고릅니다.

HTTPS Inspection Policy 는 암호화된 HTTP/HTTPS 트래픽을 검사 하게 해 줍니다. 게이트웨이는 암호화된 HTTPS를 그대로는 못 보므로, HTTPS Inspection을 켜면 게이트웨이가 외부와 새 SSL 연결을 맺어 복호화·검사 합니다. 이 검사 결과는 Anti-Bot·Anti-Virus·Application Control·Content Awareness·DLP·IPS·Threat Emulation·URL Filtering 같은 블레이드가 활용합니다(**Security Policies** > **HTTPS Inspection** > **Policy**).

Data Loss Prevention Policy 는 보호 대상 데이터가 조직 밖으로 나가기 전에 잡아 의도치 않은 유출을 막 습니다(**Manage & Settings** > **Blades** > **Data Loss Prevention** > **Configure in SmartDashboard**). **Geo Policy** 는 특정 지리·정치적 위치를 오가는 트래픽에 대한 정책 입니다.

> 중 요
>
> R81부터 게이트웨이는 SmartConsole의 Shared Policies > Geo Policy 방식의 Geo Policy를 더 이상 지원하지 않 습니다(Known Limitation PMTR-56212). 대신 Access Control 정책에서 Updatable Object를 출발지·목적지로 써서 구현합니다(sk126172).

Mobile Access Policy 는 Mobile Access 게이트웨이로 접속할 때 어떤 사용자 그룹이 어떤 애플리케이션에 접근할지 통제합니다(**Manage & Settings** > **Blades** > **Mobile Access** > **Configure in SmartDashboard**).

각 정책의 상세 구성은 해당 전용 가이드(Security Management·Threat Prevention·DLP·Mobile Access 관리자 가이드)에서 다루며, 이 장은 "어떤 정책이 무엇을 위한 것인지" 를 잡는 지도 역할입니다. 이어지는 Software Blade 총람은 이 정책들을 실제로 집행하는 기능 모듈들을 소개합니다.

04 Software Blade 총람

Software Blade 총람

Software Blade 는 게이트웨이에 얹는 보안 기능 모듈 입니다. 각 블레이드는 트래픽의 특정 측면을 검사하며, 필요한 것만 켜서 조합합니다. 원문은 블레이드마다 한 절씩 짧게 소개하는데, 여기서는 성격별로 묶어 한눈에 정리합니다. 블레이드별 상세 설정은 각 전용 관리자 가이드에 있습니다.

토대 — Firewall

Firewall Software Blade 가 모든 것의 뿌리 입니다. 게이트웨이에서 Access Control 정책과 NAT 정책을 집행 하는 주 블레이드로, 다른 블레이드들은 이 위에 얹힙니다 (보안 정책).

VPN과 원격 접속

IPsec VPN Software Blade 는 게이트웨이와 다른 게이트웨이·클라이언트 사이의 트래픽을 암호화·복호화 합니다. 함께 쓰이는 **Policy Server Software Blade** 는 Remote Access Client에 Desktop Security 정책을 집행 해, 클라이언트의 방화벽이 트래픽을 어떻게 검사할지 통제합니다.

Remote Access VPN 은 원격 사용자와 내부망 사이에 VPN 터널을 만들 어, 직원이 어디서든 안전하게 민감 정보에 접근하게 합니다. 이를 더 많은 클라이언트·배포 형태로 넓힌 것이 **Mobile Access Software Blade** 입니다. Layer 3 VPN과 SSL VPN을 모두 제공 하며, 관리되지 않는 스마트폰·태블릿에서도 Mobile Access Portal과 Check Point Mobile Apps로 회사 자원에 암호화 접속하게 해 줍니다.

Access Control 계열 블레이드

Identity Awareness 는 사용자·컴퓨터의 신원을 IP에 매핑 해, IP만 보던 전통 방화벽의 한계를 메웁니다. 덕분에 Access Role을 출발지·목적지로 써서 "Finance 그룹만 재무 보고서 접근" 같은 신원 기반 규칙 을 만들 수 있고, Active Directory 환경이든 아니든 적용됩니다.

Application Control 은 업계 최대 애플리케이션 라이브러리(AppWiki)로 4,500개 이상 앱과 10만 개 이상 Web 2.0 위젯을 식별·허용·차단 합니다. **URL Filtering** 은 웹사이트·애플리케이션을 분류(category)에 따라 접근 통제 하고, **Content Awareness** 는 데이터를 보고 통제 합니다 — 다운로드/업로드/양방향 방향과 Content Type(신용카드 번호·IBAN 등)·File Type(PDF·실행파일 등)을 Access Control 정책에서 지정합니다.

Data Loss Prevention(DLP) 은 기밀 데이터가 조직을 떠나기 전에 깊은 콘텐츠 검사로 탐지·차단 합니다. **Anti-Spam & Email Security** 는 콘텐츠 지문·IP 평판·사용자 정의 발신자 를 근거로 스팸을 막습니다.

> 참 고
>
> Content Awareness와 DLP는 둘 다 Access Control 정책의 Data Type을 쓰지만 기능이 다르고 독립적으로 동작 하며, 게이트웨이가 각각 따로 집행합니다.

Threat Prevention 계열 블레이드

Threat Prevention 은 감염 전·후를 아우르는 다층 방어로, 여러 블레이드가 함께 작동합니다.

Anti-Bot 은 감염 후 봇을 찾아내고 C&C(명령·제어) 통신을 차단 하며, **Anti-Virus** 는 여러 탐지 엔진으로 게이트웨이에서 멀웨어를 탐지·차단 합니다. 둘 다 ThreatCloud로 끊임없이 갱신됩니다. SandBlast 제품군의 **Threat Extraction** 은 파일에서 악용 가능한 콘텐츠(액티브 콘텐츠·임베디드 객체)를 제거하고 안전한 사본을 만들어 즉시 전달 하고, **Threat Emulation** 은 가상 샌드박스에서 파일을 실행해 악성 행위를 탐지 합니다(SMTP 트래픽 검사에는 **MTA(Mail Transfer Agent)** 기능 필요).

IPS 는 수천 개 시그니처와 행위·선제 보호로 침입을 방지 하는 또 한 겹의 방어층입니다. **Zero Phishing**(R81.20 도입)은 머신러닝으로 알려진·알려지지 않은(zero-day) 피싱 사이트를 실시간 차단 합니다 — 게이트웨이에서 웹 트래픽을 스캔해 Check Point 클라우드로 보내 분석하므로, 브라우저·플랫폼에 구애받지 않고 클라이언트 설치도 필요 없 습니다. URL 평판 기반 엔진과, HTML 폼을 검사하는 in-browser 자바스크립트 주입 엔진 두 가지를 씁니다(켤 때 FQDN 설정 필요, VSX·ClusterXL HA/Load Sharing 지원, Internet Explorer·HTTP 2.0·미러 트래픽은 미지원).

UserCheck — 사용자에게 직접 알리기

UserCheck 는 규정에 어긋나거나 위험한 브라우징에 대해 게이트웨이가 사용자에게 직접 메시지를 보내 는 기능입니다. 게이트웨이의 UserCheck Web Portal로 리디렉션 하거나 단말에 UserCheck Client를 설치 해 메시지를 띄웁니다. DLP와 Access Control(Application Control·URL Filtering·Content Awareness), Threat Prevention(Anti-Bot·Anti-Virus·Threat Emulation·Threat Extraction·Zero Phishing) 블레이드가 이 기능을 지원해, 사용자가 스스로 보안 사고를 막고 정책을 익히 게 돕습니다.

05 성능·가속·가상화 기능

성능·가속·가상화 기능

방화벽이 얼마나 빠르게, 얼마나 크게 확장되는지 를 좌우하는 기술들과, 한 장비를 여러 방화벽으로 쪼개는 가상화를 한데 모은 장입니다. 대부분 자세한 튜닝은 R82 Performance Tuning 관리자 가이드나 전용 가이드로 넘기고, 여기서는 각 기능이 무엇을 하는지 를 잡습니다.

이중화와 부하 분산 — ClusterXL

ClusterXL 은 동일한 게이트웨이 여러 대를 묶는 소프트웨어 클러스터 솔루션입니다. High Availability 클러스터는 장애 시 백업 게이트웨이로 투명하게 페일오버 해 연결·VPN을 지키고, Load Sharing 클러스터는 모든 멤버가 활성화되어 신뢰성과 성능을 함께 높입니다.

!ClusterXL 구성 *① 내부 네트워크 ② 내부망 스위치 ③ ClusterXL을 켠 Security Gateway들 ④ 외부망 스위치 ⑤ 인터넷*

> 참 고
>
> ClusterXL 블레이드는 Scalable Platforms에는 적용되지 않 습니다(Maestro·Chassis는 자체 확장 메커니즘 사용).

대역폭 관리 — QoS

QoS 는 정책 기반 대역폭 관리 솔루션입니다. ERP·DB·웹 같은 업무 트래픽을 우선시하고, VoIP·화상회의에 대역폭을 보장하며 지연을 통제합니다. 계층적 WFQ(Weighted Fair Queuing) 알고리즘 으로 대역폭을 정밀 배분하며, 암호화·비암호화 트래픽 모두에 동작합니다.

!QoS 구성 *① SmartConsole ② Security Management Server ③ QoS 정책 ④ QoS 블레이드를 켠 Security Gateway ⑤ 인터넷 ⑥ 내부 네트워크*

QoS는 워낙 내용이 많아 별도의 Check Point R82 QoS 관리자 가이드 로 자세히 다룹니다.

가속 — SecureXL·CoreXL·Multi-Queue·HyperFlow

SecureXL 은 게이트웨이를 지나는 트래픽을 가속 하는 가장 기본적인 가속 기술입니다. **CoreXL** 은 멀티코어 성능을 끌어내는 기술 로, 방화벽 커널을 여러 벌 복제해 각 CPU 코어에서 하나씩 돌립 니다. 복제된 인스턴스(CoreXL Firewall Instance)는 각각 완전하고 독립된 검사 커널 로 같은 인터페이스·같은 정책으로 동시에 트래픽을 처리하며, 코어 수에 거의 선형으로 성능이 확장 됩니다(관리·토폴로지 변경 불필요). CoreXL 인스턴스는 SecureXL 인스턴스와 함께 동작합니다.

Multi-Queue 는 한 걸음 더 나갑니다. 기본적으로 네트워크 인터페이스 하나는 큐 하나를 한 CPU가 처리 하므로, 가속에 쓸 수 있는 코어 수가 트래픽 처리 인터페이스 수를 넘을 수 없습니다. Multi-Queue는 한 인터페이스에 여러 큐를 둬 여러 코어를 가속에 동원 합니다 (SecureXL이 켜져 있어야 작동 — 기본값).

HyperFlow(R81.20+)는 elephant flow(대용량 연속 연결, 예: 큰 ISO 다운로드) 를 다룹니다. HyperFlow가 없으면 게이트웨이는 한 elephant 연결을 CPU 코어 하나로만 검사해, CPU 사용률이 오르면 처리량이 떨어집니다. HyperFlow는 검사 작업을 잘게 쪼개 여러 코어에 분산 해, Threat Prevention 블레이드가 켜진 상태에서도 대용량 연결의 처리량을 높이고 응답 시간을 개선합니다. User Space Firewall(USFW)에서만, 필요할 때만(CPU 여유가 있을 때만) 자동으로 작동 하며, 전체 처리량이 elephant 연결보다 우선합니다(수동 코어 할당은 불가, 임계값으로 활성/비활성만 제어). 기본은 대기(standby) 모드로, 무거운 연결이 감지되면 활성화됩니다.

콘텐츠 적응·프록시 — ICAP·HTTP(S) Proxy

ICAP(Internet Content Adaptation Protocol) 는 투명 프록시를 확장하는 가벼운 HTTP 유사 프로토콜 로, 보통 바이러스 검사·콘텐츠 필터에 쓰입니다. 게이트웨이는 ICAP Client(메시지를 ICAP 서버로 보내 콘텐츠 적응 요청)·ICAP Server(받은 메시지를 적응 처리)·또는 둘 다 로 설정할 수 있어, 토폴로지 변경 없이 서드파티 ICAP 장비와도 연동됩니다(단, Scalable Platforms는 ICAP Server 미지원).

HTTP/HTTPS Proxy 는 게이트웨이를 호스트 사이의 중개자로 만듭니다. 직접 연결을 막고 클라이언트↔프록시, 프록시↔목적지 두 연결로 분리 합니다. Transparent 모드는 클라이언트 설정 없이 지정 포트·인터페이스의 트래픽을 가로채 고, Non Transparent 모드는 클라이언트에 프록시 서버·포트를 설정 해야 합니다.

가상화 — VSX

VSX(Virtual System eXtension) 는 한 하드웨어에서 여러 가상 방화벽을 돌리는 제품입니다. 각 Virtual System은 자기 네트워크를 보호하는 게이트웨이로 동작하고, VSX Gateway는 목적지 네트워크를 보호하는 Virtual System으로 트래픽을 보냅니다.

!물리 네트워크 vs VSX 가상 네트워크 *① 인터넷 ② 라우터 ③ VSX Gateway(각 Virtual System이 물리 게이트웨이와 같은 보안·네트워킹 기능을 수행) ④ Warp Link(Virtual System과 Virtual Switch를 잇는 가상 인터페이스) ⑤ Virtual Switch(모든 Virtual System을 인터넷 라우터에 연결) ⑥ 네트워크들*

물리 환경에서는 네트워크마다 물리 게이트웨이를 따로 두지만, VSX는 하나의 VSX Gateway·클러스터가 여러 독립 네트워크를 가상으로 정의·보호 합니다. VSX 역시 별도의 Check Point R82 VSX 관리자 가이드 로 자세히 다룹니다.

06 HSM(하드웨어 보안 모듈)

HSM(하드웨어 보안 모듈)

HSM(Hardware Security Module) 은 암호화 키를 저장하는 전용 장치 입니다. 키를 소프트웨어가 아닌 전용 하드웨어에 담아 한 겹 더 보안 을 더하죠. 이 장은 Check Point 게이트웨이가 HSM과 함께 동작하는 큰 그림과 구성 흐름을 정리합니다. 벤더별(Gemalto·FutureX) 세부 절차는 분량이 매우 커서 원문 해당 절을 참고하세요.

왜, 무엇에 쓰나

게이트웨이가 HSM을 쓸 때, HSM은 오직 Outbound HTTPS Inspection을 위한 객체를 보관합니다. 구체적으로 관리자가 미리 만들어 둔 CA(인증 기관) 인증서와 키 쌍, 그리고 가짜 인증서용 RSA 키 쌍 2~3개(HTTPS Inspection 데몬 초기화 때 1024/2048/4096비트로 생성)입니다. 즉 HTTPS 검사에 쓰는 핵심 키를 HSM이 대신 안전하게 보관·제공 하는 것이 요지입니다.

Check Point 게이트웨이와 함께 쓸 수 있는 HSM은 Gemalto Luna SP SafeNet HSM과 FutureX 두 가지이며, PKCS#11 API를 쓰는 다른 벤더는 Check Point Solution Center에 문의합니다.

환경 구성

!HSM과 함께 동작하는 Check Point 환경 *① HTTPS 사이트에 접속하는 내부 컴퓨터 ② HTTPS Inspection을 켠 Check Point Security Gateway ③ 인터넷의 HTTPS 웹사이트 ④ 게이트웨이를 관리하는 Security Management Server ⑤ 상호 연결 네트워크 ⑥ SSL 키·인증서를 저장·제공하는 HSM Server ⑦ HSM Server에 CA 인증서를 만드는 HSM Client 워크스테이션*

핵심은 게이트웨이가 HSM Server를 Outbound HTTPS Inspection에만 쓴다 는 점입니다.

구성의 큰 흐름

게이트웨이를 HSM과 연동하는 절차는 3단계 로 흘러갑니다.

먼저 **HSM 없이** HTTPS Inspection을 먼저 동작시킵니다. SmartConsole에서 HTTPS Inspection을 구성한 뒤, 게이트웨이의 `$FWDIR/conf/hsm_configuration.C` 파일에서 `:enabled ("no")` 로 HSM을 꺼 둔 채 정상 동작을 확인하는 것입니다. 그다음 HSM Client 워크스테이션을 구성해 CA 인증서·키 쌍을 HSM Server에 만들 고, 마지막으로 게이트웨이에서 HSM을 켜(`enabled` "yes") 실제로 HSM의 키를 쓰도록 전환합니다.

이때 환경별 주의가 있습니다. Cluster에서는 모든 멤버를 똑같이 설정 하고, VSX에서는 각 Virtual System 컨텍스트에서 수행하며, Scalable Platforms에서는 해당 Security Group에 접속 해 진행합니다.

정리하면 HTTPS Inspection을 먼저 HSM 없이 세우고(1) → HSM Client로 키를 HSM에 만들고(2) → 게이트웨이에서 HSM을 켜는(3) 흐름이며, 이 한 줄기만 잡으면 Gemalto·FutureX 각각의 세부 명령은 그 위에 얹히는 디테일입니다. 통신을 끊거나 (Disabling Communication) HSM 연동 시 HTTPS 검사 모니터링하는 방법도 원문 해당 절에 이어집니다.

07 ISP 이중화

ISP 이중화

ISP Redundancy 는 게이트웨이를 두 개 이상의 ISP 회선으로 인터넷에 연결 해, 한 회선이 끊겨도 인터넷 연결이 유지되게 합니다. 게이트웨이가 ISP 링크들을 감시해 그때그때 가장 좋은 링크를 고릅 니다.

> 중 요
>
> ISP Redundancy는 Dynamic Routing이 구성되어 있으면 지원되지 않 습니다(Known Limitation PMTR-68991). Cluster에서의 ISP Redundancy는 R82 ClusterXL 관리자 가이드를 참고하세요.

전제와 범위는 이렇습니다. 외부 인터페이스가 최소 2개 필요하고 최대 10개까지 지원하며(3개 이상은 Management Server·게이트웨이가 R81.10 이상이어야 함), 기본적으로 내부망에서 인터넷으로 나가는 트래픽 을 위한 기능입니다.

배선 — 권장은 인터페이스 분리

!두 ISP 링크에 전용 물리 인터페이스 두 개 *① 내부 네트워크 ② Security Gateway / Security Group ③ ISP A ④ ISP B ⑤ 인터넷*

두 ISP 링크에 각각 전용 물리 인터페이스를 주는 구성이 권장 됩니다(더 단순하기 때문). 외부 인터페이스가 하나뿐이면, 같은 인터페이스에 두 서브넷을 두고 서로 다른 next-hop 라우터(보통 스위치 경유)에 연결 하는 방식으로도 가능합니다(Gaia의 인터페이스 Alias 사용).

두 가지 모드

나가는 연결의 동작은 두 모드 중 하나로 정합니다.

Load Sharing 은 모든 링크에 부하를 분산 합니다. 새 연결을 링크에 무작위 배정하고, 링크마다 상대 부하(weight)를 줘 빠른 링크에 더 많이 흘릴 수 있으며, 한 링크가 죽으면 다른 링크가 부하를 떠맡습니다. 들어오는 연결도 게이트웨이가 DNS 응답에 두 ISP의 IP를 번갈아 줘 어느 링크로든 도달할 수 있습니다.

Primary/Backup 은 한 링크(Primary)만 쓰다가 끊기면 Backup으로 전환 합니다. Primary가 복구되면 새 연결은 다시 Primary로 가고, 기존 연결은 끝날 때까지 Backup에 남습니다. 반환 패킷이 연결을 시작한 같은 ISP 링크로 나가 므로 들어오는 연결도 이득을 봅니다.

선택 기준은 단순합니다 — 링크들이 비슷하면 Load Sharing 으로 모두 활용하고, 한 링크가 더 저렴·안정적이면 그걸 Primary로 둔 Primary/Backup 을 씁니다.

나가는·들어오는 연결의 처리

나가는 연결 은 Load Sharing이면 weight 비율대로 링크에 분산 되고, Primary/Backup이면 활성 Primary 링크를 쓰며 Hide NAT으로 출발지 주소를 나가는 인터페이스 주소로 바꿔 반환 패킷이 같은 링크로 돌아오게 합니다.

들어오는 연결 을 받으려면 각 애플리케이션 서버에 ISP마다 라우팅 가능한 공인 IP를 하나씩 주고, Static NAT으로 실제 서버 주소로 변환 해야 합니다. 핵심은 게이트웨이의 내장 미니 DNS 서버 입니다. 외부 클라이언트가 `www.example.com` 을 물으면(Type A DNS 쿼리), 게이트웨이가 이를 가로채 Primary/Backup이면 활성 링크의 IP만, Load Sharing이면 두 IP를 번갈아 응답합니다. 한 링크(예: ISP A)가 죽으면 그 IP는 빠지고 클라이언트는 살아 있는 링크의 IP로 해석합니다. 호스트 이름을 모르면 원래 목적지나 도메인 DNS 서버로 쿼리를 넘깁니다.

구성 절차의 큰 줄기

SmartConsole에서 게이트웨이 객체를 열어 **Other** > **ISP Redundancy** 로 갑니다. 흐름은 ISP Redundancy 지원 켜기 → 모드 선택 → ISP 링크 구성(2~10개) → 게이트웨이를 DNS 서버로 설정 → Access Control 정책 구성 → 정책 설치 입니다.

ISP 링크는 외부 토폴로지 인터페이스가 2개 이상이면 "Set initial configuration"으로 자동 잡고(Primary/Backup이면 Primary를 목록 맨 위로), 하나뿐이면 수동 으로 추가합니다(이름·인터페이스·next-hop·Load Sharing이면 weight를 합 100%가 되게 — 2개면 각 50, 3개면 각 33 …). DNS 서버 설정은 **Enable DNS Proxy** 를 켜고 DMZ/웹 서버를 ISP별 공인 IP로 등록한 뒤 Static NAT을 구성합니다. Access Control 정책에는 나가는 연결을 시작하는 내부 객체에 Automatic Hide NAT 을 걸고, 공개 서버에는 Static NAT 규칙을 둡니다(수동 NAT을 쓰면 자동 ARP가 안 되므로 `local.arp` 를 sk30197대로 설정).

VPN, 그리고 CLI 제어

ISP Redundancy 설정은 VPN Link Selection 설정을 덮어씁 니다(우선함). 덕분에 VPN 암호화 연결도 ISP 링크 장애를 넘겨 살아남 습니다. Check Point 피어와 쓸 때는 **Apply settings to VPN traffic** 을 켜고 Link Selection이 "Use ongoing probing"으로 ISP Redundancy 모드를 따르게 합니다. 서드파티 피어와는 Load Sharing·Service Based·Route based probing이 Check Point 장비끼리만 동작 하므로, Probing을 써서 한 링크(가장 긴 prefix·가장 낮은 metric)로 연결하게 조정합니다.

CLI로도 제어합니다. `fw isp_link <링크이름> {up|down}` 으로 링크 상태를 강제 할 수 있어(Scalable Platform은 `g_fw isp_link` , Management Server는 게이트웨이 이름까지 지정) 설치 테스트나 잘못 인식된 상태 교정에 씁니다. 게이트웨이가 켜지거나 링크 상태가 바뀌면 `$FWDIR/bin/cpisp_update` 스크립트가 기본 경로를 바꾸는데, 이 스크립트는 손대지 않는 것이 권장 됩니다.

08 Mirror and Decrypt

Mirror and Decrypt

Mirror and Decrypt 는 게이트웨이를 지나는 트래픽을 복제(미러)해, 필요하면 복호화까지 해서 별도 장비로 흘려보내는 기능입니다. 트래픽 기록·분석을 위한 Recorder나 Packet-Broker에 데이터를 넘길 때 씁니다.

두 가지 동작

게이트웨이는 두 가지 일을 할 수 있습니다. **Only mirror of all traffic** 은 지나는 모든 트래픽(복호화 없는 HTTPS 포함)을 그대로 복제 해 지정 물리 인터페이스로 내보냅니다. **Mirror and Decrypt of HTTPS traffic** 은 HTTPS 트래픽을 복제하고 복호화해 평문(clear-text)으로 내보냅니다(이때는 게이트웨이에 HTTPS Inspection을 켜고 구성해야 함).

환경에 서드파티 Recorder나 Packet-Broker를 두고 트래픽을 받게 할 수 있는데, 이 장비는 monitor(promiscuous) 모드로 동작 해야 복호화·미러된 트래픽을 받습니다. 게이트웨이는 하나의 Recorder만, 지정 물리 NIC에 직접 연결 해 씁니다.

!Mirror and Decrypt 트래픽 흐름 *① 첫 번째 네트워크 ② Security Gateway ③ 두 번째 네트워크 ④ 게이트웨이의 지정 물리 인터페이스 ⑤ monitor 모드로 동작하는 Recorder/Packet-Broker · Ⓐ 첫 네트워크↔게이트웨이 흐름 Ⓑ 둘째 네트워크↔게이트웨이 흐름 Ⓒ 게이트웨이→Recorder로 복호화·미러된 트래픽 흐름*

내보내는 패킷의 출발지 MAC 주소는 동작에 따라 다릅니다 — Mirror only는 지정 인터페이스의 MAC, Mirror and Decrypt는 `00:00:00:00:00:00` 입니다.

요구 사항

핵심은 지정 인터페이스를 제대로 고르고 준비 하는 것입니다. 다른 인터페이스들의 트래픽이 모두 합쳐져 지나가므로 가장 큰 처리량(예: 10G·40G)을 가진 물리 인터페이스 를 고르고 (클러스터 멤버는 모두 같은 이름의 인터페이스), 환경의 다른 IP·서브넷과 겹치지 않는 더미 IP를 정확한 서브넷 마스크와 함께 부여합니다. SmartConsole에서 이 인터페이스에 미러링을 켜면 그쪽으로 라우팅되는 다른 트래픽은 모두 드롭 됩니다.

클러스터에서는 이 지정 인터페이스를 `$FWDIR/conf/discntd.if` 파일에 등록 해, 쓰지 않는 인터페이스가 CCP(Cluster Control Protocol) 패킷을 보내 Recorder를 압도하지 않게 합니다. 또 이 인터페이스의 MTU는 1500(기본)이거나, 적어도 다른 인터페이스 중 가장 큰 MTU 이상 이어야 합니다.

Gateway 모드와 VSX 모드 각각의 구성 절차, 그리고 Mirror and Decrypt 로그의 자세한 내용은 원문 해당 절을 참고하세요. 요지는 지나는 트래픽을 복제(필요시 복호화)해 전용 인터페이스로 모아 분석 장비에 넘긴다 는 한 줄기입니다.

09 ConnectControl — 서버 부하 분산

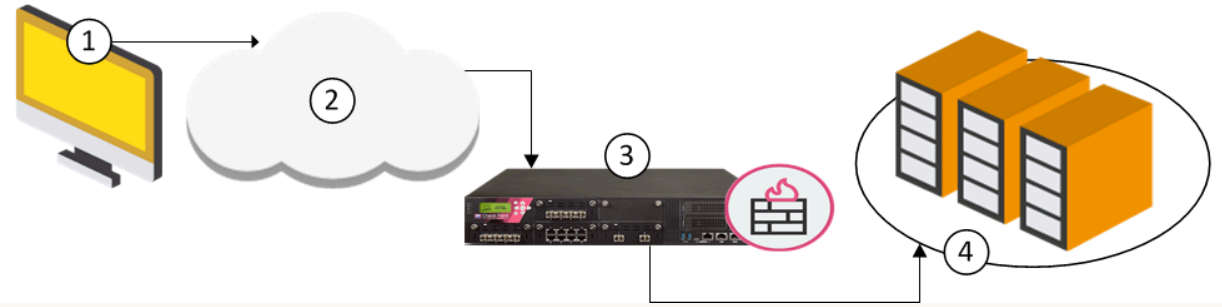
ConnectControl — 서버 부하 분산

ConnectControl 은 게이트웨이 뒤에 있는 여러 서버로 가는 트래픽을 분산 하는 Check Point 솔루션입니다. 게이트웨이가 이미 트래픽을 보고 있으니 추가 메모리·CPU 부담 없이 부하 분산을 얹는 셈입니다.

> 중 요
>
> Scalable Platforms(ElasticXL·Maestro·Chassis)는 이 기능을 지원하지 않습니다(Known Limitation MBS-14173).

동작 원리 — Logical Server

핵심 개념은 **Logical Server** 입니다. 여러 물리 서버를 하나의 가상 IP(Virtual IP)로 대표하는 객체로, 클라이언트는 이 하나의 IP로 접속하고 ConnectControl이 뒤의 적절한 물리 서버로 요청을 넘깁 니다.



ConnectControl 패킷 흐름

흐름은 이렇습니다. 클라이언트가 Logical Server의 IP로 연결을 시작 → 게이트웨이에 도착해 Logical Server 규칙에 매칭 → 게이트웨이가 Logical Server 그룹의 내부 IP로 요청을 전달 → ConnectControl이 부하 분산 방식에 따라 그룹 중 가장 적합한 서버를 선택 합니다.

> 참 고
>
> Logical Server와 서버 그룹으로 가는 트래픽을 허용하는 규칙은 같은 서비스를 허용하는 일반 Access Control 규칙보다 위에 두어야 합니다.

구성 절차의 큰 줄기

Object Explorer(Ctrl+E)에서 시작합니다. 부하 분산할 서버마다 Host 객체를 만들고, 이들을 묶는 Network Group 객체(권장 29개 이하)를 만든 뒤, Logical Server 객체를 정의하는 것이 뼈대입니다. Logical Server에는 공인 Virtual IP를 주고, 서버 타입·서버 그룹·persistency·부하 분산 방식 을 정합니다. 그다음 Access Control 정책에 Logical Server로 가는 부하 분산 규칙 을 추가하고, 전역 설정(ConnectControl)을 잡은 뒤 정책을 설치합니다.

서버 타입 — HTTP vs Other

서버 타입에 따라 ConnectControl이 클라이언트 연결을 다루는 방식이 다릅니다. **HTTP** 타입은 HTTP 리디렉션을 써 모든 HTTP 세션을 한 서버로 보냅니다(많은 웹 애플리케이션이 요구하는 동작, HTTP 프로토콜 전용, 오프사이트 서버 지원). **Other** 타입은 NAT으로 트래픽을 그룹 서버에 보내 며, 모든 프로토콜을 지원하고 가장 효과적으로 부하를 분산합니다(서버가 게이트웨이로 NAT되어야 함).

Persistency와 부하 분산 방식

Persistency(연결 고정) 는 클라이언트를 처음 고른 서버에 계속 묶 습니다. Persistency by server 는 여러 웹 서버 환경의 HTTP 폼 처럼 한 서버로 모든 요청을 보내야 할 때(폼 작성 중 데이터 손실 방지), Persistency by service 는 한 그룹에서 여러 서비스를 분산 할 때(서버마다 HTTP·FTP를 돌리는 환경에서 서비스별로 올바른 서버로) 유용합니다.

부하 분산 방식 은 세 가지입니다. **Random** 은 무작위 배정(RAM·CPU가 비슷하고 같은 세그먼트일 때 좋음), **Server load** 는 새 연결을 가장 잘 감당할 서버를 선택, **Round Robin** 은 순서대로 다음 서버 에 보냅니다(Round Trip·Domain 방식은 미지원).

전역 설정에서는 서버 가용성 점검 주기·재시도 횟수 와 Persistency by server의 타임아웃을 조정합니다. HTTP 리디렉션을 쓰는 경우, 서버 그룹이 클라이언트와 직접 통신하도록 허용하는 규칙 도 추가해야 합니다.

10 그 밖의 운영 기능

그 밖의 운영 기능

게이트웨이를 운영하다 보면 마주치는 작지만 중요한 기능들 — 실시간 모니터링, 클라우드 보안, 고급 라우팅, SNMP — 을 모은 장입니다. 각각은 대부분 전용 가이드에서 깊이 다루므로, 여기서는 무엇을 위한 기능인지 만 잡습니다.

Monitoring Software Blade — 실시간 카운터

Monitoring Software Blade 는 게이트웨이의 상태를 실시간으로 들여다보 게 해 줍니다. 시스템 카운터(CPU 사용률·가상 메모리·여유 디스크 등), 트래픽 연결 수, 트래픽 처리량 을 볼 수 있습니다.

SmartConsole에서는 **Gateways & Servers** 에서 게이트웨이를 고른 뒤 아래 **Summary** 탭의 **Device & License Information** → **System Counters / Traffic** 으로 봅니다 (클러스터는 멤버를 골라서). 사용자·VPN 터널 카운터는 **Logs & Events** > **Logs** 아래의 **Tunnel & User Monitoring** 에서 SmartView Monitor로 봅니다.

> 중 요
>
> 이 기능은 Scalable Platforms(ElasticXL·Maestro·Chassis)에서는 지원되지 않 습니다. 자세한 내용은 R82 Logging and Monitoring 관리자 가이드를 참고하세요.

Cloud Security

Cloud Security 는 클라우드의 자산을 가장 정교한 위협으로부터 보호 합니다. 동적 확장성, 지능형 프로비저닝, 물리·가상 네트워크에 걸친 일관된 제어 가 특징이며, 자세한 내용은 R82 CloudGuard Controller 관리자 가이드에 있습니다(Scalable Platforms 미지원).

Advanced Routing — 동적 라우팅

Gaia OS는 다양한 라우팅을 지원합니다. 동적 라우팅 프로토콜(OSPF·BGP·RIP), 동적 멀티캐스트 라우팅(PIM-SM·PIM-DM·PIM-SSM·IGMP), 그리고 여러 라우팅 옵션을 Gaia Portal과 Gaia Clish 에서 구성합니다. 자세한 내용은 R82 Gaia Advanced Routing 관리자 가이드에 있습니다.

SNMP — 표준 모니터링 프로토콜

SNMP 는 SNMP 관리자가 GetRequest·GetNextRequest·GetBulkRequest와 일부 trap으로 장비를 모니터링 하게 해 줍니다. Check Point 구현은 SetRequest로 sysContact·sysLocation·sysName 속성을 바꾸는 것도 지원 하는데(set이 동작하려면 읽기·쓰기 권한 설정 필요), Gaia는 SNMP v1·v2·v3 를 지원합니다. 자세한 내용은 R82 Gaia 관리자 가이드의 SNMP 절을 참고하세요.

11 Monitor 모드 ·Bridge 모드 배포

Monitor 모드·Bridge 모드 배포

게이트웨이를 기존 네트워크를 건드리지 않고 끼워 넣는 두 가지 특수 배포 방식이 있습니다. 트래픽을 보기만 하는 **Monitor 모드** 와, Layer 3에 보이지 않게 두 구간을 잇는 **Bridge 모드** 입니다.

Monitor 모드 — 보기만 한다

Monitor Mode 는 게이트웨이의 한 인터페이스를 스위치의 Mirror Port(SPAN Port)에 연결해 복제된 트래픽을 듣기만 하는 방식입니다. 스위치가 트래픽을 복제해 보내면, Monitor 모드 인터페이스가 그것을 받아 활동 로그를 기록합니다. 운영 환경을 바꾸지 않고 트래픽을 분석 할 수 있죠.

쓰임새는 애플리케이션 사용 현황을 상시 모니터링 하거나 Software Blade의 성능을 평가 할 때입니다. 이 모드에서 게이트웨이는 어떤 보안 정책도 집행하지 않고, 차단·드롭·거부 같은 능동 동작을 하지 않으며, 도착한 패킷을 전달하지 않고 종료 합니다. 장점은 운영 환경에 위험이 없고, 설정이 최소이며, 비싼 TAP 장비가 필요 없 다는 것입니다.

!Monitor 모드 토폴로지 *① 모든 패킷을 복제하는 mirror/SPAN 포트를 가진 스위치 (게이트웨이가 여기에 연결) ② 서버 ③ 클라이언트 ④ Monitor 모드 인터페이스를 가진 Security Gateway ⑤ 게이트웨이를 관리하는 Security Management Server*

자세한 절차는 R82 Installation and Upgrade Guide의 "Deploying a Security Gateway in Monitor Mode"를 참고합니다.

Bridge 모드 — 보이지 않게 끼운다

Bridge Mode 는 기존 네트워크를 IP 주소가 다른 여러 네트워크로 나눌 수 없을 때 쓰는 방식입니다. Bridge 모드의 게이트웨이(또는 ClusterXL)는 Layer 3 트래픽에 보이지 않습니다. 한쪽 bridge 인터페이스로 트래픽이 들어오면 검사한 뒤 다른 쪽 bridge 인터페이스로 넘깁 니다 — 즉 하나의 네트워크를 두 개의 Layer 2 구간으로 나눠 그 사이에 투명하게 끼어드는 것입니다.

!단일 게이트웨이 Bridge 모드 토폴로지 *① 두 Layer 2 구간으로 나눌 네트워크 ② 첫 번째 네트워크 구간 ③ 첫 구간을 잇는 스위치 ④ bridge 종속 인터페이스(예: eth1) ⑤ Bridge 모드 Security Gateway ⑥ 다른 bridge 종속 인터페이스(예: eth2) ⑦ 전용 Gaia 관리 인터페이스(예: eth0) ⑧ 둘째 구간을 잇는 스위치 ⑨ 두 번째 네트워크 구간*

여기서 관리용 인터페이스(eth0)는 bridge와 별개 라는 점에 주목하세요. 자세한 절차는 R82 Installation and Upgrade Guide의 "Deploying a Security Gateway or a ClusterXL in Bridge Mode"를 참고합니다.

12 방화벽 활성화 전 보안

방화벽 활성화 전 보안

게이트웨이가 부팅 중이거나 정책이 아직 설치되지 않은 그 빈틈 에도 무방비여서는 안 됩니다. Check Point 게이트웨이는 이를 위해 두 겹의 기본 보안 을 둡니다 — 부팅 동안의 **Boot Security**(Default Filter), 그리고 첫 정책 설치 전까지의 **Initial Policy** 입니다.

> 중 요
>
> 이 장은 Scalable Platforms(ElasticXL·Maestro·Chassis)에는 해당하지 않 습니다. 또 Boot Security를 끄거나 설치된 정책을 내리면 게이트웨이가 무방비 가 되므로, 끄기 전에 게이트웨이를 네트워크에서 완전히 분리하는 것이 권장됩니다.

Boot Security와 Default Filter

Boot Security 는 부팅 동안 Linux 커널의 IP Forwarding을 끄고 Default Filter 정책을 로드 해 게이트웨이와 네트워크를 보호합니다. **Default Filter Policy(defaultfilter)** 는 게이트웨이가 켜진 순간부터 사용자 정의 정책이 설치될 때까지 를 지킵니다.

Default Filter에는 세 가지 템플릿이 있습니다. **Boot Filter**(`defaultfilter.boot`)는 게이트웨이 인터페이스 IP를 출발지로 가진 들어오는 패킷을 드롭하고 나가는 패킷은 허용 합니다. **Drop Filter**(`defaultfilter.drop`)는 들어오고 나가는 패킷을 모두 드롭 합니다 (부팅 중 다른 호스트와 통신해야 하면 쓰지 말 것). **DAG Filter**(`defaultfilter.dag`)는 동적 IP 게이트웨이용 으로, DHCP 요청·응답을 허용하면서 Boot Filter처럼 동작합니다.

Default Filter를 바꾸는 큰 줄기는 기존 파일 백업 → `$FWDIR/lib` 의 템플릿을 `$FWDIR/conf/defaultfilter.pf` 로 복사 → `fw defaultgen` 으로 컴파일 (`default.bin` / `default.bin6` 생성) → 컴파일된 파일을 부팅 경로에 복사 → 시리얼 콘솔 연결 후 재부팅 입니다. Check Point INSPECT 언어를 아는 관리자는 템플릿을 바탕으로 커스텀 Default Filter 도 만들 수 있는데, 이때 로깅·인증·암호화·Content Security 기능은 쓰면 안 됩니다.

> 참 고
>
> 유지보수로 게이트웨이를 잠시 멈출 때, `cpstop -fwflag -default` 는 Check Point 프로세스를 내리고 Default Filter를 로드 하고, `cpstop -fwflag -proc` 는 현재 커널 정책과 연결 테이블을 유지 해 `cpstart` 후 "out of state" 드롭을 막습니다.

Initial Policy

Initial Policy 는 관리자가 첫 정책을 설치하기 전까지 보안을 집행합니다. Default Filter에 미리 정의된 implied rule을 더해 만들어지며, 대부분의 통신은 막되 정책 설치에 필요한 통신만 허용 합니다. 업그레이드 중, SIC 인증서 재설정 시, 라이선스 만료 시에도 게이트웨이를 보호하며(이때는 Initial Policy가 사용자 정의 정책을 덮어씀).

부팅 순서는 이렇게 흐릅니다. 부팅 → IP Forwarding 끄고 Default Filter 로드 → 인터페이스 구성 → 서비스 시작 → 로컬에서 Initial Policy fetch → 관리자가 사용자 정의 정책 설치. 이후 재부팅에서는 Default Filter 다음에 사용자 정의 정책이 바로 로드 됩니다.

구성에 따라 Initial Policy가 허용하는 범위가 다릅니다. Standalone(관리·게이트웨이 한 장비)은 CPMI 관리 통신만 허용 해 SmartConsole이 붙게 하고, Distributed(분리)는 SIC·정책 설치를 위한 cpd·fwd 통신만 허용 하며 게이트웨이를 통한 CPMI 연결은 막습니다 (그래서 게이트웨이 너머의 Management Server에는 SmartConsole이 못 붙을 수 있음).

문제 해결 — 재부팅이 안 끝날 때

드물게 Default Filter가 부팅에 필요한 트래픽까지 막아 재부팅이 안 끝나 는 경우가 있습니다. 먼저 Default Filter가 부팅에 필요한 트래픽을 허용하는지 살피고, 그래도 안 되면 시리얼 콘솔로 접속 → 재부팅 → 부팅 중 아무 키나 눌러 Boot Menu 진입 → maintenance 모드 → `fwboot bootconf set_def` 로 Default Filter를 다시 로드하지 않게 설정 한 뒤 재부팅합니다(`boot.conf` 의 `DEFAULT_FILTER_PATH` 값 확인).

이 모든 명령(`control_bootsec` , `comp_init_policy` , `fw unloadlocal` , `cpstat -f policy fw` 등)의 자세한 사용법은 명령줄·커널 참조에서 가리키는 R82 CLI Reference Guide에 있습니다.

13 명령줄·커널 참조

명령줄·커널 참조

게이트웨이를 깊이 다루다 보면 명령줄과 커널 수준 으로 내려가야 할 때가 옵니다. 이 장은 그 세 가지 참조 영역 — **명령줄 전반**, **커널 파라미터**, **커널 디버그** — 을 개념 중심으로 정리합니다. 명령 하나하나의 방대한 목록은 전용 참조 문서에 있으니, 여기서는 무엇을 위한 것이고 어떻게 접근하는지 를 잡습니다.

명령줄 참조

게이트웨이 운영 명령 전체는 **R82 CLI Reference Guide** 에 정리되어 있습니다. 앞 장들에서 본 `fw isp_link` (ISP 이중화), `control_bootsec` · `fw unloadlocal` (방화벽 활성화 전 보안) 같은 명령도 거기에서 자세히 다룹니다. Scalable Platforms의 글로벌 명령은 해당 Scalable Platforms 가이드의 "Managing Security Groups > Global Commands" 절을 참고합니다.

커널 파라미터 다루기

커널 파라미터 는 게이트웨이의 고급 동작을 바꾸는 설정값 입니다. 정수(Integer) 또는 문자열(String) 타입이며, 게이트웨이는 그 이름과 기본값을 `$FWDIR/boot/modules/` · `$PPKDIR/boot/modules/` 의 커널 모듈 파일에서 가져옵니다.

방화벽 동작의 내부 기본값을 바꾸거나 특수 설정을 할 때 **Firewall 커널 파라미터** 를 씁니다. 적용 방법이 세 갈래 입니다. `fw ctl set` 으로 즉석에서(on-the-fly) 바꾸면 재부팅 후 사라지 고(`fw ctl set -f` 로 영구화 가능), 일부는 `$FWDIR/boot/modules/fwkern.conf` (또는 `vpnkern.conf`)에 적어야만 영구 적용 되는데 이 변경은 재부팅 후에야 효력 이 생겨 점검 창(maintenance window)이 필요합니다.

주의할 점이 있습니다. 파라미터 이름은 대소문자를 구분 하고, Cluster에서는 모든 멤버를 같은 값으로, VSX에서는 그 값이 모든 Virtual System·Virtual Router에 적용 되며, Scalable Platforms에서는 해당 Security Group에 접속 해 설정합니다. 구체적인 파라미터 이름·값은 Check Point Support Center의 여러 SK 문서나 지원팀이 제공 하므로, 임의로 바꾸기보다 안내에 따르는 것이 안전합니다.

커널 디버그

커널 디버그 는 게이트웨이가 특정 연결을 어떻게 처리하는지 보여 주는 특수 디버그 메시지를 수집 하는 작업입니다. 주로 Check Point 지원팀·R&D가 문제를 분석할 때 씁니다.

수집은 5단계 흐름 으로 진행합니다. ① 기본 디버그 설정으로 초기화하고 디버그 버퍼를 할당 → ② 필요한 디버그 모듈과 플래그를 설정(필요한 메시지만 모으도록) → ③ 디버그를 출력 파일로 수집 시작 → ④ 디버그 중지 → ⑤ 기본 설정으로 복원 입니다. 여기서도 Cluster는 모든 멤버에서 같은 방식으로, Scalable Platforms는 해당 Security Group에서 수행합니다.

구체적인 모듈·플래그 목록, 연결 수명주기(Connection Life Cycle)까지 따라가는 절차, CPU 코어가 72개 이상인 게이트웨이에서의 동작 차이 등 세부 절차는 분량이 매우 커서 원문 해당 절(Kernel Debug 장)을 참고하세요. 요지는 초기화 → 모듈·플래그 설정 → 수집 → 중지 → 복원 이라는 한 줄기이며, 이 흐름만 잡으면 세부 명령은 그 위에 얹히는 디테일입니다.