데이터센터 보안 게이트웨이 테스트 보고서

주니퍼 네트웍스 SRX5400 JUNOS 18.2X30.1 Kernel 64-bit JNPR-11.0-20190316.df99236

2019 년 11 월 12 일

글 – Keith Bormann

주니퍼 네트웍스 SRX5400 JUNOS 18.2X30.1 Kernel 64-bit JNPR-11.0-20190316.df99236

요약	2019 년 4 분기에 NSS Labs 는 주니퍼 네트웍스 SRX5400 JUNOS 18.2X30.1 Kernel 64-bit JNPR-11.0-20190316.df99236 에 대한 독립적인 테스트를 수행했습니다. 이 보고서는 데이터센터 보안 게이트웨이 테스트(DCSG) 제품에 대한 보안, 비용, 성능/기능과 같은 주요 차별화 요소에 초점을 맞춥니다.	
보안	이상적인 DCSG 는 기대되는 안정성 및 신뢰성은 물론 높은 익스플로잇 및 회피 차단율을 제공합니다.	
	보안 효과	
	익스플로잇 차단율	99.62%
	회피 차단율	126/126
	안정성 및 신뢰성	통과
	오탐	통과
성능/기능	이상적인 DCSG 는 높은 수준의 보안을 제공하는 동시에 높은 성능을 제공합니다.	
	성능	
	트랜잭션 사용 사례	10,377Mbps
	멀티미디어 사용 사례	18,176Mbps
	기업 사용 사례	13,332Mbps
	최대 용량	CPS
	이론적인 최대 동시 TCP 연결	5,638,689
	최대 TCP 연결/초	127,900
	최대 HTTP 연결/초	152,200
	최대 HTTP 트랜잭션/초	329,400
	HTTP 용량	CPS
	2,500 연결/초 – 44-KB Response	41,190
	5,000 연결/초 – 21-KB Response	63,380
	10,000 연결/초 – 10-KB Response	78,400
	20,000 연결/초 – 4.5-KB Response	101,700
	40,000 연결/초 – 1.7-KB Response	115,300
비용	이상적인 DCSG 는 확장 가능하며, 지속적인 가동 시간, 낮은 유지 보수 및 지원 비용을 제공합니다.	
	총소유비용(TCO)	
	3 년 TCO(미화)	$201,736

이 제품은 데이터센터 네트워크 보안(DCNS) 테스트 방법론 v3.1 과 회피 테스트 방법론 v1.1(www.nsslabs.com 에서 제공)을 기반으로 완전한 테스트를 거쳤습니다. 모든 NNSS Labs 그룹 테스트와 마찬가지로, 본 보고서에 설명된 테스트는 무료로 수행되었습니다.

목차

그림 목록

보안 효과

이 섹션에서는 디바이스가 보안 정책을 효율적으로 적용할 수 있는지 검증합니다.

NSS Research 에 따르면 일반적으로 DCSG 디바이스는 데이터센터 자산을 보호하기 위해 구축되며 대부분의 기업들은 DCSG 내에서 침입 방지 시스템(IPS) 모듈을 조정한다고 합니다. 따라서 독자가 예상 사용량을 기준으로 한 관련 보안 효과와 성능을 확인할 수 있도록 NSS 테스트 중 DCSG 제품은 조정된 정책 설정으로 구성됩니다.

NSS 익스플로잇 라이브러리

NSS 의 보안 효과 테스트는 필요에 따라 다양한 상용, 오픈 소스 및 전용 툴을 활용하는 당사 엔지니어들의 심도 있는 전문성을 활용합니다. 2,300 개 이상의 익스플로잇을 사용하는 이 테스트는 현재까지 업계에서 가장 포괄적인 테스트로 알려져 있습니다.

제품	실행된 총 위협의 수	차단된 총 위협의 수	차단 비율
주니퍼 네트웍스 SRX5400 JUNOS 18.2X30.1 Kernel 64-bit JNPR-11.0-20190316.df99236	2,363	2,354	99.62%

그림 1 – 차단된 위협의 수(%)

날짜별 보호 범위

그림 2 는 벤더가 성능 수준을 유지하기에 충분한 보호 시그니처를 적극적으로 만료시키고 있는지 여부에 대한 인사이트를 제공합니다. 또한, 최신 취약점 보호를 위해 제품 지연이 발생했는지 여부를 나타냅니다.

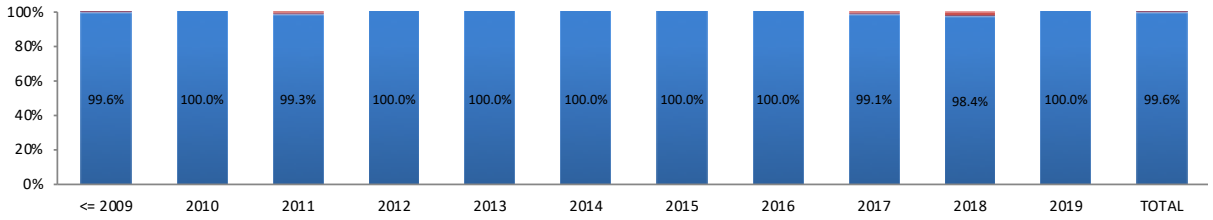

그림 2 – 날짜별 제품 보호 범위

대상 벤더별 보호 범위

NSS Exploit Library 내에 있는 익스플로잇은 다양한 범위의 프로토콜과 애플리케이션을 대상으로 합니다. 그림 3 은 이 테스트 대상인 일부 상위 벤더가 제공하는 보호 범위를 보여줍니다. 고객은 NSS 에 연락하여 보다 자세한 정보를 얻을 수 있습니다.

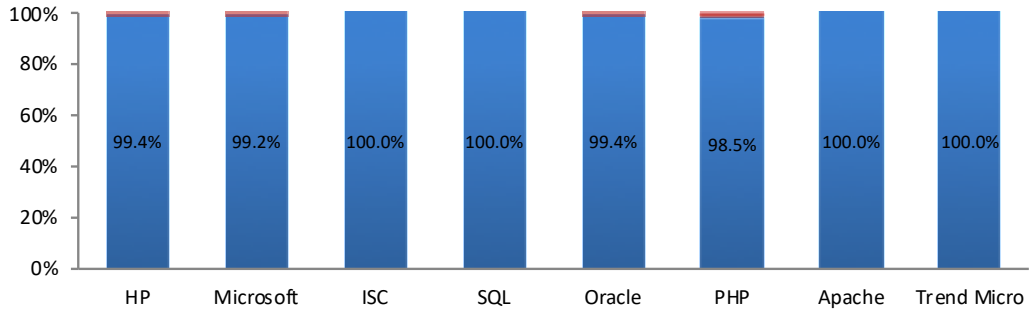

그림 3 – 대상 벤더별 제품 보호 범위

회피 기술에 대한 저항

회피 기술은 보안 솔루션에 의한 탐지 및 차단을 피하기 위해 딜리버리 위치에서 공격을 숨기고 변경하는 방법을 의미합니다. 보안 디바이스가 특정 유형의 회피를 올바르게 식별하지 못하면 잠재적으로 공격자는 보호되는 것으로 간주되는 디바이스를 대상으로 모든 유형의 익스플로잇을 사용할 가능성이 있습니다.

여러 유형의 회피(예: IP 패킷 단편화, TCP 스트림 세그먼테이션 및 RPC 회피)가 누락되는 경우가 많을 수록, 디바이스의 보호 효과가 감소합니다. 예를 들어, RPC 회피 등 한 회피 범주의 모든 기술을 놓치는 편이 각 범주에 있는 회피 기법 하나씩을 놓치는 것보다 낫습니다. 후자의 경우 더 광범위한 공격 노출이 발생합니다.

또한 네트워크 스택의 더 하위 레이어에서 작동하는 회피(IP 패킷 세그먼테이션 또는 TCP 스트림 세그먼테이션)의 경우 상위 레이어에서 작동하는 회피(예: 난독 URL)에 비해 보안 효과에 더 큰 영향을 미칩니다. 이 테스트에 사용된 많은 기법은 수년간 널리 알려져 왔기 때문에 최소한의 요건으로 간주되어야 합니다.

각 회피는 활성 익스플로잇(예: pcaps 없음)을 사용했습니다. 회피가 피해 머신의 보호 기능을 회피한 경우, 쉘을 통해 피해 머신에 액세스할 수 있게 되어 머신이 손상되었습니다. 테스트 하네스에 포함된 피해 머신에는 설치된 엔드포인트가 없었습니다.

132 건의 회피에 대해 디바이스를 테스트했으며, 이 중 126 건은 *회피* 섹션에 포함되었습니다. 나머지 6 건은 *차단율*에 포함된 복원력 회피에 있습니다.

그림 4 는 SRX5400 에 대한 회피 테스트의 결과를 제공합니다. 디바이스가 *회피* 점수를 계산하는 데 사용된 126 건의 모든 회피를 차단했습니다. 자세한 내용은 부록 A 를 참조하십시오.

테스트	결과
IP 패킷 단편화	통과
TCP 세그먼트 분할	통과
복원력	각주 참조[1]
○ 페이로드	통과
○ 트리거	통과
○ 공백	통과
RPC 단편화	통과
URL 난독 처리	통과
FTP 회피	통과
Telnet 회피	통과

그림 4 – 회피 결과에 대한 저항

[1] 복원력 테스트의 결과는 익스플로잇 차단율 계산에 포함되어 있습니다.

IP 패킷 단편화

IP 패킷 단편화	결과
소형 IP 패킷 조각, 가비지 페이로드가 있는 중복 복제 패킷 조각(서버측 익스플로잇)	통과
역순으로 소형 중복 IP 패킷 조각(서버측 익스플로잇)	통과
임의의 순서로 소형 중복 IP 패킷 조각(서버측 익스플로잇)	통과
소형 IP 패킷 조각, 첫 번째 패킷 조각 지연(서버측 익스플로잇)	통과
역순으로 소형 IP 패킷 조각, 마지막 패킷 조각 지연(서버측 익스플로잇)	통과
소형 IP 패킷 조각, 인터리브 샤프 이후(잘못된 IP 옵션) (서버측 익스플로잇)	통과
임의의 순서로 소형 IP 패킷 조각, 인터리브 샤프 샌드위치(잘못된 IP 옵션) (서버측 익스플로잇)	통과
임의의 순서로 소형 중복 IP 패킷 조각, 인터리브 샤프 샌드위치(잘못된 IP 옵션), 임의의 패킷 조각 지연(서버측 익스플로잇)	통과
소형 IP 패킷 조각, 인터리브 샤프 이전(잘못된 IP 옵션), DSCP 값 16(서버측 익스플로잇)	통과
임의의 순서로 소형 IP 패킷 조각, 인터리브 샤프 이후(잘못된 IP 옵션), 임의의 패킷 조각 지연, DSCP 값 34(서버측 익스플로잇)	통과
정크 데이터가 있는 패킷 조각 사이에 양호한 데이터가 삽입된 중복 아토믹(atomic) 패킷 조각이 있는 IPv4 단편화(서버측 익스플로잇)	통과
양호한 데이터가 있는 패킷 조각 사이에 정크 데이터가 삽입된 중복 아토믹(atomic) 패킷 조각이 있는 IPv4 단편화(서버측 익스플로잇)	통과
소형 IPv6 패킷 조각(서버측 익스플로잇)	통과
역순으로 소형 IPv6 패킷 조각(서버측 익스플로잇)	통과
임의의 순서로 소형 IPv6 패킷 조각(서버측 익스플로잇)	통과
소형 IPv6 패킷 조각, 첫 번째 패킷 조각 지연(서버측 익스플로잇)	통과
역순으로 소형 IPv6 패킷 조각, 가비지 페이로드가 있는 인터리브 복제 패킷 조각, 첫 번째 패킷 조각 지연(서버측 익스플로잇)	통과
역순으로 소형 IPv6 패킷 조각, 마지막 패킷 조각 지연(서버측 익스플로잇)	통과
역순으로 소형 IPv6 패킷 조각, 가비지 페이로드가 있는 인터리브 복제 패킷 조각, 임의의 패킷 조각 지연(서버측 익스플로잇)	통과
임의의 순서로 소형 IPv6 패킷 조각, 첫 번째 패킷 조각 지연(서버측 익스플로잇)	통과
임의의 순서로 소형 IPv6 패킷 조각, 마지막 패킷 조각 지연(서버측 익스플로잇)	통과
임의의 순서로 소형 IPv6 패킷 조각, 임의의 패킷 조각 지연(서버측 익스플로잇)	통과

그림 5 – IP 단편화 결과

공격자는 역순으로 전송하거나, 첫 번째 패킷 조각을 지연시키거나, 가비지 페이로드가 있는 중복 복제 패킷 조각을 전송하는 등의 여러 방법으로 **IP 패킷을 단편화하여** 탐지를 회피할 수 있습니다.

인터넷은 인터넷 프로토콜(IP)을 사용하여 한 컴퓨터에서 다른 컴퓨터로 트래픽을 전송하고 라우팅합니다. IP 에는 연결이 없으므로 호스트가 데이터를 교환할 준비가 되었는지 여부를 모르는 상태에서 원격 호스트로 데이터를 전송합니다. IP 에는 오류 감지/교정을 위한 시설이 없기 때문에 데이터그램 수신을 보장할 수 없습니다.

전송 중에 데이터그램이 손실되거나 손상될 가능성이 항상 있습니다. IP 데이터그램은 "있는 그대로" 수신 단말의 전송 제어 프로토콜(TCP) 계층으로 전달됩니다. 그런 다음 TCP 는 누락되거나 오류를 포함하는 데이터그램을 요청해야 합니다.

다른 기능들과 함께, IP 에는 여러 소형 패킷에서 대형 패킷으로의 단편화를 지원하는 기능이 포함되어 있습니다. 한 컴퓨터가 IP 를 사용하여 다른 컴퓨터와 통신할 때 패킷 조각을 다시 배치하는 방법에 대한 지침이 IP 헤더에 포함되어 있습니다. IP 단편화는 단일 IP 패킷을 작은 크기의 여러 패킷으로 쪼개는 프로세스입니다. 이는 IP 네트워크의 정상적인 동작이며 공격에 대한 지표 자체는 아닙니다. 따라서, 심층 검사를 수행하는 인라인 보안 솔루션은 검사를 수행하기 전에 IP 패킷 조각을 다시 조합합니다. 제품을 개발하는 프로그래머가 IP 패킷을 다시 조합하는 실수를 했을 경우(그리고 개발자는 항상 실수를 합니다), 공격자는 역순으로 전송하거나, 첫 번째 패킷 조각을 지연시키거나, 가비지 페이로드가 있는 중복 복제 패킷 조각을 전송하는 등의 여러 방법으로 IP 패킷을 단편화하여 탐지를 회피할 수 있습니다.

TCP 세그먼테이션

TCP 세그먼테이션	결과
소형 TCP 세그먼트, 가비지 페이로드가 있는 중복 복제 세그먼트(서버측 익스플로잇)	통과
역순으로 소형 TCP 세그먼트(서버측 익스플로잇)	통과
임의의 순서로 소형 TCP 세그먼트(서버측 익스플로잇)	통과
소형 TCP 세그먼트, 첫 번째 세그먼트 지연(서버측 익스플로잇)	통과
역순으로 소형 TCP 세그먼트, 마지막 세그먼트 지연(서버측 익스플로잇)	통과
소형 TCP 세그먼트, 인터리브 샤프 이후(잘못된 TCP 체크섬), 첫 번째 세그먼트 지연(서버측 익스플로잇)	통과
임의의 순서로 소형 TCP 세그먼트, 인터리브 샤프 이전(잘못된 TCP 체크섬), 임의의 세그먼트 지연(서버측 익스플로잇)	통과
임의의 순서로 소형 TCP 세그먼트, 인터리브 샤프 샌드위치(창 밖 시퀀스 번호), TCP MSS 옵션(서버측 익스플로잇)	통과
임의의 순서로 소형 TCP 세그먼트, 인터리브 샤프 이후(리싱크 시퀀스 번호 미드 스트림 요청), TCP 창 확장 옵션(서버측 익스플로잇)	통과
임의의 순서로 소형 TCP 세그먼트, 인터리브 샤프 샌드위치(리싱크 시퀀스 번호 미드 스트림 요청), TCP 창 확장 옵션, 첫 번째 세그먼트 지연(서버측 익스플로잇)	통과
소형 중복 TCP 세그먼트(서버측 익스플로잇)	통과
소형 중복 TCP 세그먼트, 방법 2(서버측 익스플로잇)	통과
소형 중복 TCP 세그먼트, 방법 3(서버측 익스플로잇)	통과
소형 TCP 세그먼트, 소형 IP 패킷 조각(서버측 익스플로잇)	통과
소형 TCP 세그먼트, 역순으로 소형 IP 패킷 조각(서버측 익스플로잇)	통과
임의의 순서로 소형 TCP 세그먼트, 소형 IP 패킷 조각(서버측 익스플로잇)	통과
소형 TCP 세그먼트, 임의의 순서로 소형 IP 패킷 조각(서버측 익스플로잇)	통과
임의의 순서로 소형 TCP 세그먼트, 역순으로 소형 IP 패킷 조각(서버측 익스플로잇)	통과
임의의 순서로 소형 TCP 세그먼트, 인터리브 샤프 샌드위치(잘못된 TCP 체크섬), 역순으로 소형 중복 IP 패킷 조각, 인터리브 샤프 이후(잘못된 IP 옵션) (서버측 익스플로잇)	통과

TCP 세그먼테이션	결과
소형 TCP 세그먼트, 인터리브 샤프 이후(잘못된 TCP 체크섬), 마지막 세그먼트 지연, 소형 IP 패킷 조각, 인터리브 샤프 이전(잘못된 IP 옵션) (서버측 익스플로잇)	통과
소형 TCP 세그먼트, 인터리브 샤프 샌드위치(잘못된 TCP 체크섬), 소형 IP 패킷 조각, 인터리브 샤프 샌드위치(잘못된 IP 옵션), 마지막 패킷 조각 지연(서버측 익스플로잇)	통과
임의의 순서로 소형 TCP 세그먼트, 인터리브 샤프 이전(창 밖 시퀀스 번호), TCP MSS 옵션, 임의의 순서로 소형 IP 패킷 조각, 인터리브 샤프 이전(잘못된 IP 옵션), 임의의 패킷 조각 지연(서버측 익스플로잇)	통과
임의의 순서로 소형 TCP 세그먼트, 인터리브 샤프 샌드위치(리싱크 시퀀스 번호 미드 스트림 요청), TCP 창 확장 옵션, 첫 번째 세그먼트 지연, 소형 IP 패킷 조각(서버측 익스플로잇)	통과
소형 중복 TCP 세그먼트, 중복 소형 패킷 조각(서버측 익스플로잇)	통과
소형 중복 TCP 세그먼트, 마지막 세그먼트 지연, 중복 소형 패킷 조각, 마지막 패킷 조각 지연(서버측 익스플로잇)	통과
소형 TCP 세그먼트, 인터리브 샤프(잘못된 IP 옵션, 잘못된 길이) (서버측 익스플로잇)	통과
소형 TCP 세그먼트, 인터리브 샤프 샌드위치(잘못된 IP 옵션, 잘못된 느슨한 소스 경로 포인터가 빈 주소 필드 이후를 지시함) (서버측 익스플로잇)	통과
소형 TCP 세그먼트, 인터리브 샤프(잘못된 IP 옵션, 잘못된 Loose Source Route 포인터가 첫 번째 주소 이전을 지시함) (서버측 익스플로잇)	통과
소형 TCP 세그먼트, 인터리브 샤프(잘못된 IP 옵션, 잘못된 Loose Source Route 포인터가 마지막 주소 이후를 지시함) (서버측 익스플로잇)	통과
소형 TCP 세그먼트, 인터리브 샤프(잘못된 IP 옵션, 잘못된 Loose Source Route 포인터가 첫 번째 주소 중간을 지시함) (서버측 익스플로잇)	통과
소형 TCP 세그먼트, 인터리브 샤프(잘못된 IP 옵션, 2 개의 Loose Source Route 옵션 초과) (서버측 익스플로잇)	통과
소형 TCP 세그먼트, 인터리브 샤프(잘못된 IP 옵션, 잘못된 Strict Source Route 포인터가 첫 번째 주소 이전을 지시함) (서버측 익스플로잇)	통과
소형 TCP 세그먼트, 인터리브 샤프(잘못된 IP 옵션, 잘못된 Strict Source Route 포인터가 마지막 주소 이후를 지시함) (서버측 익스플로잇)	통과
소형 TCP 세그먼트, 인터리브 샤프(잘못된 IP 옵션, 잘못된 Strict Source Route 포인터가 첫 번째 주소의 중간을 지시함) (서버측 익스플로잇)	통과
소형 TCP 세그먼트, 인터리브 샤프(잘못된 IP 옵션, 2 개의 Strict Source Route 옵션 초과) (서버측 익스플로잇)	통과
소형 TCP 세그먼트, 인터리브 샤프 샌드위치(잘못된 IP 옵션, 잘못된 Strict Source Route 포인터가 빈 주소 필드 이후를 지시함) (서버측 익스플로잇)	통과
소형 TCP 세그먼트, IPv6 을 통해(서버측 익스플로잇)	통과
역순으로 소형 TCP 세그먼트, IPv6 을 통해(서버측 익스플로잇)	통과
임의의 순서로 소형 TCP 세그먼트, IPv6 을 통해(서버측 익스플로잇)	통과
소형 TCP 세그먼트, 첫 번째 세그먼트 지연, IPv6 을 통해(서버측 익스플로잇)	통과
역순으로 소형 TCP 세그먼트, 마지막 세그먼트 지연, IPv6 을 통해(서버측 익스플로잇)	통과
소형 TCP 세그먼트, 인터리브 샤프(잘못된 TCP 체크섬), 첫 번째 세그먼트 지연, IPv6 을 통해(서버측 익스플로잇)	통과
임의의 순서로 소형 TCP 세그먼트, 인터리브 샤프 이후(기존 PAWS 타임 스탬프), 마지막 세그먼트 지연, IPv6 을 통해(서버측 익스플로잇)	통과
임의의 순서로 소형 TCP 세그먼트, 인터리브 샤프 이전(창 밖 시퀀스 번호), TCP MSS 옵션, IPv6 을 통해(서버측 익스플로잇)	통과

TCP 세그먼테이션	결과
임의의 순서로 소형 TCP 세그먼트, 인터리브 샤프 샌드위치(리싱크 시퀀스 번호 미드 스트림 요청), TCP Window Scale 옵션, IPv6 을 통해(서버측 익스플로잇)	통과
임의의 순서로 소형 TCP 세그먼트, 인터리브 샤프 이전(리싱크 시퀀스 번호 미드 스트림 요청), TCP Window Scale 옵션, 첫 번째 세그먼트 지연, IPv6 을 통해(서버측 익스플로잇)	통과
소형 중복 TCP 세그먼트, IPv6 을 통해(서버측 익스플로잇)	통과
소형 TCP 세그먼트, 소형 IPv6 패킷 조각(서버측 익스플로잇)	통과
소형 TCP 세그먼트, 역순으로 소형 IPv6 패킷 조각(서버측 익스플로잇)	통과
임의의 순서로 소형 TCP 세그먼트, 소형 IPv6 패킷 조각(서버측 익스플로잇)	통과
소형 TCP 세그먼트, 임의의 순서로 소형 IPv6 패킷 조각(서버측 익스플로잇)	통과
임의의 순서로 소형 TCP 세그먼트, 역순으로 소형 IPv6 패킷 조각(서버측 익스플로잇)	통과
임의의 순서로 소형 TCP 세그먼트, 인터리브 샤프 이전(잘못된 TCP 체크섬), 역순으로 소형 IPv6 패킷 조각(서버측 익스플로잇)	통과
소형 TCP 세그먼트, 인터리브 샤프 이후(잘못된 TCP 체크섬), 마지막 세그먼트 지연, 소형 IPv6 패킷 조각(서버측 익스플로잇)	통과
소형 TCP 세그먼트, 인터리브 샤프 샌드위치(잘못된 TCP 체크섬), 소형 IPv6 패킷 조각, 마지막 패킷 조각 지연(서버측 익스플로잇)	통과
임의의 순서로 소형 TCP 세그먼트, 인터리브 샤프 샌드위치(기간 외 시퀀스 번호), 임의의 순서로 소형 IPv6 패킷 조각, 임의의 패킷 조각 지연(서버측 익스플로잇)	통과
임의의 순서로 소형 TCP 세그먼트, 인터리브 샤프 이후(리싱크 시퀀스 번호 미드 스트림 요청), TCP 창 확장 옵션, 첫 번째 세그먼트 지연, 소형 IPv6 패킷 조각(서버측 익스플로잇)	통과
소형 중복 TCP 세그먼트, 소형 IPv6 패킷 조각(서버측 익스플로잇)	통과
소형 중복 TCP 세그먼트, 마지막 세그먼트 지연, 소형 IPv6 패킷 조각, 마지막 패킷 조각 지연(서버측 익스플로잇)	통과
소형 TCP 세그먼트, 인터리브 샤프(잘못된 IP 옵션, IPv6 Invalid Destination Options Extension Header)(서버측 익스플로잇)	통과
TCP 세션을 열고 61 분 동안 기다려서 회피 전송(서버측 익스플로잇)	통과
TCP 세션을 열고 소량의 애플리케이션 프로토콜 헤더 전송, 각 조각 간의 일시 중지(서버측 익스플로잇)	통과
TCP 세션을 열고 소량의 애플리케이션 프로토콜 헤더 전송, 각 조각 간의 일시 중지, IPv6 을 통해(서버측 익스플로잇)	통과

그림 6 – TCP 세그먼트 분할 결과

공격자는 역순으로 전송하거나, 첫 번째 세그먼트를 지연시키거나, 가비지 페이로드가 있는 중복 복제 세그먼트를 전송하는 등의 여러 방법으로 **TCP 스트림을 세그먼트 분할하여** 탐지를 회피할 수 있습니다. 또한 공격자는 TCP 세그먼트를 분할하고 IP 단편화를 수행하는 회피 기술을 모두 결합할 수 있습니다.

TCP 는 IP 의 상위에서 실행되는 기본 프로토콜 중 하나입니다. IP 가 스테이트리스인 경우 TCP 는 스테이트풀이며, TCP/IP 를 통해 전송 및 수신된 것을 추적한다는 의미입니다. IP 가 단편화될 수 있는 것처럼 TCP 도 단편화될 수 있습니다. 한 컴퓨터가 TCP/IP 를 사용하여 다른 컴퓨터와 통신할 때 TCP 세그먼트를 다시 배치하는 방법에 대한 지침이 TCP 헤더에 포함되어 있습니다. 이는 네트워크 트래픽 내에서 공통이며 공격 그 자체를 표시하는 것이 아닙니다.

심층 검사를 수행하는 인라인 보안 솔루션은 검사를 수행하기 전에 TCP 스트림을 리어셈블해야 합니다. 제품을 개발하는 프로그래머가 TCP 스트림을 리어셈블리하는 실수를 했을 경우, 공격자는 역순으로 전송하거나, 첫 번째 세그먼트를 지연시키거나, 가비지 페이로드가 있는 중복 복제 세그먼트를 전송하는 등의 여러 방법으로 TCP 스트림을 세그먼트 분할하여 탐지를 회피할 수 있습니다. 또한 공격자는 TCP 세그먼트를 분할하고 IP 단편화를 수행하는 회피 기술을 모두 결합할 수 있습니다.

RPC 단편화

Sun/ONC RPC 와 MS-RPC 모두를 사용하면 전송 애플리케이션이 요청을 단편화할 수 있으며, 모든 MS-RPC 서비스에는 내장형 단편화 리어셈블리 메커니즘이 있습니다.

공격자는 BIND 에 이어 악의적인 페이로드 패킷 조각이 있는 100 개의 실제 요청에 걸쳐 단편화된 단일 요청을 전송할 수 있습니다. 또는 공격자는 하나의 큰 TCP 세그먼트에서 BIND 와 요청 패킷 조각을 모두 전송할 수 있으며, 따라서 단순한 크기 검사를 사용하는 모든 시그니처를 속일 수 있습니다.

RPC 단편화	결과
1 바이트 단편화(ONC)	통과
2 바이트 단편화(ONC)	통과
마지막 패킷 조각(LF)을 포함하는 모든 패킷 조각 조각들은 하나의 TCP 세그먼트(ONC)로 전송됩니다.	통과
마지막 패킷 조각(LF)을 제외한 모든 패킷 조각 조각들은 하나의 TCP 세그먼트로 전송됩니다. LF 는 별도의 TCP 세그먼트(ONC)에서 전송됩니다.	통과
TCP 세그먼트(ONC)당 하나의 RPC 패킷 조각이 전송됩니다.	통과
하나의 LF 가 둘 이상의 TCP 세그먼트로 분할됩니다. 이 경우 RPC 단편화가 수행되지 않습니다(ONC).	통과
Canvas Reference Implementation Level 1(MS)	통과
Canvas Reference Implementation Level 2 (MS)	통과
Canvas Reference Implementation Level 3 (MS)	통과
Canvas Reference Implementation Level 4 (MS)	통과
Canvas Reference Implementation Level 5 (MS)	통과
Canvas Reference Implementation Level 6 (MS)	통과
Canvas Reference Implementation Level 7 (MS)	통과
Canvas Reference Implementation Level 8 (MS)	통과
Canvas Reference Implementation Level 9 (MS)	통과
Canvas Reference Implementation Level 10 (MS)	통과

그림 7 – RPC 단편화 결과

URL 난독화

무작위 URL 인코딩 기술은 패턴 일치 시그니처에 흔히 사용되는 단순 URL 을 변형시켜 의미 없는 이스케이프 시퀀스 문자열로 바꾸어 놓습니다. 또한 다음 기법의 조합을 사용해 경로 문자를 확장합니다.

- 이스케이프 인코딩(% 인코딩)
- Microsoft %u 인코딩
- 경로 문자 변환 및 확장(/./ , //, \ )

이러한 기법은 최소 변환에서 최대(모든 문자 변환)에 이르기까지 테스트된 각 URL 에 대해 다양한 방식으로 결합 사용됩니다. 변환된 모든 URL 은 변환 후에도 계속 예상대로 작동할 수 있도록 검증됩니다.

URL 난독화	결과
URL 인코딩 – 레벨 1(최소)	통과
URL 인코딩 – 레벨 2	통과
URL 인코딩 – 레벨 3	통과
URL 인코딩 – 레벨 4	통과
URL 인코딩 – 레벨 5	통과
URL 인코딩 – 레벨 6	통과
URL 인코딩 – 레벨 7	통과
URL 인코딩 – 레벨 8(극대)	통과
디렉토리 삽입	통과
조기 URL 종료	통과
긴 URL	통과
가짜 매개 변수	통과
탭 분리	통과
대소문자 구분	통과
Windows\구분 기호	통과
세션 스플라이싱	통과

그림 8 – URL 난독 처리 결과

FTP 및 Telnet 회피

FTP 및 텔넷 익스플로잇을 시도할 때, FTP 및 Telnet 명령에 추가 공간을 삽입하고 Telnet 컨트롤 시퀀스를 삽입하여 몇 가지 심층 검사 제품을 회피할 수 있습니다.

이러한 테스트는 RFC 를 준수하는 합법적인 서비스를 처리하고 구문 분석을 할 수 있는 유효한 Telnet 제어 시퀀스의 범위를 삽입합니다. 제어 연산 코드는 그 뒤 최소 삽입(한 쌍의 연산 코드만)에서 최대(명령의 모든 문자 사이의 연산 코드)에 이르기까지 무작위로 삽입됩니다.

FTP 및 Telnet 회피	결과
FTP 명령줄에 공백 삽입	통과
텍스트가 아닌 Telnet 연산 코드 삽입 – 레벨 1(최소)	통과
텍스트가 아닌 Telnet 연산 코드 삽입 – 레벨 2	통과
텍스트가 아닌 Telnet 연산 코드 삽입 – 레벨 3	통과
텍스트가 아닌 Telnet 연산 코드 삽입 – 레벨 4	통과
텍스트가 아닌 Telnet 연산 코드 삽입 – 레벨 5	통과
텍스트가 아닌 Telnet 연산 코드 삽입 – 레벨 6	통과
텍스트가 아닌 Telnet 연산 코드 삽입 – 레벨 7	통과
텍스트가 아닌 Telnet 연산 코드 삽입 – 레벨 8(극대)	통과

그림 9 – Telnet 및 FTP 회피 결과

성능

보안 효과와 성능은 양립할 수 없는 경우가 많습니다. 양립이 불가능하기 때문에 성능의 맥락에서 제품의 보안 효과를 판단하거나, 그 반대의 경우에서 보안 효과를 판단하는 것이 중요합니다. 그러면 새로운 보안 보호 기능이 성능에 부정적인 영향을 미쳐 이러한 보안 조치가 성능의 개선 또는 유지를 저해하지 않도록 보장할 수 있습니다.

데이터센터를 위한 보안 디바이스를 고려할 때, 이론적 제한 및 연결 역학을 이해하는 것이 핵심입니다. 데이터센터 보안 디바이스는 프라이빗 클라우드에서 애플리케이션에 액세스하는 다수의 사용자를 위한 트래픽을 처리하는 동시에 훨씬 높은 동시 연결, 초당 연결 수 및 트랜잭션을 지원해야 합니다. 스테이트리스 UDP 트래픽(네트워크 파일 시스템 [NFS]에 표시된 것과 같은) 및 장기적인 전송 제어 프로토콜(TCP) 연결(iSCSI 스토리지 영역 네트워크[SAN] 또는 백업 애플리케이션에서 보이는 바와 같이)이 높고 지속적인 부하를 제공합니다. 마지막으로, 과도한 지연으로 인해 민감한 애플리케이션 기능이 제대로 작동하지 않을 수 있습니다.

최대 용량

트래픽 생성 어플라이언스를 사용하면 NSS 엔지니어가 테스트를 위한 백그라운드 로드로 멀티기가비트 속도에서 "실제" 트래픽을 생성할 수 있습니다. 이러한 테스트의 목적은 검사 엔진에 대해 스트레스 테스트를 통해 초당 대량 TCP 연결, 초당 애플리케이션 레이어 트랜잭션, 동시에 진행 중인 연결에 대처하는 방법을 결정하는 것입니다. 모든 패킷에는 유효한 페이로드 및 주소 데이터가 포함되어 있으며, 이러한 테스트는 다양한 연결/트랜잭션 속도에서 현재 사용 중인 네트워크를 잘 보여줍니다.

모든 테스트에서는 다음 핵심 "한계점"을 사용하며, 이 한계점은 최종 측정이 발생하는 위치를 의미합니다.

- **과도한 동시 TCP 연결** – 디바이스 내의 지연으로 인해 진행 중인 연결이 수용할 수 없는 수준으로 증가됩니다.
- **과다한 동시 HTTP 연결** – 디바이스 내의 지연으로 인해 과도한 지연이 발생하고 응답 시간이 길어집니다.
- **실패한 HTTP 트랜잭션** – 일반적으로, 실패한 트랜잭션은 0 이어야 합니다. 이러한 문제가 발생하면, 디바이스 내의 과도한 지연으로 인해 연결이 시간 초과될 수 있다는 것을 의미합니다.

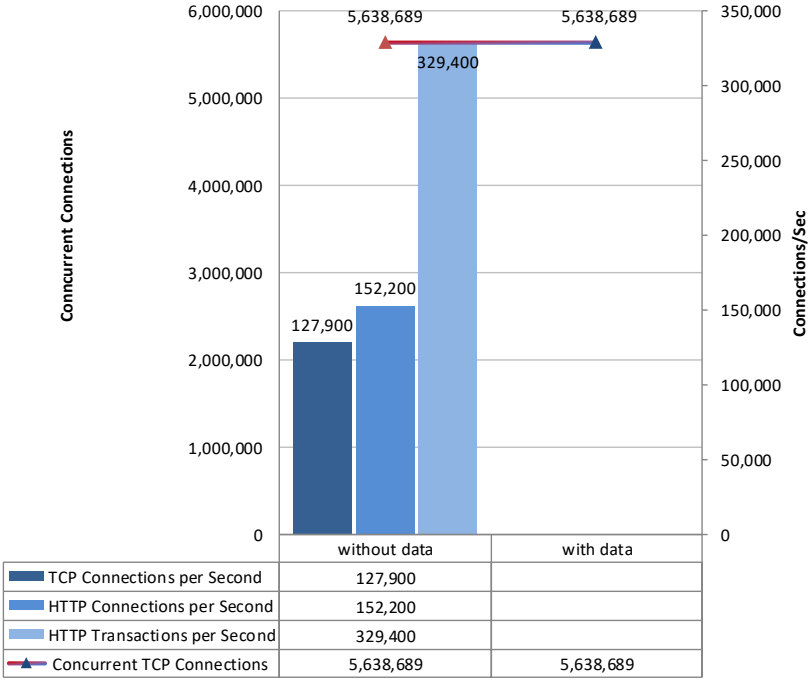

그림 10 – 최대 용량(동시성 및 연결 속도)

HTTP 용량

이 테스트의 목적은 HTTP 탐지 엔진에 대한 스트레스 테스트를 통해 다양한 평균 패킷 크기에 대한 네트워크 부하와 초마다 변화하는 연결 수에 대처하는 방법을 결정하는 것입니다. 다양한 세션 길이로 진짜 세션 기반 트래픽을 생성함으로써, 디바이스는 유효한 TCP 세션을 추적하여 단순한 패킷 기반 백그라운드 트래픽보다 높은 워크로드를 보장합니다. 이로서 가능한 한 실제 조건에 근접한 테스트 환경을 제공하고 절대적인 정확성과 반복성을 보장합니다.

각 트랜잭션은 단일 HTTP GET 요청으로 구성됩니다. 모든 패킷에는 유효한 페이로드(바이너리와 ASCII 개체의 혼합)와 주소 데이터가 포함되어 있습니다. 이 테스트는 다양한 네트워크 로드에서 현재 사용 중인 네트워크(HTTP 트래픽에 대한 단일 편향이 발생하더라도)를 잘 보여줍니다.

그림 11 은 트랜잭션 지연 테스트가 없는 HTTP 용량의 결과를 보여줍니다.

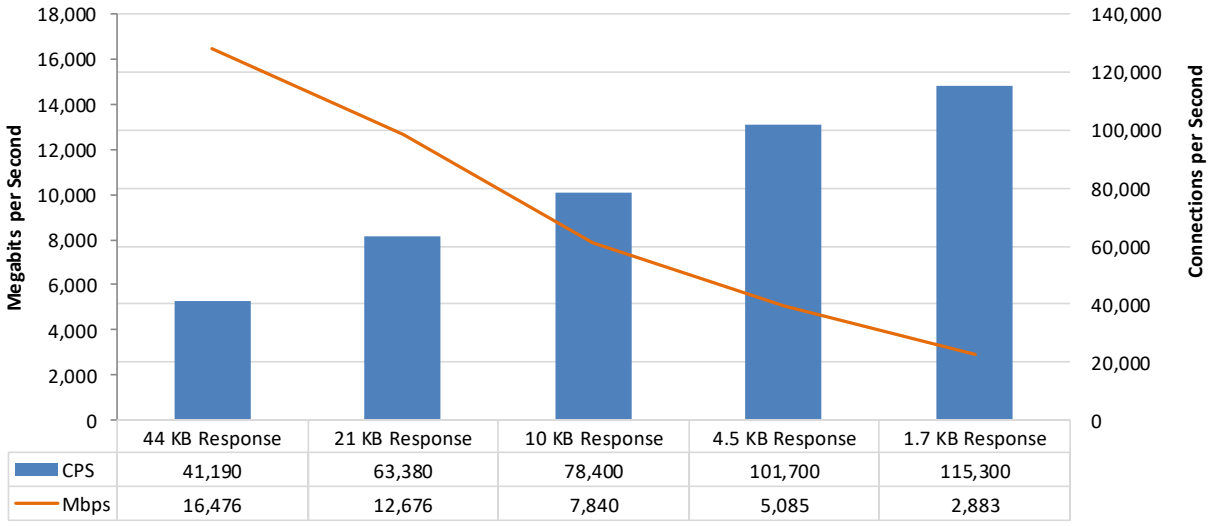

그림 11 – 트랜잭션 지연이 없는 HTTP 용량

애플리케이션 평균 응답 시간 – HTTP

애플리케이션 평균 응답 시간 – HTTP(95% 최대 부하에서)	결과
2,500 연결/초 – 44-KB Response	4.86
5,000 연결/초 – 21-KB Response	2.38
10,000 연결/초 – 10-KB Response	1.91
20,000 연결/초 – 4.5-KB Response	1.17
40,000 연결/초 – 1.7-KB Response	1.33

그림 12 – 평균 애플리케이션 응답 시간(밀리초)

HTTP 영구 연결 상태의 HTTP 용량

이 테스트의 목적은 DCSG 가 트래픽을 검사하면서 변화하는 평균 패킷 크기의 네트워크 부하와 초마다 변화하는 연결 수에 대처하는 방법을 결정하는 것입니다. 다양한 세션 길이로 진짜 세션 기반 트래픽을 생성함으로써, DCSG 는 유효한 TCP 세션을 추적하여 단순한 패킷 기반 배경 트래픽보다 높은 워크로드를 보장합니다. 이는 테스트 환경은 가능한 한 실제 조건에 가까운 테스트 환경을 제공하면서도 절대적인 정확성과 반복성을 보장합니다.

이 테스트는 10 개의 HTTP GET 및 관련 응답을 포함하는 각 TCP 연결과 함께 HTTP 영구 연결을 사용합니다. 모든 패킷에는 유효한 페이로드(바이너리와 ASCII 개체의 혼합) 및 주소 데이터가 포함되어 있으며, 이 테스트는 다양한 네트워크 부하에서 현재 사용 중인 네트워크를 잘 보여줍니다.. 명시된 응답 크기는 단일 TCP 세션 내의 모든 HTTP 응답의 합계입니다.

그림 13 은 HTTP 영구 연결 테스트가 있는 HTTP 용량의 결과를 보여줍니다.

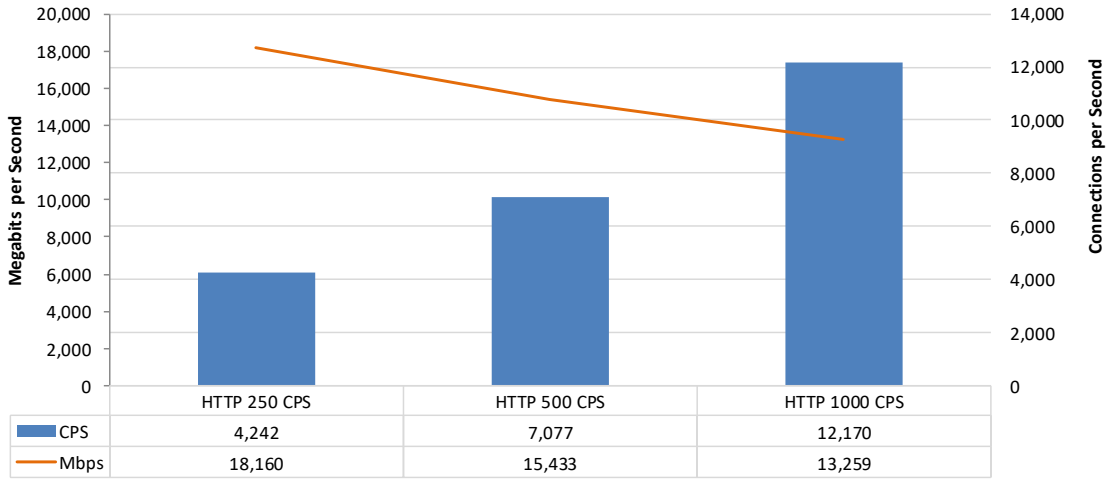

그림 13 – HTTP 영구 연결 상태의 HTTP 용량

단일 애플리케이션 플로우

이 테스트는 단일 애플리케이션 플로우를 통해 디바이스의 성능을 측정합니다. 단일 애플리케이션 플로우 테스트에 대한 자세한 내용은 NSS Labs 데이터센터 네트워크 보안(DCNS) 테스트 방법론 v3.1(www.nsslabs.com 에서 사용 가능)을 참조하십시오.그림 14 는 단일 애플리케이션 플로우 테스트의 결과를 보여줍니다.

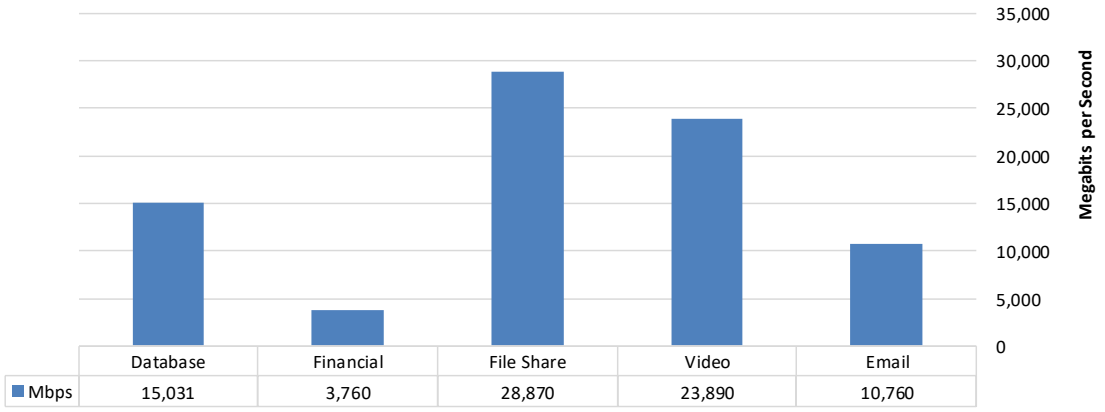

그림 14 – 단일 애플리케이션 플로우

원시 패킷 처리 성능(UDP 처리량)

이 테스트는 테스트 장비에서 생성되는 변화하는 크기의 UDP 패킷을 사용합니다. 고정 소스 포트에서 고정 대상 포트로 전송되는 가변 소스 및 대상 IP 주소를 사용하는 적절한 패킷 크기의 지속적인 스트림은 디바이스의 각 포트 쌍을 통해 양방향으로 전송됩니다.

각 패킷은 더미 데이터를 포함하고 있으며 대상 서브넷의 유효한 IP 주소에 대한 유효한 포트를 대상으로 합니다. 각 인라인 포트 쌍 전반의 초당 부하 및 프레임(fps) 백분율은 각 테스트를 시작하기 전에 네트워크 모니터링 툴에 의해 검증됩니다. 여러 테스트가 실행되고 필요한 경우 평균이 산정됩니다.

이 트래픽은 어떠한 실제 네트워크 상태도 시뮬레이션하려고 시도하지 않습니다. 이 테스트 중에는 TCP 세션이 생성되지 않으며 탐지 엔진이 수행할 작업은 거의 없습니다. 그러나 각 벤더는 테스트 패킷을 탐지하는 데 필요한 시그니처를 작성하여 탐지 엔진을 우회하고 "쉬운 노출 경로"가 되지 않도록 보장해야 합니다.

이 테스트의 목적은 디바이스의 각 인라인 포트 쌍의 원시 패킷의 처리 기능을 결정하는 것입니다. 또한 가장 적은 수준의 지연으로 최고의 네트워크 성능을 제공할 수 있도록 빠르게 패킷을 전달하는 디바이스의 효율성을 결정하는 것이기도 합니다.

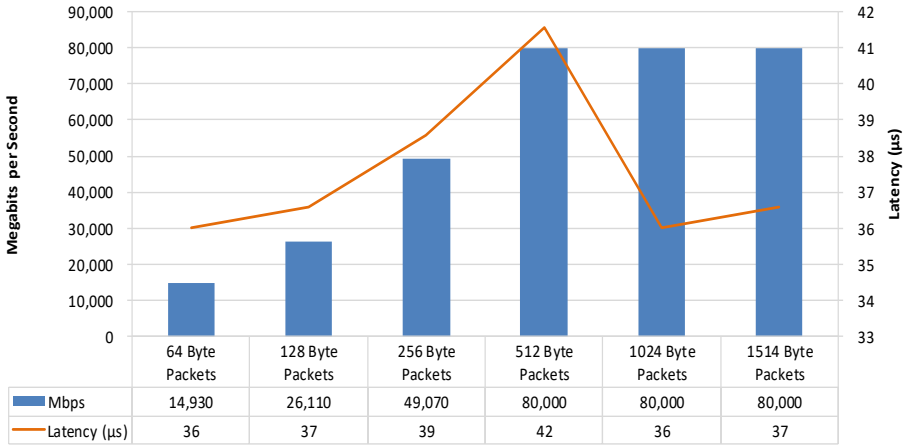

그림 15 – 원시 패킷 처리 성능 – UDP 트래픽

원시 패킷 처리 성능(UDP 지연)

높은 수준의 지연을 가져오는 DCSG 는 특히 여러 보안 디바이스가 데이터 경로에 배치되는 경우 사용자가 수용할 수 없는 응답 시간을 초래합니다. 그림 16 은 UDP 처리량 테스트 중에 최대 부하의 95%로 기록된 UDP 지연(마이크로초)을 보여줍니다. UDP 는 IPv4 를 통해 테스트되었습니다.

지연 – UDP	결과
64 바이트 패킷	36.00
128 바이트 패킷	36.57
256 바이트 패킷	38.56
512 바이트 패킷	41.55
1024 바이트 패킷	36.01
1514 바이트 패킷	36.58

그림 16 – UDP 지연(마이크로초)

NSS 테스트 처리량: 사용 사례

데이터센터 네트워크 트래픽은 업계 및 기업에 따라 크게 다를 수 있기 때문에 NSS 는 3 개의 개별 사용 사례를 작성했습니다. 데이터센터에서 볼 수 있는 다양한 사용 사례(예: 트랜잭션, 멀티미디어, 기업)에 적합하도록 각 사용 사례는 테스트 결과에 가중치를 더했습니다.

기업 사용 사례는 이메일 및 엔터프라이즈 자원 계획 소프트웨어(ERP)와 같은 미션 크리티컬 애플리케이션을 유지하는 일반적인 엔터프라이즈의 데이터 센터 공간으로 가장 잘 설명할 수 있습니다. **정격 처리량은 이메일, 데이터베이스, 파일 공유와 같은 상황에서 발견될 수 있는 다양한 패킷 크기 및 프로토콜을 말합니다.**

트랜잭션 사용 사례는 본질적으로 트랜잭션 수를 초과하는 트래픽을 포함하는 데이터센터를 반영합니다. 이 예에는 B2B(Business-to-Business) 또는 B2C(Business-to-Consumer) 전자 상거래 등이 포함될 수 있습니다. **정격 처리량은 초당 소형 패킷 크기와 연결을 잘 보여줍니다.**

멀티미디어 사용 사례는 미디어 컨텐트 처리에 사용하는 데이터센터를 반영합니다. **정격 처리량은 더 큰 패킷 크기, 최대 동시 세션, 스트리밍 프로토콜을 잘 보여줍니다.**

사용 사례	결과
트랜잭션(소형 패킷, 데이터베이스, 이메일)	10,377Mbps
멀티미디어(비디오, 대형 패킷, 데이터베이스, 이메일)	18,176Mbps
기업(이메일, 파일 공유, 데이터베이스, 혼합된 패킷 크기)	13,332Mbps

그림 17 – NSS 테스트 처리량: 사용 사례

안정성 및 신뢰성

장애가 네트워크 중단을 일으킬 수 있으므로, 장기적 안정성은 인라인 디바이스에 특히 중요합니다. 이러한 테스트는 정상적인 부하 상태 및 악성/비악성 트래픽을 검사하는 동안 보안 효과 유지 기능과 디바이스의 안정성을 검증합니다. 악의적인 공격을 받는 동안 적법한 트래픽을 유지하지 못하는 제품(또는 충돌)은 통과하지 못합니다. 안정성 및 신뢰성은 IPv4 및 IPv6 을 통해 테스트되었습니다.

디바이스는 이러한 테스트 전체에서 작동성 및 안정성을 유지하고 이전에 차단된 트래픽의 100%를 차단하여 각각에 대해 경고를 발생시켜야 합니다. 트래픽 볼륨으로 인해, 또는 어떤 이유로든 디바이스가 열리지 않으면서 허용되지 않는 트래픽이 통과하는 경우, 디바이스는 테스트를 실패하게 됩니다.

안정성 및 신뢰성	결과
공격 탐지/차단 – 정상 부하	통과
상태 보존 – 일반 부하	통과
합법적인 트래픽 통과 – 정상 부하	통과
상태 보존 – 최대 초과	통과
전원 장애	통과
데이터 지속	통과
고가용성(HA) – 옵션	통과

그림 18 – 안정성 및 신뢰성 결과

또한 이러한 테스트는 부하 상태에서의 정적 엔진 동작을 결정합니다. 리소스(예: 상태 테이블 메모리)가 부족하거나 트래픽 부하가 해당 용량을 초과하면 DCSG 디바이스는 새 연결을 끊습니다. 이론적으로, 이는 DCSG 가 합법적인 트래픽은 차단하지만 기존 연결의 상태를 유지하고 공격 누출을 방지한다는 것을 의미합니다.

총소유비용(TCO)

보안 솔루션의 구현은 구축, 유지 보수, 유지 관리에 대한 전체적인 비용에 영향을 미치는 몇 가지 요인으로 인해 복잡해질 수 있습니다. 제품의 유효 수명 동안 다음 사항을 모두 고려해야 합니다.

- **제품 구매** – 취득 비용
- **제품 유지 보수** – 소프트웨어 및 하드웨어 지원, 유지 보수 및 기타 업데이트를 포함하여 벤더에게 지불되는 비용
- **설치** – 디바이스를 박스에서 꺼내고, 구성하고, 네트워크에 설치하고, 업데이트와 패치를 적용하고, 원하는 로깅과 보고를 설정하는 데 필요한 시간
- **유지 관리** – 하드웨어, 소프트웨어, 기타 업데이트를 포함한 벤더의 주기적 업데이트 및 패치 적용에 필요한 시간
- **관리** – 디바이스 구성, 정책 업데이트, 정책 구축, 경고 처리 등을 포함한 일상적인 관리 작업

설치 시간

*설치 시간*은 로컬 관리 옵션만을 사용하여 각 디바이스를 설치하는 데 필요한 작업 시간입니다. NSS 엔지니어와 벤더의 엔지니어가 테스트 하네스에서 디바이스가 작동하고 적법한 트래픽을 통과시키고 악성/금지 트래픽을 통과시킬 수 있도록 디바이스를 설치하고 구성하는 데 소요되는 시간입니다.

제품	설치(시간)
주니퍼 네트웍스 SRX5400 JUNOS 18.2X30.1 커널 64 비트 JNPR-11.0-20190316.df99236	8

그림 19 – 디바이스 설치 시간(시간)

총소유비용

계산은 벤더에서 제공하는 가격 정보를 기반으로 합니다. 가능한 경우, 24 시간 교체를 포함한 24/7 유지 보수 및 지원 옵션이 활용됩니다. 기업 고객이 일반적으로 이 옵션을 선택합니다. 가격은 단일 디바이스 관리 및 유지 보수에만 해당되며, 중앙 관리 솔루션(CMS) 비용이 추가될 수 있습니다.

제품	구매 가격	유지 보수/년	1 년차 비용	2 년차 비용	3 년차 비용	3 년 TCO
주니퍼 네트웍스 SRX5400 JUNOS 18.2X30.1 Kernel 64-bit JNPR-11.0-20190316.df99236	$96,313	$37,013	$127,110	$37,013	$37,013	$201,736

그림 20 – 3 년 TCO(미화)

- **1 년차 비용**은 설치 비용(투입된 전체 노동력 시간 비용(미화 $75/시간) x 설치 시간) + 구매 가격 + 1 년차 유지 보수/지원 비용으로 계산됩니다.
- **2 년차 비용**은 유지 보수/지원 비용으로만 구성됩니다.
- **3 년차 비용**은 유지 보수/지원 비용으로만 구성됩니다.

CMS 를 포함한 추가 TCO 분석은 TCO 비교 보고서를 참조하십시오.

부록 A: 제품 스코어카드

보안 효과	
차단율	99.62%
오탐 테스트	통과
회피 및 공격 누출	
IP 패킷 단편화	
소형 IP 패킷 조각, 가비지 페이로드가 있는 중복 복제 패킷 조각(서버측 익스플로잇)	통과
역순으로 소형 중복 IP 패킷 조각(서버측 익스플로잇)	통과
임의의 순서로 소형 중복 IP 패킷 조각(서버측 익스플로잇)	통과
소형 IP 패킷 조각, 첫 번째 패킷 조각 지연(서버측 익스플로잇)	통과
역순으로 소형 IP 패킷 조각, 마지막 패킷 조각 지연(서버측 익스플로잇)	통과
소형 IP 패킷 조각, 인터리브 샤프 이후(잘못된 IP 옵션) (서버측 익스플로잇)	통과
임의의 순서로 소형 IP 패킷 조각, 인터리브 샤프 샌드위치(잘못된 IP 옵션) (서버측 익스플로잇)	통과
임의의 순서로 소형 중복 IP 패킷 조각, 인터리브 샤프 샌드위치(잘못된 IP 옵션), 임의의 패킷 조각 지연(서버측 익스플로잇)	통과
소형 IP 패킷 조각, 인터리브 샤프 이전(잘못된 IP 옵션), DSCP 값 16(서버측 익스플로잇)	통과
임의의 순서로 소형 IP 패킷 조각, 인터리브 샤프 이후(잘못된 IP 옵션), 임의의 패킷 조각 지연, DSCP 값 34(서버측 익스플로잇)	통과
정크 데이터가 있는 패킷 조각 사이에 양호한 데이터가 삽입된 중복 아토믹(atomic) 패킷 조각이 있는 IPv4 단편화(서버측 익스플로잇)	통과
양호한 데이터가 있는 패킷 조각 사이에 정크 데이터가 삽입된 중복 아토믹(atomic) 패킷 조각이 있는 IPv4 단편화(서버측 익스플로잇)	통과
소형 IPv6 패킷 조각(서버측 익스플로잇)	통과
역순으로 소형 IPv6 패킷 조각(서버측 익스플로잇)	통과
임의의 순서로 소형 IPv6 패킷 조각(서버측 익스플로잇)	통과
소형 IPv6 패킷 조각, 첫 번째 패킷 조각 지연(서버측 익스플로잇)	통과

역순으로 소형 IPv6 패킷 조각, 가비지 페이로드가 있는 인터리브 복제 패킷 조각, 첫 번째 패킷 조각 지연(서버측 익스플로잇)	통과
역순으로 소형 IPv6 패킷 조각, 마지막 패킷 조각 지연(서버측 익스플로잇)	통과
역순으로 소형 IPv6 패킷 조각, 가비지 페이로드가 있는 인터리브 복제 패킷 조각, 임의의 패킷 조각 지연(서버측 익스플로잇)	통과
임의의 순서로 소형 IPv6 패킷 조각, 첫 번째 패킷 조각 지연(서버측 익스플로잇)	통과
임의의 순서로 소형 IPv6 패킷 조각, 마지막 패킷 조각 지연(서버측 익스플로잇)	통과
임의의 순서로 소형 IPv6 패킷 조각, 임의의 패킷 조각 지연(서버측 익스플로잇)	통과
TCP 세그먼테이션	
소형 TCP 세그먼트, 가비지 페이로드가 있는 중복 복제 세그먼트(서버측 익스플로잇)	통과
역순으로 소형 TCP 세그먼트(서버측 익스플로잇)	통과
임의의 순서로 소형 TCP 세그먼트(서버측 익스플로잇)	통과
소형 TCP 세그먼트, 첫 번째 세그먼트 지연(서버측 익스플로잇)	통과
역순으로 소형 TCP 세그먼트, 마지막 세그먼트 지연(서버측 익스플로잇)	통과
소형 TCP 세그먼트, 인터리브 샤프 이후(잘못된 TCP 체크섬), 첫 번째 세그먼트 지연(서버측 익스플로잇)	통과
임의의 순서로 소형 TCP 세그먼트, 인터리브 샤프 이전(잘못된 TCP 체크섬), 임의의 세그먼트 지연(서버측 익스플로잇)	통과
임의의 순서로 소형 TCP 세그먼트, 인터리브 샤프 샌드위치(out-of-window 시퀀스 번호), TCP MSS 옵션(서버측 익스플로잇)	통과
임의의 순서로 소형 TCP 세그먼트, 인터리브 샤프 이후(시퀀스 번호 미드 스트림 리싱크 요청), TCP Window Scale 옵션(서버측 익스플로잇)	통과
임의의 순서로 소형 TCP 세그먼트, 인터리브 샤프 샌드위치(시퀀스 번호 미드 스트림 리싱크 요청), TCP Window Scale 옵션, 첫 번째 세그먼트 지연(서버측 익스플로잇)	통과
소형 중복 TCP 세그먼트(서버측 익스플로잇)	통과
소형 중복 TCP 세그먼트, 방법 2(서버측 익스플로잇)	통과
소형 중복 TCP 세그먼트, 방법 3(서버측 익스플로잇)	통과
소형 TCP 세그먼트, 소형 IP 패킷 조각(서버측 익스플로잇)	통과
소형 TCP 세그먼트, 역순으로 소형 IP 패킷 조각(서버측 익스플로잇)	통과
임의의 순서로 소형 TCP 세그먼트, 소형 IP 패킷 조각(서버측 익스플로잇)	통과
소형 TCP 세그먼트, 임의의 순서로 소형 IP 패킷 조각(서버측 익스플로잇)	통과
임의의 순서로 소형 TCP 세그먼트, 역순으로 소형 IP 패킷 조각(서버측 익스플로잇)	통과
임의의 순서로 소형 TCP 세그먼트, 인터리브 샤프 샌드위치(잘못된 TCP 체크섬), 역순으로 소형 중복 IP 패킷 조각, 인터리브 샤프 이후(잘못된 IP 옵션) (서버측 익스플로잇)	통과
소형 TCP 세그먼트, 인터리브 샤프 이후(잘못된 TCP 체크섬), 마지막 세그먼트 지연, 소형 IP 패킷 조각, 인터리브 샤프 이전(잘못된 IP 옵션) (서버측 익스플로잇)	통과
소형 TCP 세그먼트, 인터리브 샤프 샌드위치(잘못된 TCP 체크섬), 소형 IP 패킷 조각, 인터리브 샤프 샌드위치(잘못된 IP 옵션), 마지막 패킷 조각 지연(서버측 익스플로잇)	통과
임의의 순서로 소형 TCP 세그먼트, 인터리브 샤프 이전(창 밖 시퀀스 번호), TCP MSS 옵션, 임의의 순서로 소형 IP 패킷 조각, 인터리브 샤프 이전(잘못된 IP 옵션), 임의의 패킷 조각 지연(서버측 익스플로잇)	통과
임의의 순서로 소형 TCP 세그먼트, 인터리브 샤프 샌드위치(시퀀스 번호 미드 스트림 리싱크 요청), TCP Window Scale 옵션, 첫 번째 세그먼트 지연, 소형 IP 패킷 조각(서버측 익스플로잇)	통과
소형 중복 TCP 세그먼트, 중복 소형 패킷 조각(서버측 익스플로잇)	통과

소형 중복 TCP 세그먼트, 마지막 세그먼트 지연, 중복 소형 패킷 조각, 마지막 패킷 조각 지연(서버측 익스플로잇)	통과
소형 TCP 세그먼트, 인터리브 샤프(잘못된 IP 옵션, 잘못된 길이) (서버측 익스플로잇)	통과
소형 TCP 세그먼트, 인터리브 샤프 샌드위치(잘못된 IP 옵션, 잘못된 Loose Source Route 포인터가 빈 주소 필드 이후를 지시함) (서버측 익스플로잇)	통과
소형 TCP 세그먼트, 인터리브 샤프(잘못된 IP 옵션, 잘못된 Loose Source Route 포인터가 첫 번째 주소 이전을 지시함) (서버측 익스플로잇)	통과
소형 TCP 세그먼트, 인터리브 샤프(잘못된 IP 옵션, 잘못된 Loose Source Route 포인터가 마지막 주소 이후를 지시함) (서버측 익스플로잇)	통과
소형 TCP 세그먼트, 인터리브 샤프(잘못된 IP 옵션, 잘못된 Loose Source Route 포인터가 첫 번째 주소 중간을 지시함) (서버측 익스플로잇)	통과
소형 TCP 세그먼트, 인터리브 샤프(잘못된 IP 옵션, 2 개의 Loose Source Route 옵션 초과) (서버측 익스플로잇)	통과
소형 TCP 세그먼트, 인터리브 샤프(잘못된 IP 옵션, 잘못된 Strict Source Route 포인터가 첫 번째 주소 이전을 지시함) (서버측 익스플로잇)	통과
소형 TCP 세그먼트, 인터리브 샤프(잘못된 IP 옵션, 잘못된 Strict Source Route 포인터가 마지막 주소 이후를 지시함) (서버측 익스플로잇)	통과
소형 TCP 세그먼트, 인터리브 샤프(잘못된 IP 옵션, 잘못된 Strict Source Route 포인터가 첫 번째 주소의 중간을 지시함) (서버측 익스플로잇)	통과
소형 TCP 세그먼트, 인터리브 샤프(잘못된 IP 옵션, 2 개의 Strict Source Route 옵션 초과) (서버측 익스플로잇)	통과
소형 TCP 세그먼트, 인터리브 샤프 샌드위치(잘못된 IP 옵션, 잘못된 Strict Source Route 포인터가 빈 주소 필드 이후를 지시함) (서버측 익스플로잇)	통과
소형 TCP 세그먼트, IPv6 을 통해(서버측 익스플로잇)	통과
역순으로 소형 TCP 세그먼트, IPv6 을 통해(서버측 익스플로잇)	통과
임의의 순서로 소형 TCP 세그먼트, IPv6 을 통해(서버측 익스플로잇)	통과
소형 TCP 세그먼트, 첫 번째 세그먼트 지연, IPv6 을 통해(서버측 익스플로잇)	통과
역순으로 소형 TCP 세그먼트, 마지막 세그먼트 지연, IPv6 을 통해(서버측 익스플로잇)	통과
소형 TCP 세그먼트, 인터리브 샤프(잘못된 TCP 체크섬), 첫 번째 세그먼트 지연, IPv6 을 통해(서버측 익스플로잇)	통과
임의의 순서로 소형 TCP 세그먼트, 인터리브 샤프 이후(기존 PAWS 타임 스탬프), 마지막 세그먼트 지연, IPv6 을 통해(서버측 익스플로잇)	통과
임의의 순서로 소형 TCP 세그먼트, 인터리브 샤프 이전(out-of-window 시퀀스 번호), TCP MSS 옵션, IPv6 을 통해(서버측 익스플로잇)	통과
임의의 순서로 소형 TCP 세그먼트, 인터리브 샤프 샌드위치(리싱크 시퀀스 번호 미드 스트림 요청), TCP Window Scale 옵션, IPv6 을 통해(서버측 익스플로잇)	통과
임의의 순서로 소형 TCP 세그먼트, 인터리브 샤프 이전(리싱크 시퀀스 번호 미드 스트림 요청), TCP Window Scale 옵션, 첫 번째 세그먼트 지연, IPv6 을 통해(서버측 익스플로잇)	통과
소형 중복 TCP 세그먼트, IPv6 을 통해(서버측 익스플로잇)	통과
소형 TCP 세그먼트, 소형 IPv6 패킷 조각(서버측 익스플로잇)	통과
소형 TCP 세그먼트, 역순으로 소형 IPv6 패킷 조각(서버측 익스플로잇)	통과
임의의 순서로 소형 TCP 세그먼트, 소형 IPv6 패킷 조각(서버측 익스플로잇)	통과

소형 TCP 세그먼트, 임의의 순서로 소형 IPv6 패킷 조각(서버측 익스플로잇)	통과
임의의 순서로 소형 TCP 세그먼트, 역순으로 소형 IPv6 패킷 조각(서버측 익스플로잇)	통과
임의의 순서로 소형 TCP 세그먼트, 인터리브 샤프 이전(잘못된 TCP 체크섬), 역순으로 소형 IPv6 패킷 조각(서버측 익스플로잇)	통과
소형 TCP 세그먼트, 인터리브 샤프 이후(잘못된 TCP 체크섬), 마지막 세그먼트 지연, 소형 IPv6 패킷 조각(서버측 익스플로잇)	통과
소형 TCP 세그먼트, 인터리브 샤프 샌드위치(잘못된 TCP 체크섬), 소형 IPv6 패킷 조각, 마지막 패킷 조각 지연(서버측 익스플로잇)	통과
임의의 순서로 소형 TCP 세그먼트, 인터리브 샤프 샌드위치(창 밖 시퀀스 번호), 임의의 순서로 소형 IPv6 패킷 조각, 임의의 패킷 조각 지연(서버측 익스플로잇)	통과
임의의 순서로 소형 TCP 세그먼트, 인터리브 샤프 이후(시퀀스 번호 미드 스트림 리싱크 요청), TCP Window Scale 옵션, 첫 번째 세그먼트 지연, 소형 IPv6 패킷 조각(서버측 익스플로잇)	통과
소형 중복 TCP 세그먼트, 소형 IPv6 패킷 조각(서버측 익스플로잇)	통과
소형 중복 TCP 세그먼트, 마지막 세그먼트 지연, 소형 IPv6 패킷 조각, 마지막 패킷 조각 지연(서버측 익스플로잇)	통과
소형 TCP 세그먼트, 인터리브 샤프(잘못된 IP 옵션, IPv6 Invalid Destination Options Extension Header) (서버측 익스플로잇)	통과
TCP 세션을 열고 61 분 동안 기다려서 회피 전송(서버측 익스플로잇)	통과
TCP 세션을 열고 소량의 애플리케이션 프로토콜 헤더 전송, 각 조각 간의 일시 중지(서버측 익스플로잇)	통과
TCP 세션을 열고 소량의 애플리케이션 프로토콜 헤더 전송, 각 조각 간의 일시 중지, IPv6 을 통해(서버측 익스플로잇)	통과
복원력	
• 페이로드	
사후 익스플로잇 페이로드를 위해 Python 원-라이너 바인드 셸 리스너, 기본 익스플로잇의 nc 대신(서버측 익스플로잇)	통과
• 트리거	
시스템 () PHP 기능을 사용하여 페이로드 실행(서버측 익스플로잇)	통과
패스스루 () PHP 기능을 사용하여 페이로드 실행(서버측 익스플로잇)	통과
• White Space	
페이로드 전에 "mail[#markup]" 필드에 추가된 공백(서버측 익스플로잇)	통과
• 페이로드 + 공백	
sres-wsp-001 에서 사용되는 방법의 조합, 및 sres-pay-001(서버측 익스플로잇)	통과
• 페이로드 + 트리거 + 공백	
sres-wsp-001 에서 사용되는 방법의 조합, sres-pay-001, 및 sres-trg-001(서버측 익스플로잇)	통과
RPC 단편화	
1 바이트 단편화(ONC)	통과
2 바이트 단편화(ONC)	통과
마지막 패킷 조각(LF)을 포함하는 모든 패킷 조각은 하나의 TCP 세그먼트(ONC)로 전송됩니다.	통과
마지막 패킷 조각(LF)을 제외한 모든 패킷 조각은 하나의 TCP 세그먼트로 전송됩니다. LF 는 별도의 TCP 세그먼트(ONC)에서 전송됩니다.	통과
TCP 세그먼트(ONC) 당 하나의 RPC 패킷 조각이 전송됩니다.	통과
하나의 LF 가 둘 이상의 TCP 세그먼트로 분할됩니다. 이 경우 RPC 단편화가 수행되지 않습니다(ONC).	통과

Canvas Reference Implementation Level 1 (MS)	통과
Canvas Reference Implementation Level 2 (MS)	통과
Canvas Reference Implementation Level 3 (MS)	통과
Canvas Reference Implementation Level 4 (MS)	통과
Canvas Reference Implementation Level 5 (MS)	통과
Canvas Reference Implementation Level 6 (MS)	통과
Canvas Reference Implementation Level 7 (MS)	통과
Canvas Reference Implementation Level 8 (MS)	통과
Canvas Reference Implementation Level 9 (MS)	통과
Canvas Reference Implementation Level 10 (MS)	통과
URL 난독화	
URL 인코딩 – 레벨 1(최소)	통과
URL 인코딩 – 레벨 2	통과
URL 인코딩 – 레벨 3	통과
URL 인코딩 – 레벨 4	통과
URL 인코딩 – 레벨 5	통과
URL 인코딩 – 레벨 6	통과
URL 인코딩 – 레벨 7	통과
URL 인코딩 – 레벨 8(극대)	통과
디렉토리 삽입	통과
조기 URL 종료	통과
긴 URL	통과
가짜 매개 변수	통과
탭 분리	통과
대소문자 구분	통과
Windows \ 구분 기호	통과
세션 스플라이싱	통과
FTP 회피	
FTP 명령줄에 공백 삽입	통과
Telnet 회피	
텍스트가 아닌 Telnet 연산 코드 삽입 – 레벨 1(최소)	통과
텍스트가 아닌 Telnet 연산 코드 삽입 – 레벨 2	통과
텍스트가 아닌 Telnet 연산 코드 삽입 – 레벨 3	통과
텍스트가 아닌 Telnet 연산 코드 삽입 – 레벨 4	통과
텍스트가 아닌 Telnet 연산 코드 삽입 – 레벨 5	통과
텍스트가 아닌 Telnet 연산 코드 삽입 – 레벨 6	통과
텍스트가 아닌 Telnet 연산 코드 삽입 – 레벨 7	통과
텍스트가 아닌 Telnet 연산 코드 삽입 – 레벨 8(극대)	통과
성능	
원시 패킷 처리 성능(UDP 트래픽) (IPV4 만 해당)	Mbps
64 바이트 패킷	14,930
128 바이트 패킷	26,110

256 바이트 패킷	49,070
512 바이트 패킷	80,000
1,024 바이트 패킷	80,000
1,514 바이트 패킷	80,000
지연 – UDP	마이크로초
64 바이트 패킷	36.00
128 바이트 패킷	36.57
256 바이트 패킷	38.56
512 바이트 패킷	41.55
1,024 바이트 패킷	36.01
1,514 바이트 패킷	36.58
최대 용량	
이론적인 최대 동시 TCP 연결	5,638,689
최대 TCP 연결/초	127,900
최대 HTTP 연결/초	152,200
최대 HTTP 트랜잭션/초	329,400
트랜잭션 지연이 없는 HTTP 용량	
25,000 연결/초 – 44-KB 응답	41,190
50,000 연결/초 – 21-KB 응답	63,380
100,000 연결/초 – 10-KB 응답	78,400
200,000 연결/초 – 4.5-KB 응답	101,700
400,000 연결/초 – 1.7-KB 응답	115,300
애플리케이션 평균 응답 시간 최대 HTTP(95% 최대 부하에서)	밀리초
25,000 연결/초 – 44-KB 응답	4.86
50,000 연결/초 – 21-KB 응답	2.38
100,000 연결/초 – 10-KB 응답	1.91
200,000 연결/초 – 4.5-KB 응답	1.17
400,000 연결/초 – 1.7-KB 응답	1.33
HTTP 영구 연결 상태의 HTTP 용량	CPS
250 연결/초	4,242
500 연결/초	7,077
1,000 연결/초	12,170
단일 애플리케이션 플로우	Mbps
데이터베이스	15,031
재정	3,760
파일 공유	28,870
비디오	23,890
이메일	10,760
안정성 및 신뢰성	
공격이 있는 확장 로드 하에서의 차단	통과
로드 하에서의 상태 엔진의 동작	통과
상태 보존 – 일반 부하	통과

상태 보존 – 최대 초과	통과
전원 장애	통과
데이터 지속	통과
고가용성(HA) – 옵션	통과
총소유비용	
간편한 사용	
최초 설정(시간)	8
유지에 필요한 시간(연간 시간)	NSS Labs 에 문의
조정에 필요한 시간(연간 시간)	NSS Labs 에 문의
예상 비용	
최초 구매(테스트된 하드웨어)	$96,313
설치 인건비(@ $75/hr)	$600
유지 보수 및 지원의 연간 비용(하드웨어/소프트웨어)	$30,797
연간 업데이트 비용(IPS/AV/기타)	$6,216
최초 구매(중앙 관리 시스템)	NSS Labs 에 문의
유지 보수 및 지원의 연간 비용(중앙 관리 시스템)	NSS Labs 에 문의
관리 인건비(연간 @ $75/hr)	NSS Labs 에 문의
조정 인건비(연간 @ $75/hr)	NSS Labs 에 문의
총소유비용	
1 년차	$127,110
2 년차	$37,013
3 년차	$37,013
3 년 총소유비용	$201,736

그림 21 – 상세 스코어카드

테스트 방법론

데이터센터 네트워크 보안(DCNS) 테스트 방법론 v3.1

회피 테스트 방법론 v1.1

테스트 방법론의 복사본은 www.nsslabs.com 에서 확인할 수 있습니다.

연락처 정보

NSS Labs, Inc.
3711 South MoPac Expressway
Building 1, Suite 400
Austin, TX 78746-8022
USA
info@nsslabs.com
www.nsslabs.com

이 문서 및 기타 관련 문서는 **www.nsslabs.com** 에서 확인할 수 있습니다. 라이선스를 부여 받은 사본을 받거나 오용을 보고하려면 NSS Labs 에 문의하십시오.