ISBN: 978-89-5884-885-1 98560

망성능측정도구 설치 가이드 - NDT, BWCTL, OWAMP

(Installation Guide - NDT, BWCTL and OWAMP)

일자: 2007 년 6 월 19 일

부서: 고성능연구망사업단 연구망개발팀

제출자: 이길재, 권윤주, 석우진, 곽재승

{giljael, yulli, wjseok, jskwak}@kisti.re.kr

한국과학기술정보연구원
305-806 대전광역시 유성구 어은동 52번지
TEL (042)869-0676 / FAX (042)869-0679
www.kisti.re.kr

망성능측정도구 설치 가이드 – NDT, BWCTL, OWAMP

작성자: 이길재, 권윤주, 석우진, 곽재승

연구망개발팀, 한국과학기술정보연구원
{giljael, yulli, wjseok, jskwak}@kisti.re.kr

Abstract

본 문서는 Internet2 에서 개발한 NDT, BWCTL, OWAMPD 서버의 설치 및 그 설정 방법에 관한 Internet2 의 문서를 번역 및 편집한 문서임.

Topics

1. NDT 서버 설치 및 설정 방법
2. BWCTL 서버 설치 및 설정 방법
3. OWAMPD 서버 설치 및 설정 방법

1. NDT 서버 설치 및 설정 방법

구성요소

NDT는 두 개의 주요 구성요소로 이루어진다.

- Web100 기반 서버 프로그램
  - 테스트 엔진 및 분석 엔진
  - 옵션인 'lite' 웹 서버
- 별개인 두 개의 클라이언트 프로그램
  - 클라이언트에 설치되는 JVM 웹 브라우저(예; MS Explorer) 플러그 인을 필요로 하는 자바 애플릿(Java Applet) 기반 클라이언트
  - 특정적인 클라이언트 운영체제를 위해 실행되는 것을 요구하는 커멘트 라인(Command-line) 기반 클라이언트

이들은 서버를 가동하는데 필요한 구성요소들이다. 그들 모두 하나의 다운 가능한 tar 파일에 담겨있다. 화일들은 Internet2 End-to-End Performance Initiative (e2epi) 웹사이트

(http//e2epi.internet2.edu/ndt/download.html)에 저장되어 있다.

하드웨어 요구사양

NDT 시스템은 호스트상에 많은 설정을 요구하지 않는다. 기본적인 질문으로, 망 관리자들에 의해 구체화되는 환경에서 동작할 수 있는가? 만약, 주 클라이언트가 이더넷(Ethernet), WiFi, 혹은 페스트이더넷(Fast Ethernet) 접속 망을 가진 학교 내에 위치해 있다면, 낮은 사양의 서버만으로 충분할 것이다. 만약, 주된 목표가 기가 비트 이더넷(Gigabit Ethernet)에 연결된 호스트 및 대륙간 접속하거나 국가간 접속하는 클라이언트들을 위한 서버라면, 보다 강력한 성능의 호스트가 요구될 것이다.

단지 학교 서버를 위한 최소사양:

- 500MHz 인텔 혹은 AMD CPU
- 64 MB 메모리
- 페스트 이더넷

만약, 좀더 고성능 혹은 새로운 장비를 구입할 수 있다면, 미래를 위해 특정 서버는 다음과 같다:

- 2 GHz 혹은 그 이상의 프로세서
- 256 MB 메모리
- 기가 비트 이더넷

디스크 용량은 실행 및 로그파일들을 위해 필요하다. 하지만, 테스트 동안 관련된 디스크 입출력은 없다. 그러므로, 기본적인 리눅스 운영체제를 위한 디스크 용량이면 기본적인 NDT 서버로는 충분하다.

소프트웨어 요구사양

요구되는 Web100 향상을 위한 방법들은 다음을 통해 이루어진다:

- 리눅스 커널(Linux Kernel)
- 사용자 라이브러리

소스(source)를 컴파일(compile)하기 위해 필요한 그 외의 소프트웨어는:

- 자바(Java) SDK
- pcap 라이브러리
- 클라이언트는 자바 JRE (확인된 버전)을 사용한다.

그리고, 물론 NDT 소스파일, 즉 테스트 엔진(wen100srv)는 서버관리자 계정이 필요하다.

권장 설정사항

웹 기반 자바 애플릿을 위한 옵션이나 설정은 없다. 단지 이는 고정된 테스트 설정을 제한 시간 동안 사용자가 시작할 수 있도록 허용한다. 명령어 라인 클라이언트는 사용자가 다양한 설정을 할 수 있도록 하면서 TCP 버퍼 크기에 대한 설정들은 지원한다.

테스트 엔진은 많은 옵션들을 가진다. 그 옵션들은 관리자가 많은 기능들은 제어할 수 있도록 허용한다. 시스템 용이성을 개선할 그 기본적인 옵션들은 다음을 포함한다:

- 관리자 상태를 시작 (-a 옵션)
- 만약, 다중 네트워크 인터페이스가 존재한다면, 이더넷을 감시할 특정 인터페이스를 명시하기 위해 i 옵션을 사용한다.

그리고, 단순한 웹 서버(fakewww)를 위해, 로그파일을 생성하는 -l fn 옵션을 사용하도록 권장한다.

잠재적인 위험 요소 및 대안

비표준 커널이 요구되고 GUI 도구들은 다른 포트(port)를 감시하기 위해 사용될 수 있다는 것을 생각하라. 또한, 공용 서버들은 원격 사용자들에 의한 오류 보고를 발생한다. NDT 서버 소유자는 반드시 오류보고들에 대한 응답을 하든 이메일을 무시하든지 해야 한다. 또 다른 잠재적인 위험요소는 테스트 스트림(stream)이 IDS 경보를 가동시킬 수 있다는 것이다. 즉, 이를 위한 방안은 NDT 서버를 무시하기 위한 IDS를 설정하는 것이다.

대안방안은 다음 클라이언트 테스트를 할 수 있는 도구들을 포함한다. 예를 들어,

- 여러 웹 서버들은 사용자들이 업로드(upload) 및 다운로드(download) 속도를 측정할 수 있는 기능을 제공한다.
- Internet2/Surfnet Detective
- NCSA 어드바이저(advisor)

서버 설치

적당한 리눅스 배포버전을 이용하여 기본적인 리눅스 시스템을 설치한다. 그런 후, Web100 파일들을 Web100.org 웹사이트로부터 다운 받는다.

- http://www.web100.org/download
- RPM 형식의 미리 컴파일-빌드된 커널
- Tar 형식의 파일 패치(patch)
- 시스템 라이브러리와 tar 형식의 유틸리티 패키지(utility package)

최신 버전의 미리 컴파일-빌드된 커널 혹은 패치 파일 그리고 사용자 라이브러리를 다운 받

는다.

커널 선택

커널을 선택할 때, 사용자는 두 가지 선택사항들이 있다. 일반적인 커널을 사용하는 것과 미리 컴파일-빌드된 커널을 사용하는 것이다.

미리 컴파일 된 커널 사용하기:

미리 컴파일-빌드된 커널 RPM을 설치한다. (RedHat 패키지)

- rpm i kernel-web100-*version*.rpm

부트 로더 설정파일을 수정한다.

- /etc/grub.conf 혹은 /etc/lilo.conf

시스템을 재 부팅하고 새로운 커널을 테스트 한다.

일반 커널 컴파일-빌드 하기:

kernel.org. ftp 사이트로부터 커널 소스를 다운 받는다.

- ftp ftp.kernel.org와 적당한 리눅스 커널 디렉터리로 이동, 변경한다

/usr/src 디렉터리에 있는 커널 tar파일 압축을 풀고, Web100 커널 패치 파일 압축을 푼다. 그리고 패치 한다.

- Patch p3 <*커널 패치 파일 설치경로*>

커널 패치 파일들이 리눅스 커널의 특정 버전에 맞추어져 있다는 것은 명시하는 것이 좋다. 다운 및 패치에 필요한 커널 버전을 결정하기 위해 Web100 커널 readme 파일을 한번 검토하라.

새로운 버전의 커널을 설정하고 컴파일-빌드하라. 그때 분명, Web100 옵션이 "*Networking Options* " 에 나타나도록 하기 위해 "*Prompt for development code...* " 옵션을 가용하도록 선택해야 하는 것을 명심한다. 부트 로더 설정파일을 수정하고, 재부팅 후 새로운 시스템을 테스트한다.

[리눅스 설정 절차는 이 문서에서는 다루지 않는다. 만약 관리자가 그런 절차들에 익숙하지 않다면, 미리 컴파일-빌드된 커널을 선택하는 것이 보다 현명하다. 또한, 커널2.4.x를 위한 커널 설정 메뉴와 빌드 절차는 커널2.6.x와는 다르다.]

Web100 라이브러리 컴파일-빌드하기

Web100 사용자 라이브러리 tar 파일 압축을 푼다. 표준 GNU 자동 컴파일 명령어를 사용한다.

- 현재 디렉터리를 패키지 디렉터리로 이동한다
- make 파일을 생성한다 (./configure {—*prefix=dir*})
- 유틸리티 파일과 라이브러리 파일들을 컴파일-빌드한다 (make)
- 유틸리티 파일과 라이브러리 파일들을 설치한다 (make install)

일단 커널이 설치되고 동작되면, 자동적으로 서버와의 모든 TCP 연결상에 존재하는 데이터를 모으기 시작할 것이다. 사용자 라이브러리 파일은 시스템으로부터 그 커널 데이터의 압축 푸는데 필요한 루틴(routine)을 포함하고 있다. 관리자는 그 커널이 /usr/local/bin/gutil 프로그램이 사용됨으로써 충분히 기록하는지 살펴볼 수 있다. 이 GUI에 기반한 X-윈도우는 서버와의 어떤 TCP 연결이라도 관리자를 통해 감시하도록 해준다.

자바 소프트웨어 개발 도구(SDK) 구하기

관리자는 반드시 자바 SDK를 다운받아서 설치해야 한다. 이는 애플릿 클라이언트 소스를 빌드하는데 있어 필요하다. 그 SDK는 자바 컴파일러(javac)와 빌더(jar) 프로그램을 담고 있다. 이 실행 파일들은 경로 설정(path)이 필요하며, 그렇지 않으면 NDT 빌드 프로세스는 실패할 것이다. 관리자는 그 SDK 파일들을 시스템 상에 어느 곳이든 둘 수 있기 때문에, 반드시 수동으로 이를 해야만 한다.

SDK를 Sun Microsystems 웹 사이트(http://java.sun.com)로부터 다운 받는다. 버전 1.2.2, 1.3.1, 그리고 1.4.2는 NDT에서 테스트 되었다. 단 버전 1.4.2는 실패할 예전 JRE 클라이언트의 원인이 될지도 모른다. 그 SDK를 설치하기 위해 패키지의 지시사항들을 따른다. 일단 설치되면, SDK *bin* 디렉터리를 경로설정(path)에 추가시킨다.

- Export PATH=$PATH:/sdk/path/bin

Libcap 라이브러리 구하기

pcap 라이브러리는 네트워크 인터페이스에 네트워크 하위 계층까지 접근할 수 있도록 해준다. 그 NDT는 병목상태인 링크 속도 및 형태를 결정할 수 있는 패킷 이중 분산 기술을 사용한다. 이는 libpcap.so 루틴 라이브러리는 반드시 NDT 서버에 설치되어야 함을 의미한다. 이 라이브러리는 모든 리눅스 배포 판을 위한 표준이다. 관리자는 NDT 서버가 시스템에 빌드될 때 설치되도록 해주는지 확인해야 한다. 다음 명령어,

ls /usr/lib/libpcap*

는 라이브러리가 설치되었다면 여러 개의 파일들을 보여줄 것이다. 만약 설치 되지 않았다면, 자주 가는 미러 웹사이트를 통해 rpm을 다운받거나, http://www.tcpdump.org 웹 사이트로부터 소스를 다운받아 설치한다.

NDT 프로그램 컴파일-빌드하기

일단, 먼저 요구되는 모든 소프트웨어들이 설치되면, 관리자에게는 NDT 시스템이 생성할 준비가 된다. http://e2epi.internet2.edu/ndt/download.html에서 가장 최신 tarball 파일을 쉽게 다운받아 실행파일들을 생성하기 위한 GNU 자동 만들기 도구를 이용한다.

- 현재 디렉터리를 패키지 디렉터리로 이동한다.
- make 파일을 생성한다 (./configure {—*prefix=/some/dir*})
- 실행 가능한 파일들을 컴파일-빌드한다 (make)
- 실행 가능한 파일들을 설치한다 (make install)

이런 과정은 서버 프로그램(web100srv와 fakewww)와 클라이언트 도구(analyze, web100ctl) 모두 생성하고, 자바 클래스 및 jar 프로그램을 생성한다. make install 과정은 적당한 장소에 실행 파일들을 모두 복사해준다.

설치 맞춤화(install customizing) 하기

일단 실행 파일들이 설치되면, NDT의 웹 페이지 홈이 만들어질 필요가 있다. 그 tarball은 양식 및 웹 페이지를 맞춤화하여 생성하기 위한 간단한 쉘 스크립트를 포함한다. NDT 웹 페이지 맞춤화를 생성하기 위해서,

- 스크립트(./conf/create-html.노)를 실행한다. 그리고 프롬프트를 따른다.
- 스크립트는 설치 디렉터리로 프롬프트를 옮긴다. 그래서 *prefix*= 값 및 /ndt를 추가한다.

서버 프로세스들을 가동하기 위해 예제와 같이 conf/start.ndt 스크립트를 사용한다. 그 conf/ ndt 스크립트는 부팅 시작 시 /etc/init.d 디렉터리 안에 복사될 것이다. 한번 웹사이트 홈이 생성되면, 서버 프로세스를 가동 시킬 준비는 완료된다. 그 스크립트는 시작 프로세스를 간편화 시켜줄 것이다.

서버 맞춤화(Server Customizing)

관리자는 하나의 선택을 해야만 한다. 패키지에 포함된 웹 서버(fakewww)를 가동시키거나 풀버젼 웹 서버 (apache)를 가동해야 한다. 그 시작 스크립트들은 패키지에 포함된 웹 서버 (fakewww)에 맞추어져 있다. 이 서버는 클라이언트에게 되돌려질 파일 리스트를 포함한다. 이는 그 시스템의 공격 당하기 쉬운 부분들은 줄여주고, 악의적인 사용자들이 생성하는 요청을 처리하는 것으로부터 서버를 보호한다. 로그파일은 요청된 것들을 감시하는 데 유용하다.

포함된 웹 서버(fakewww)를 위한 몇 가지 유용한 옵션들은 다음과 같다.

- 선택적으로 포트 번호 설정 (-p80)

- 연방 모드 상에서의 실행 (-F)
- 로그 웹 요청 (-l logfile)

테스트 엔진 및 분석엔진(web100srv)는 그 시스템이 작업하도록 운영되어야만 안다. 만약 이러한 과정이 동작하지 않는다면, 사용자는 에러 메시지를 받을 것이다. 서버 사용자들은 이러한 테스트들을 실행시키기 위해 3개의 TCP 포트들을 사용한다. 관리자는 어떤 사이트 방화벽이나 라우터 필터가 이들 포트들을 통과하도록 구성되도록 해야 한다. 기본설정은 3001, 2003, 그리고 3003 포트이다. 테스트 엔진(web100srv)와 함께 사용할 몇 가지 유용한 옵션들은 다음과 같다.

- 기본 사용정보 보이기 (-a)
- 이미 정해진 설정 파일 사용하기 (-c)

서버 프로그램과 명령어 라인 클라이언트 모두 (d)플래그 디버깅을 허용한다. 디버그 메시지들은 여러 단계의 레벨로 분류 가능하며, 한 개 이상의 d 플래그가 명시될 때, 레벨은 상승한다(예를 들면, -ddd는 수준0 에서 2로 상승을 의미). 게다가 서버 프로그램은 기본적인 사용 정보 안내를 (-h) 플래그를 통해 화면에 표시함으로써 도움말을 제공한다.

연합서버(Federated Server) 생성

몇 가지 경우에서, 협조적인 형태의 다중 NDT 서버들을 가동하는 것은 유익하다. 앞에서 언급한 바와 같이, NDT 서버는 설정 및 성능 문제 모두를 규명할 수 있다. 그는 설정 문제점들은 성능문제점들에 비해 더욱 심각하고 클라이언트 가까이에서 보통 발생한다고 가정한다. 이것은 가장 가까운 NDT 서버가 설정 문제점들을 보다 쉽게 발견할 수 있음을 의미한다. NDT 서버들의 연합을 운영하는 것은 그런 작업을 달성시킨다.

간단히 각각의 NDT 서버는 모든 서버에게 연합에 있어 traceroute 형태로 실행한다.

- 파일(/tmp/traceroute.data)에 출력 결과를 저장한다.
- 그림 2.1은 예제 스크립트를 보여준다.
- 흔적들 사이에 빈 공간라인이 필요로 함을 염두한다.

일단, 이 데이터 파일이 생성되면, 그는 이중 트리 파일로 변형될 수 있다.

- /usr/local/bin/tr-mkmap b {-f fn}

이런 과정을 자동화하기 위해 같은 작업을 생성할 수도 있음을 알아두라.

동작 검사(Verifying Operation)

관리자가 모든 것들이 충분히 잘 동작함을 확신하도록 할 수 있도록 기본적인 점검사항은 다음과 같다.

1) 프로세스 테이블 검사
   A. ps auxw | grep fakewww
   B. ps auxw | grep web100srv
2) TCP 포트 상태 검사
   A. Fakewww = netstat nat | grep 7123
   B. Web100srv = netstat nat | grep 300
      i. 알림: 포트 7123과 3001은 수신대기 상태이다.
3) 서버는 올바른 네트워크 인터페이스를 감시하고 있도록 확실히 한다. 현재 web100srv 과정은 어떤 인터페이스를 감시할 지에 대해 알 필요가 있음을 염두한다. 만약, 서버가 다중 인터페이스들을 가진다면, 관리자는 주 인터페이스를 명시할 필요가 있으며 모든 테스트 트래픽은 이 인터페이스를 통과해야만 한다. 링크 방향을 위해 인터페이스를 점검한다.
   A. netstat nr

커스텀(custom) 명령어 라인 클라이언트 빌드하기

만약 필요하다면, 다중 운영체제를 위해 커스텀 명령어 라인 클라이언트를 빌드 할 수 있다.

- 최신 NDT 패키지를 다운 받아 압축을 푼다
- 자동 만들기 ./configure 명령어를 실행한다
- 디렉터리를 *src* 하위디렉터리로 이동하고, 클라이언트를 빌드한다(make web100srv). [이때 web100lib 혹은 커널은 필요하지 않다.]

이것은 선택적인 단계이다. 만약 또 다른 호스트 상에서 명령어 라인 클라이언트를 실행시키길 원한다면, 관리자는 이 단계를 실행할 필요가 있다. src하위 디렉터리로 변경하고, web100clt 와 함께 make를 실행한다. 만약, 원하지 않는다면, 서버프로그램을 위해 필요한 web100lib를 찾지 못하기 때문에 make에 있어 실패할 것이다. NDT는 여러 다른 호스트들(FreeBSD, IRIX, MacOS-10, 그리고 Cygwin) 상에서 동작해왔다 그러므로, 그는 다른 호스트들에서도 동작해야 한다.

추가적 기능들

모든 서버 프로그램들을 위한 man 페이지는 준비되어있다. 분석 프로그램은 로그 파일을 읽고 해석할 수 있으며 무엇이 일어나는지 보고한다. 결국, 서버 프로그램들은 모든 활동들을 로그파일에 기록하며 관리자에 의해 감시될 수 있다.

검사결과 얻기

각각 방향에 있어 다른 두 10초 검사들은 클라이언트와 서버 사이에서 동작한다. 어떤 진단 데이터도 수집되지 않는다. 그러면 그는 서버에서 클라이언트로 하나의 10초 검사를 실

행한다. 그러면, Web100 진단 데이터는 검사 후 수집되고, 마지막으로 요약 상태 메시지가 출력된다.

- 링크 속도 및 이중
- 정보 혹은 경고 메시지

서버가 스트림을 받고 있는 때에는 많은 원하는 데이터를 얻지 못한다는 것을 염두한다. 테스트가 끝난 후, 어떤 병목 링크가 있는지 그리고 어떤 이중 모드 상태가 존재하는지 사항을 알리면서 간략한 요약결과가 출력된다. 결국, 이중 불일치와 같은 어떤 주요한 문제도 보고된다.

검사결과 분석하기

사용자가 “statistics “ 버튼을 클릭하면, 팝업(pop-up) 창이 생성된다. 이 창은 몇 가지 추가적인 세부사항들을 보여준다.

- 달성된 송수신 전송률
- 5가지 설정 검사에 대한 세부사항 (링크 형태, 이중 모드, 혼잡, 극단적 에러, 이중 불일치 상태)
- 전송률 제한 부분 (S-R-N제한율, RTT, 손실율, 고장율)
- 교섭된 설정 (성능 향상을 위한 TCP 수정)

Web100 문제상태들(송신자 제한, 수신자 제한, 혹은 네트워크 제한) 중 하나에서의 시간 연결 확률과 같은 정보는 보고된다. 또한 손실, 고장, RTT 그리고 MSS 값 또한 보고된다. 결국, 사용 가능 및 불가능 할 수 있는 TCP 옵션들이 나열된다. 설정된 값이 아닌 교섭된 값들이 존재한다는 것을 염두한다. 미들박스(middlebox)는 사용불능 옵션들을 하나 이상 가지고 있다. NDT는 그와 같은 현상이 발생하는지 결정하기 어렵다.

“more details “ 버튼은 또하나의 팝업 메뉴창을 생성한다. 이 창은 Web100과 분석을 실행하기 위해 서버에 의해 사용된 파생된 값들의 모든 리스트를 보여준다. 클라이언트 프로그램은 단순히 미리 계산된 대답들을 출력한다. 또한, 그는 다음 정보를 조정하는 성능을 제공한다.

- Web100에 의해 수집된 개별적인 TCP 카운터들
- 조건부 테스트 요소들
- 이론적인 제한, 대역폭과 시간지연의 곱, 손실률, 그리고 버퍼 크기를 포함한 전송률 분석 부분

“report problem “ 버튼은 오류 보고를 송신하기 위한 간단한 방법을 제공한다. 이 버튼이 눌러지면, 애플릿은 기본적인 클라이언트 e-메일 프로그램을 통한 e-메일 메시지를 생

성한다. 이 메일의 내용은 자동적으로 테스트 결과에 의해 작성된다. 희망하는 바대로 사용자는 어떤 오류가 있는지 정확하게 e-메일을 통해 보고할 수 있다. 마지막으로 관리자는 아무런 동작 오류가 존재하지 않는다고 확신할 수 있도록 서버 로그 파일을 확인할 수 있다.

서버는 상태 테스트에 사용된 모든 카운터 변수들 로그파일에 기록한다.

2. BWCTL 서버 설치 및 설정 방법

이번 장은 BWCTL의 설치 및 설정에 대한 정보를 제공한다. 도구에 대한 보다 많은 내용은 http://e2epi.internet2.edu/bwctl/ 에서 찾아볼 수 있다.

구성요소

모든 것들은 하나의 다운로드 가능한 tar 파일에 담겨 있다. 그 파일은 Internet2 웹사이트, http://e2epi.internet2.edu/bwctl/download.html 에 저장되어 있다. 시스템은 다음을 포함하고 있음을 가정한다.

- FreeBSD 4.x, 5.x
- 리눅스 2.4, 2.6
- 가장 최신의 리눅스 버전 가능

하드웨어 요구사양

- CPU, 메모리, 버스 속도, NIC에 있어 제한된 요구사양은 없다.
- 하드웨어는 실행되는 테스트의 가능한 강도에 따를 것이다.
  - 보다 많은 작업 테스트들은 보다 높은 하드웨어 사양을 요구할 것이다.

Abilene일 경우에는, 인텔 SCB2 메인보드와 함께 다음을 사용한다.

- 2 x 1.266 GHz PIII, 512KB L2 cache, 133 MHz FSB
- 2 x 512 MB ECC 등록된 메모리 (인터리빙 가능한 슬롯)
- 2 x 시게이트 18 GB SCSI (ST318406LC)
- 시스커넥트 기가비트 이더넷 SK-9843 SX

우리는 시스템들과 함께 위치한 Abilene PoP들 사이에 990Mbps TCP 흐름을 지원할 수 있도록 위와 같은 구성을 사용한다. 보다 자세한 시스템 사양은 실행하려는 Iperf 테스트의 내용에 따라 다르다. Abilene PoP에 함께 위치한 망 측정 컴퓨터에 대한 보다 많은 정보를 위해서는 http://abilene.ucaid.edu/observatory/를 참고하라.

소프트웨어 요구사양

- Iperf 버전 1.7.0 혹은 2.0
- 시스템상의 NTP (ntpd) 동기화 시계
- 방화벽: 통신과 테스트를 위한 많은 포트들
- 단말 호스트를 반드시 변형되어야 한다. 다음 http://www.psc.edu/networking/perf_tune.html, http://www-didc.lbl.gov/TCP-

tuning/buffers.html 을 참고하라.

NTP는 뜻밖의 요구사항일 것이다. 하지만, 테스트들 사이에 너무 많은 시간이 낭비되지 않은 스케줄링 테스트를 위해서 지역 시계의 정확성을 위해서는 사실 필요하다.

망 요구사항

만약 방화벽을 사용한다면, 통신과 테스트를 위해 필요한 포트를 열어야 할 것이다.

- TCP/8423 (제어 통신 클라이언트에서 서버로)
- TCP/단 한번만 필요 (제어 통신 서버에서 서버로: bwctld.conf에 설정 가능한 사용 단말 포트들 범위)
- TCP/5001 (Iperf 테스트 포트 – bwctld.conf에 테스트 포트들 사용하는 설정 가능한 범위)
- UDP/5001 (Iperf 테스트 포트 – bwctld.conf에 테스트 포트들 사용하는 설정 가능한 범위)

권장 설정사항

Abilene에서는 더 이상 할 수 없을 때까지 공개하는 편이다. 대역폭 테스트 제어도구를 위해 다음이 제안된다.

- 오직 TCP로의 제한적인 테스트 (UDP는 TCP가 잘 동작하지 않을 때만 일반적으로 유용하다. 그래서 TCP로 시작하는 것이 좋다.)
- 모든 테스트들을 위해 인증된 통신이 필요 (AES 키 값)
- AES 키 값에 대해 보호한다. 그 값들은 명확한 텍스트 암호이다.

일반 보안 관련사항

본 매뉴얼에서 앞서 설명된 바와 같이, Abilene에서의 가장 큰 이슈 DoS 공격 금지이다. 일반적인 접근은 1) 해(시스템이 DoS 공격의 소스(source)가 되지 않길 원하지만 디버깅을 위해서는 가능한 유용하고 이용할 수 있길 원한다)가 되지 않고, 2) 불유쾌한 것(애매한 상태 이지만, 그들 스스로 기계를 강화하는 것)이다.

기계를 강화하는 것과 관련하여, 어떤 것도 따로 해줄 필요는 없다. 단지, 보안 패치를 통해 최신으로 항상 업데이트한다. 만약 가능하다면, 지역적 방화벽을 시스템 내에 가동하라. 하지만, 그것이 측정 결과에 영향을 주는지 확인하고, 기본적으로 대역폭 제어 테스트 도구는 단말 연결을 위해 단발적인 TCP 포트(아마도 1024가 넘은 하지만 OS에 따라 다양하다)를 사용할 것이다. 로그인할 수 있는 상황에서는 그를 제한 하는 것에 대해서 고려한다. 만약 가능 하다면, 결산 보고서 프로그램을 시스템에 대해 구축한다.

대역폭 테스트 제어기 보안 관련사항

이는 대역폭 테스트 제어기의 사용으로부터 직접적인 위험에 대한 자료들이다.

- 사용될 대역폭 제한
- 테스트 타입 (TCP/UDP) 제한

대역폭 테스트 제어기 컴파일-빌드하기

압축을 풀고 컴파일-빌드하고 설치하기 위해, 가장 최신의 tarball을 http://e2epi.internet2.edu/bwctl/download.html에서 구한다. 그리고 그 tar 파일을 압축 푼 다음, 제공된 설정 스크립트를 이용하여, 실행파일을 만들어서 설치한다.

```
% gzip    cd bwctl-$VERS.tar.gz | tar xf
% cd bwctl-$VERS
% ./configure    prefix=/ami
        # --prefix 는 단지 기본설치를 원지 않을 경우 사용한다.
        # (시스템상의 /usr/local)
% make
% make install
```

이는 설정 파일은 설치하지 않는다는 것을 염두한다.

자원 파티션 하기

자원을 보호하기 위해서는 얼마나 많은 자원을 가지고 있는지 그리고 누가 사용하기 원하는지 결정해야 한다.

- 정확히 얼마의 자원 량이 테스트 호스트에게 있어 사용가능한지 결정한다.
- 사용자들 사이에서 그 자원들을 어떻게 할당할지 결정한다.
- 얼마나 많은 타임슬롯(time slot)이 사용자 그룹에 반납되어 있는지?
- 얼마나 많은 대역폭이 있는지? 그룹당 얼마인지?
- 시스템 및 망 부하에 대해 고려한다. 만약 시스템 부하가 너무 크면 데이터의 정확성에 악영향을 줄 것이다.

계층적인 제한계층을 이용한 자원 할당

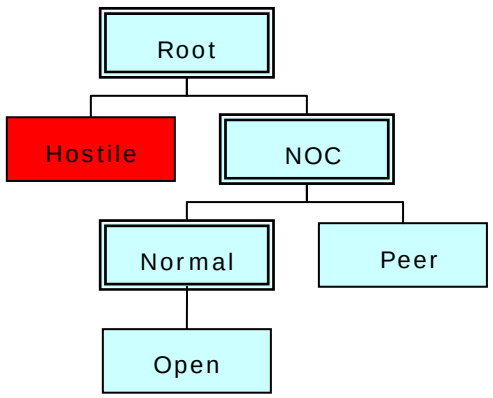
대역폭 테스트 제어기는 이용 자원이 계층적으로 할당될 수 있도록 정의된 계층적 제한계층을 허용한다.

- 사용자들은 계층적 제한 수준별로 그룹화된다.
- 일단, parent-less 계층으로 허용되면, 모든 양의 자원을 사용하도록 정의한다.
- 제한계층이 정의되면, 부모 계층으로부터 상속되며, 제한된다.

- 사용 자원이 필요로 할 때, 제한계층과 모든 부모계층 수준은 반드시 만족된다. (메모리, 대역폭)

계층적으로 조직된 제한계층의 예는 다음과 같다.

Root	모든 자원 사용가능
NOC	수퍼 사용자 제한
Peer	peer NOC 테스트를 위한 확장 제한
Normal	단말 사용자를 위한 제한
Open	누구에게도 보호적인 제한
Hostile	비우호적인 사용자에 대한 제한



특정한 사용자 그룹을 위해 만들어진 계층을 정의될 것이다. (물론, 만약 사용자 그룹이 전혀 관련되지 않는다면, 모든 그룹이 "root" 그룹의 바로 아래 자식일 때, 수평한 공간으로 정의되는 것이 가능하다.) 또 다른 가능한 계층은 다른 도메인의 사용자들이 "normal"로부터 하위제한계층을 형성함으로써 정의될 수 있다.

연결 분류하기

이는 현재 그대로 가능하면 간단하게 유지된다. 어떤 종류의 DNS 연결도 없다. 연결을 분류 하기 위해 두 가지 방법이 사용된다.

IP/넷 마스크

- 클라이언트의 IP 주소는 상세화된 서브넷과 클라이언트 주소를 기반으로 한 제한 수준에 배정된 IP 넷 마스크의 리스트와 연결된다.
- 가장 구체화된 넷 마스크는 맞추어진 알고리즘에서 이용된다.

이는 라우팅을 위한 서브넷이 될 필요는 없다. 여기에서 넷 마스크는 단지 주소 영역을 표현하는 방법일 뿐이다.

사용자 이름 및 AES 키

- 클라이언트는 사용자이름을 구체화한다. 그리고 서버는 반드시 인가 AES 키 값을 알고 있어야 한다.
- AES 키 값은 대칭적 세션 키로써 사용된다(클라이언트와 서버는 공유 비밀 키로써 그를 사용한다).

이는 기본적으로 정적 대칭 세션 키 설정이다. 대역폭 테스트 제어기는 똑같이 낮은 레벨의 도구이며, 그의 인증의 사용은 역시 명확하다. 현재 인증 방안은 보다 완전한 방법으로 통합되고 구현될 수 있도록 쉬워야만 하기 때문에 그와 같이 선택되었다. 예를 들어, PKI는 Diffie-Helman 형태 키 합의를 통해 사용된다. 이는 대역폭 테스트 제어기 프로토콜내에서 사용되는 AES 세션 키를 자동으로 할당하기 위함이다.

대역폭 테스트 제어기 서버 설정하기

bwctld를 설정하기 위한 기본 절차는 bwctld.conf를 형성한 후, 선택적이지만, bwctld.limits 파일과 bwctld.keys 파일을 생성하는 것이다. 이 파일들은 같은 디렉터리 내에 설치될 필요가 있다. 이 디렉터리는 c 옵션에 의해 bwctld에 상세화된다. 권장되는 디렉터리는 /ami/etc 이다. (etc 디렉터리는 root에 설치되어 있다.) 배포판 bwctl-$VERS/conf의 하위 디렉터리 안에 여기 파일들의 예제가 있다.

bwctld.conf 설정하기

bwctld.conf 파일은 bwctld 데몬을 위한 설정 파일이다. 이는 수신대기 포트, Iperf 을 위한 경로, 오류 기록등과 같은 서버의 기존 적인 동작을 설정하기 위한 것이다.

배포판의 conf 하위 디렉터리내의 bwctld.conf 파일 예제는 모든 가능한 옵션들을 설명하기 위해 주석처리를 잘 해놓았으며, bectld.conf 매뉴얼 페이지, http://e2epi.internet2.edu/bwctl/bwctld.conf.man.html 또한 모든 가능한 설정 옵션들을 기술해 놓았다.

대부분 설치는 다음 옵션만을 수정하여 사용할 것이다.

vardir	bwctld.pid 파일이 저장된 디렉터리
user	bwctld 프로세스가 동작할 uid 명세
group	bwctld 프로세스가 동작할 gid 명세

bwctld.limits 설정

bwctld.limits 파일은 데몬을 위한 정책 제한을 설정하기 위해 사용된다. 그는 시스템 관리자가 여러가지 방법으로 자원을 할당하도록 허용한다. 정책 설정에 두 가지 부분이 존재한다.

인증 (Authentication)

누가 요청을 하는가? 이것은 매우 개별 사용자에게 상세적이거나, 어떤 특정 망으로부터 그런 연결에 있어서는 보다 일반적일 수 있다.

인가(Authorization)

현재, bwctld는 그 연결이 일반적으로 확인되도록 허용할 것인가?

인증은 들어올 때 새로운 연결마다 제한계층을 배정함으로써 완료된다. 인가는 그와 연관된 각각의 제한 수준 셋(set)을 사용함으로써 달성된다. 그 각 제한계층에 배정된 제한들은 계층적이다. 그래서 연결은 반드시 그 배정된 제한계층뿐 아니라 모든 부모 계층의 제한을 통과해야 한다.

bwctld.limits 파일 내에, 배정 라인들은 주어진 연결로 제한계층을 배정하기 위해 사용된다. 제한라인들은 각각의 제한계층과 연관된 제한을 설정하고 그 제한계층을 정의하는데 이용된다. 그 파일은 순서대로 읽어진다. 그리고, 그는 제한 라인을 이용하여 정의되기 전에는 제한계층을 이용하도록 허용하지 않는다. 제한 계층을 정의하는 예제는 다음과 같다.

```
# total available
limit      root with \
           bandwidth=900m, \
           duration=0, \
           allow_tcp=on, \
           allow_udp=on, \
           allow_open_mode=off
# Hostile
limit      hostile with parent=root, \
           bandwidth=1, \
           allow_tcp=off, \
           allow_udp=off
# NOC
limit      noc with parent=root, \
           allow_open_mode=on
```

이 예제는 위에 기술된 계층으로부터 단지 가능한 제한계층들 중 세가지를 보여준다. 주어진 제한계층을 제한하는데 가능한 모든 설정 옵션들 셋(set)은 bwctld.limits(5) 매뉴얼 페이지 (http://e2epi.internet2.edu/bwctl/bwctld.limits.man.html).

다음 예제는 각각의 호스트들로부터 오는 연결들을 분류하기 위해 어떻게 IP/넷마스크 할당을 사용할 수 있는지 보여준다.

```
# loopback
assign net ::/127 noc
assign net 127.0.0.1/32 noc
# abilene nmslan (observatory systems)
assign net 2001:464:0::/40 noc
assign net 198.32.10.1/23 noc
assign net 10.0.0.0/16 hostile
```

이 예제는 어떻게 loopback 인터페이스로부터 서버까지 어떻게 연결이 noc 제한계층과 연관된 제한을 배정되는지 보여준다. 추가적으로, nmslan 시스템은 같은 제한계층으로 배정되어 있다. 지역 호스트에 bwctl을 사용하여 대역폭 테스트를 하는 것은 가능하지 않다. 그 이유는 수신자와 송신자 측 모두를 위해 같은 기간 동안 스케줄 슬롯(schedule slot)을 열 필요가 있을 것이기 때문이다. 그리고 bwctld는 그 시간에 열 스케줄 슬롯의 개수를 제한하기 때문이기도 하다. 그러나, 만약, 지역 bwctld를 실행할 때 지역 인증을 위해서 AES 인증을 무시통과하기 원한다면 loopback/localhost 라인은 중요하게 된다. (기억하라. 만약 지역 bwctld가 실행되지 않는다면, bwctl 은 스스로 직접 테스트를 실행할 것이다.)

이 예제는 또한 주어진 서브넷으로부터의 온 모든 연결들이 인증되지 않으면 모두 거부당하는지 어떻게 확인할 수 있는지 알려준다(10.0.0.0/16를 보라). 그 *hostile* 제한계층은 allow_open_mode를 설정하지 않는다. 그래서, 열린 모드 통신들은 이 주소 영역으로부터 수락되지 않을 것이다. 그러나, 이 서브넷의 사용자들은 아직 사용자이름/AES 키 인증을 사용할 수 있다. (만약 아무런 통신이 주어진 서브넷에 요구되지 않는다면, 그 기능성은 방화벽 응용과 함께 보다 좋게 제공된다.)

넷 마스크 배정은 너무 강하게 신뢰되면 안된다. Loopback은 정당하고 충분히 “local” 망 하지만 그외 모델을 확장하기 전에 많은 염려가 되어야 한다.

다음 예제는 주어진 사용자로부터 연결을 분류하기 위해 사용자이름 배정을 어떻게 사용할지 보여준다.

```
# network admins
        assign user joe root
        assign user jim root
        assign user bob root
# measurement geeks
        assign user boote noc
```

bwctld 서버는 주어진 사용자가 어떤 이들을 애기하는지 인증할 수 있을 필요가 있다. 이는 128-bit 공유키를 사용함 으로써 이루어진다. 키 연관으로의 사용자 이름은 아래와 같이 기술된 bwctld.keys 파일을 사용하는 bwctld에게 알려진다. 사용자는 bwctld.limits 파일에 사용되기 위해 반드시 bwctld.keys 파일 안에 포함되어 있어야 한다. 만약 사용자가 bwctld.limits에 포함되어있고, bwctld.keys에 포함되어 있지 않다면 그 bwctld 프로세스는 시작을 거부할 것이다.

bwctld.keys 설정

bwctld.keys 파일은 bwctld가 사용자를 인증하도록 필요한 두 개의 동일성/AES 키들을 담아두기 위해 사용된다. 이 파일의 형식은 aespasswd(1) 매뉴얼 페이지에 기술되어 있다. bwctld.keys 파일의 위치는 bwctld의 c 옵션에 의해 제어된다.

bwctld는 인증을 위해 대칭 AES 키들을 사용한다. 그러므로, bwctl 클라이언트는 AES 동작에 의해 인증을 위해서 정확히 동일한 AES 키를 통해 접근해야만 할 것이다. 대부분, 사용자는 단순히 처음에 만들어진 AES 키를 통과절차로만 알고 있을 것이다. 추가적으로, 그 키는 타협된 것이 아니라는 것을 단말 사용자와 시스템 관리자가 확신하는 것 또한 중요하다.

만약 bwctl 클라이언트가 나타난 AES 키와 ID를 사용해서 인증 할 수 있다면, bwctld는 이 연결에 정책적 제한을 조사하기 위해 bwctld.limits 파일에서 발견되는 지시사항을 사용할 것이다.

사용자 미름과 AES 키 규칙:

- 사용자 이름은 16 글자로 제한한다.
- AES 키는 128 bit 세션 키이다.
- AES 키는 key 파일에서 암호화 되지 않는다. 그리고 그를 보호하기 위해 유닉스 허용방안(permission)을 이용한다.
- AES 키를 발생시키기 위해 암호(passphrase)를 사용한다.
- 암호 발생 키를 key 파일에 추가시키기 위해 aespasswd를 사용한다.
- 클라이언트: 응용은 암호를 위해 사용자에게 촉구한다.

일반 유닉스 보호 방법에서는 특정 사용자 혹은 그룹 허용방안(permission)과 함께 데몬을 실행 시키려 할 것이다. 그때 그 허용방안은 key 파일을 읽을 수 있도록 허용한다. 하지만, 그에게 접근하는 사용하는 제한한다. 예제 key 파일은 다음과 같을 것이다.

```
joe     a0167ac6101b360d2f4dd164abba2337
```

bob 2d2d2a31b3e2f4af54524423f43c2323
sam 2234c35d522a4d34aae43ef4323ffa52b

이것은 단순히 16진수로 표현된 128 bit 값이다.

지금까지 bwctld.keys 파일을 생성하고 유지하는 방법을 알아온 것은 aespasswd 응용을 이용하기 위한 것이다.

aespasswd

이는 htpasswd (apache 웹서버)와 비슷하다. Key 파일에 추가될 ID를 구체화하고 암호를 위해 프롬프트 상태로 한다. 사용자는 128 bit 양을 기억하는데 쉽지 않기 때문에 용이하다. 이는 암호를 128-bit 16진수 키로 변환시키는데 사용된다. 보다 많은 정보를 원한다면, http://e2epi.internet2.edu/bwctl/aespasswd.man.html 을 보라. 이런 같은 응용은 bwctl을 위한 key 파일을 관리하기 위해 사용된다. 그리고 bwctl는 수정되는 파일을 세심히 관찰한다.

새로운 key 파일을 생성하기 위해 ‘ n’ 옵션을 사용한다.

% aespasswd n f bwctld.keys demo

추가적으로 사용자 이름은 ‘ n’ 을 통해 추가될 수 있다

% aespasswd f bwctld.keys joe

보다 자세한 정보를 원한다면 다음을 참고한다.
http://e2epi.internet2.edu/bwctl/bwctld.keys.man.html,
http://e2epi.internet2.edu/bwctl/aespasswd.man.html, 그리고
http://e2epi.internet2.edu/bwctl/bwctld.limits.man.html.

대역폭 테스트 제어기 실행

반드시 대역폭 테스트 제어기를 시험하기 위해 준비된 두 대의 호스트가 존재해야 한다. (단일 호스트로 종단간 시험은 시도될 수 없다; 그의 주요한 이유 중 하나는 단 하나의 시험이 어떤 시간에 일어나는지 확인하기 위함이다.) 그러나, 클라이언트로부터 직접 양쪽 단말 중 하나에서의 실행은 가능하다.

대역폭 테스트 제어기(bwctl) 검사하기

먼저, 클라이언트 시스템에서 internet2 호스트들 중 하나로의 간단한 시험을 시도한다. (이 호스트는 단지 시험작업 동안만 이용 가능하도록 보장된다. 때때로 여기서 새로운 것들을 시도할 것이다.) 이미 internet2 bwctld로 설정된 사용자 ID/AES 키를 가지고 있을 필요가 있을 것이다.

% /ami/bin/bwctl s nmsx-aami.abilene.ucaid. edu A AESKEY jimbob

Bwctl은 공유키로써 사용되는 AES 키를 발생하는데 이용될 사용자 jimbob의 암호를 요구할 것이다.

대역폭 테스트 제어기(bwctld) 검사하기

다음 단계는 지역 데몬을 실행시키는 것이다. '-Z' 옵션을 사용해서 시험하는 동안 전면으로데몬을 가동한다.

% /usr/local/bin/bwctld c /usr/local/etc/ Z

많은 명령어 라인 옵션들은 config 파일 파라메터를 겹쳐서 사용된다. 예를 들어, 데몬이 config 파일로부터 시작되지 않는다면 ' c' 옵션은 항상 사용될 것이다.

더 많은 정보를 필요로 한다면, 다음을 참고한다.
http://e2epi.internet2.edu/bwctl/bwctld.man.html.

이중 호스트, 이중 데몬, 단일 클라이언트

다음 단계는 전과 같이 같은 지역 호스트에서 원격 호스트에게 하기 위함이다. 하지만, 그를 실행하는 지역 bwctld를 가지는 것은 클라이언트 응용 대신 지역 종단 테스트를 관리할 것이다. (이 테스트를 실행하기 위한 또 다른 창을 사용해야 한다. 그러면 클라이언트가 실행되는 동안 bwctld 프로세스의 출력형태를 볼 수 있다.)

% /ami/bin/bwctl s nmsx-aami.abilene.ucaid.edu A AESKEY jimbob

만약 지역 IP 마스크를 이용한 인증을 그냥 넘어가지 않는다면, c 옵션을 명시적으로 사용하여 테스트의 지역호스트(localhost) 한쪽을 명시할 필요가 있을 것이다. 혹은, 만약 같은 사용자 ID와 AES 키를 Abilene 호스트처럼 지역 호스트를 위해 사용했다면, 위와 같이 s 플래그의 양쪽 단말상에 인증 정보를 추가하는 대신 테스트 양쪽 단말을 위한 인증을 명시하기 위해 A 플래그를 사용할 수 있다. (이는 후에 보다 구체적으로 설명된다.)

다른 호스트

다음 단계는 사용하는 두 호스트들 사이(혹은 호스트와 Abilene 호스트들 사이)에 검사하기 이다.

인증 옵션 테스트 하기

Bwctl 클라이언트 응용이 다중 키를 지원한다는 사실은 유용하다. 그 이유는 아마도 대칭적인 특성을 가진 "internal(내부)" 키를 외부 기관들과 공유하기를 원하지 않을지도 모르기 때문이다.

단일 인증 도메인에서 (같은 AES 키):

bwctl A AE AESKEY myname s hostA c hostB

s / c 옵션으로 인증 정보를 추가한 다른 인증 도메인들 사이에서:

bwctl s hostA AE AESKEY myname c hostB AE AESKEY *othername*

문제해결

다음은 가장 자주 일어나는 문제들이다.

1. No control connection (제어 연결 없음)
   - 데몬 실행 안 할 경우
   - 방화벽에 의해 제어 포트(4823)이 닫힐 경우
2. Control connection denied (제어 연결 거부)
   - 부적당한 설정 때문
   - 유효하지 않은 증명서 때문
3. Initial control connection works peer connection fails (초기 제어 연결 성공 동등(peer) 연결 실패)
   - 방화벽 TCP 포트의 범위를 열어주고, 동등(peer) 포트도 열어준다.
4. Scheduling problems (스케쥴링 문제점)
   - 창 내에서 슬롯을 열지 않을 경우 후의 슬롯을 기다리기 위해 L 플래그 사용
   - 올바르지 않은 NTP 설정 일 경우 'syncfuzz' 를 설정
5. Iperf connections fail (Iperf 연결 실패)
   - 방화벽 TCP/UDP 포트의 범위를 열고 iperf 포트를 열어준다.
6. Iperf results are bad or "no data" (Iperf 결과 오류– 혹은 데이터 없음)
   - 초기 창이 너무 크고, iperf는 주어진 시간 내에 종료할 수 없다. 아마도 실제 망 문

제점을 지적한다. 하지만, 그는 결과를 가능성 조차 없다. ‘ i’ 플래그를 사용하여 중간 결과 데이터를 얻도록 시도하라.

- Iperf 는 때때로 이상한 상호작용을 반복적으로 한다. 이럴 경우, 다시 시도한다.

3. OWAMPD 서버 설치 및 설정 방법

이번 장은 OWAMPD의 설치 및 설정에 대한 정보를 제공한다. 도구에 대한 보다 많은 내용은 http://e2epi.internet2.edu/owamp/ 에서 찾아볼 수 있다.

구성요소

모든 것들은 하나의 다운로드 가능한 tar 파일에 담겨 있다. 그 파일은 Internet2 웹사이트, http://e2epi.internet2.edu/owamp/download.html에 저장되어 있다. 시스템은 다음을 포함하고 있음을 가정한다.

지원 시스템

- FreeBSD 4.x, 5.x
- 리눅스 2.4, 2.6
- 가장 최신의 리눅스 버전 가능

다음 버전은 마지막 릴리즈 버전 이전에 SunOS에서 테스트 될 것이다.

요구 및 권장사항

이번 장은 하드웨어 요구사양, 소프트웨어 요구사양, 그리고 권장 설정사항을 다룬다.

하드웨어 요구사양

- 안정된 시스템 시계
  - 온도 제어 환경
  - CPU의 파워 관리 시스템 필요 없음
- CPU, 메모리, 버스 속도, NIC에 있어 제한된 요구사양은 없다.
- 보다 많은 작업 테스트들은 보다 높은 하드웨어 사양을 요구할 것이다.

안정적인 시스템 시계는 가장 중요한 부분이다. Abilene일 경우에는, 인텔 SCB2 메인보드와 함께 다음을 사용한다.

- 2 x 1.266 GHz PIII, 512KB L2 cache, 133 MHz FSB
- 2 x 512 MB ECC 등록된 메모리 (인터리빙 가능한 슬롯)
- 2 x 시게이트 18 GB SCSI (ST318406LC) Inter Ethernet Pro
- 10/100+ (i82555) (메인보드 장착됨)

우리는 Abilene PoP와 함께 위치한 시스템 상의 초당 10 packets (990Mbps TCP 흐름)

의 최소 44 개의 스트림을 지원하기 위해 이와 같은 시스템 구성을 사용한다. 그 44개 스트림은 intra-Abilene 검사를 대표한다. 외부망의 호스트들과 함께 테스트하는 그 Abilene 측정 호스트들은 추가적인 스트림에 참여한다. 그 명세된 시스템 요구사양은 특정 망 테스트에 상당히 의존한다. 보다 어렵게 짜여진 스케쥴은 보다 높은 사양의 시스템 하드웨어를 필요로 할 것이다. Abilene PoP에 함께 위치한 망 측정 컴퓨터에 대한 보다 많은 정보를 위해서는 http://abilene.ucaid.edu/observatory/를 참고하라.

소프트웨어 요구사양

만약 두 시스템에 연관된 시계들이 동기화 되어있다고 하면, 단방향 시간 지연 측정들은 일반적으로도 의미를 갖는다. (지터(jitter)와 같은 몇가지 사항들은 동기화되지 않더라도 의미를 가진다.) OWAMP는 시스템들 사이에 아주 정확한 시간을 제공하기 위해 필요한 시스템 시계를 동기화하기 위해 NTP(ntpd)에 의존한다. 그 시계는 OWAMP가 보다 정확할 수 있도록 반드시 안정적이어야만 한다. 그래서 NTP는 반드시 안정성, 회복성과 함께 설정되어야 한다. (모다 많은 세부사항을 위한다면, NTP clock book, http://e2epi.internet2.edu/library-list.html)을 참고하라. OWAMP는 가능하면 NTP-기반 시스템 콜(call)을 사용한다.

만약 방화벽을 사용한다면, 통신과 테스트를 위해 필요한 포트를 열려야 할 것이다.

- TCP/8424 (제어 통신)
- UDP/단한번만 필요 (owampd.conf에 설정 가능한 사용 테스트 포트들)

권장 설정사항

Abilene에서는 더 이상 할 수 없을 때까지 공개하는 편이다. 다음이 제안된다.

- 제한된 대역폭
- 버퍼링을 가능하게 하는 제한된 디스크 용량
- 심하지 않은 어떤 임의의 테스트 허용
- 만약에 보다 강한 테스트를 허용하기 위해 AES 키를 사용한다면, 그들을 보호하라.

일반 보안 관련사항

본 매뉴얼에서 앞서 설명된 바와 같이, Abilene에서의 가장 큰 이슈 DoS 공격 금지이다. 일반적인 접근은 1) 해(시스템이 DoS 공격의 소스(source)가 되지 않길 원하지만 디버깅을 위해서는 가능한 유용하고 이용할 수 있길 원한다)가 되지 않고, 2) 불유쾌한 것(애매한 상태 이지만, 그들 스스로 기계를 강화하는 것)이다.

기계를 강화하는 것과 관련하여, 어떤 것도 따로 해줄 필요는 없다. 단지, 보안 패치를 통해 최신으로 항상 업데이트한다. 만약 가능하다면, 지역적 방화벽을 시스템 내에 가동하라. 하

지만, 그것이 측정 결과에 영향을 주는지 확인하고, 기본적으로 OWAMP는 단말 연결을 위해 단발적인 UDP 포트(아마도 1024가 넘은 하지만 OS에 따라 다양하다)를 사용할 것이다. 로그인할 수 있는 상황에서는 그를 제한 하는 것에 대해서 고려한다. 만약 가능 하다면, 결산 보고서 프로그램을 시스템에 대해 구축한다.

OWAMP 보안 관련사항

이는 OWAMP의 사용으로부터 직접적인 위험에 대한 자료들이다.

- 사용될 대역폭 제한
- 테스트 호스트상에서 사용될 수 있는 메모리/디스크 제한

OWAMP 컴파일-빌드하기

압축을 풀고 컴파일-빌드하고 설치하기 위해, 가장 최신의 tarball을 http://e2epi.internet2.edu/owamp/download.html에서 구한다. 그리고 그 tar 파일을 압축 푼 다음, 제공된 설정 스크립트를 이용하여, 실행파일을 만들어서 설치한다.

```
% gzip    cd owamp-$VERS.tar.gz | tar xf
% cd owamp-$VERS
% ./configure    prefix=/ami
        # —prefix 는 단지 기본설치를 원지 않을 경우 사용한다.
        # (시스템상의 /usr/local)
% make
% make install
```

이는 설정 파일은 설치하지 않는다는 것을 염두한다.

자원 파티션 하기

자원을 보호하기 위해서는 얼마나 많은 자원을 가지고 있는지 그리고 누가 그를 사용하기 원하는지 결정해야 한다.

- 정확히 얼마의 자원 량이 테스트 호스트에게 있어 사용가능한지 결정한다.
- 사용자들 사이에서 그 자원들을 어떻게 할당할지 결정한다.
- 얼마나 많은 타임슬롯(time slot)이 사용자 그룹에 반납되어 있는지?
- 얼마나 많은 대역폭이 있는지? 그룹당 얼마인지?
- 시스템 부하 및 망 부하에 대해 고려한다. 만약 시스템 부하가 너무 크면 데이터의 정확성에 악영향을 줄 것이다.

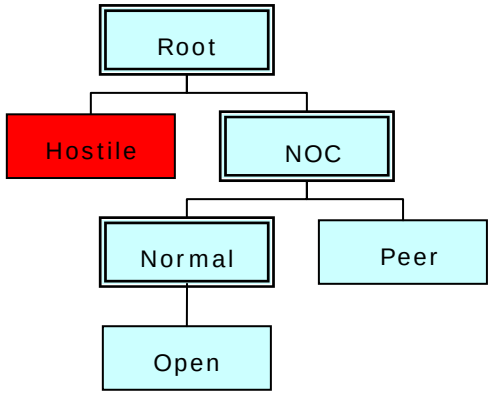
계층적인 제한계층을 이용한 자원 할당

OWAMP는 이용 자원이 계층적으로 할당될 수 있도록 정의된 계층적 제한계층을 허용한다.

- 사용자들은 계층적 제한 수준별로 그룹화된다.
- 일단, parent-less 계층으로 허용되면, 모든 양의 자원을 사용하도록 정의한다.
- 제한계층이 정의되면, 부모 계층으로부터 상속되며, 제한된다.
- 사용 자원이 필요로 할 때, 제한계층과 모든 부모계층 수준은 반드시 만족된다. (메모리, 대역폭, 타임슬롯)

계층적으로 조직된 제한계층의 예는 다음과 같다.

Root	모든 자원 사용가능
NOC	수퍼 사용자 제한
Peer	peer NOC 테스트를 위한 확장 제한
Normal	단말 사용자를 위한 제한
Open	누구에게도 보호적인 제한
Hostile	비우호적인 사용자에 대한 제한



특정한 사용자 그룹을 위해 만들어진 계층을 정의될 것이다. (물론, 만약 사용자 그룹이 전혀 관련되지 않는다면, 모든 그룹이 “root” 그룹의 바로 아래 자식일 때, 수평한 공간으로 정의되는 것이 가능하다.) 또 다른 가능한 계층은 다른 도메인의 사용자들이 “normal”로 부터 하위제한계층을 형성하므로써 정의될 수 있다.

연결 분류하기

이는 현재 그대로 가능하면 간단하게 유지된다. 어떤 종류의 DNS 연결도 없다. 연결을 분류 하기 위해 두 가지 방법이 사용된다.

IP/넷 마스크

- 클라이언트의 IP 주소는 상세화된 서브넷과 클라이언트 주소를 기반으로 한 제한 수준에 배정된 IP 넷 마스크의 리스트와 연결된다.
- 가장 구체화된 넷 마스크는 맞추어진 알고리즘에서 이용된다.

이는 라우팅을 위한 서브넷이 될 필요는 없다. 여기에서 넷 마스크는 단지 주소 영역을 표현하는 방법일 뿐이다.

사용자 이름 및 AES 키

- 클라이언트는 사용자이름을 구체화한다. 그리고 서버는 반드시 인가 AES 키 값을 알고 있어야 한다.
- AES 키 값은 대칭적 세션 키로써 사용된다(클라이언트와 서버는 공유 비밀 키로써 그를 사용한다).

이는 기본적으로 정적 대칭 세션 키 설정이다. OWAMP는 똑같이 낮은 레벨의 도구이며, 그의 인증의 사용은 똑같이 명확하다. 현재 인증 방안은 보다 완전한 방법 통합되고 구현될 수 있도록 쉬워야만 하기 때문에 그와 같이 선택되었다. 예를 들어, PKI는 Diffie-Helman 형태 키 합의를 통해 사용된다. 이는 OWAMP 프로토콜내에서 사용되는 AES 세션 키를 자동으로 할당하기 위함이다.

owampd 서버 설정하기

owampd를 설정하기 위한 기본 절차는 owampd.conf를 형성한 후, 선택적이지만, owampd.limits 파일과 owampd.keys 파일을 생성하는 것이다. 이 파일들은 같은 디렉터리 내에 설치될 필요가 있다. 이 디렉터리는 c 옵션에 의해 owampd에 상세화된다. 권장되는 디렉터리는 /ami/etc 이다. (etc 디렉터리는 root에 설치되어 있다.) 배포판 owamp-$VERS/conf의 하위 디렉터리 안에 여기 파일들의 예제가 있다.

owampd.conf 설정하기

owampd.conf 파일은 owampd 데몬을 위한 설정 파일이다. 그것은 서버의 기본적인 동작을 설정하기 위해 사용된다. 예를 들어, 무슨 주소와 무슨 포트를 그는 수신대기 해야 하는지, 어디서 에러메시지를 송신하는지, 그리고 어디서 파일을 저장하는지 등이다.

배포판의 conf 하위 디렉터리내의 owampd.conf 파일 예제는 모든 가능한 옵션들을 설명하기 위해 주석처리를 잘 해놓았으며, owampd.conf 매뉴얼 페이지, http://e2epi.internet2.edu/owamp/owampd.conf.man.html 또한 모든 가능한 설정 옵션들을 기술해 놓았다.

대부분 설치는 다음 옵션만을 수정하여 사용할 것이다.

datadir	서버에 의해 수신된 테스트 세션을 위한 버퍼에 저장된 데이터를 담아두는 디렉터리
vardir	owampd.info 파일과 owampd.pid 파일이 저장된 디렉터리
user	owampd 프로세스가 동작할 uid 명세
group	owampd 프로세스가 동작할 gid 명세

owampd.limits 설정

owampd.limits 파일은 데몬을 위한 정책 제한을 설정하기 위해 사용된다. 그는 시스템 관리자가 여러가지 방법으로 자원을 할당하도록 허용한다. 정책 설정에 두가지 부분이 존재한다.

인증 (Authentication)

누가 요청을 하는가? 이것은 매우 개별 사용자에게 상세적이거나, 어떤 특정 망으로부터 그런 연결에 있어서는 보다 일반적일 수 있다.

인가(Authorization)

현재, owampd는 그 연결이 일반적으로 확인되도록 허용할 것인가?

인증은 들어올 때 새로운 연결마다 제한계층을 배정함으로써 완료된다. 인가는 그와 연관된 각각의 제한 수준 셋(set)을 사용함으로써 달성된다. 그 각 제한계층에 배정된 제한들은 계층적이다. 그래서 연결은 반드시 그 배정된 제한계층뿐 아니라 모든 부모 계층의 제한을 통과해야 한다.

owampd.limits 파일 내에, 배정 라인들은 주어진 연결로 제한계층을 배정하기 위해 사용된다. 제한라인들은 각각의 제한계층과 연관된 제한을 설정하고 그 제한계층을 정의하는데 이용된다. 그 파일은 순서대로 읽어진다. 그리고, 그는 제한 라인을 이용하여 정의되기 전에는 제한계층을 이용하도록 허용하지 않는다. 제한 계층 정의 예제는 다음과 같다.

```
# total available
limit     root with \
          disk=100m, \
          bandwidth=0, \
          delete_on_fetch=on, \
          allow_open_mode=off
# Hostile
limit     hostile with parent=root, \
          disk=1, \
          bandwidth=1
# NOC
limit     noc with parent=root, \
          allow_open_mode=on
```

이 예제는 위에 기술된 계층으로부터 단지 가능한 제한계층들 중 세가지를 보여준다. 주어진 제한계층을 제한하는데 가능한 모든 설정 옵션들 셋(set)은 owampd.limits(5) 매뉴얼 페이지 (http://e2epi.internet2.edu/owamp/owampd.limits.man.html).

다음 예제는 각각의 호스트들로부터 오는 연결들을 분류하기 위해 어떻게 IP/넷마스크 할당을 사용할 수 있는지 보여준다.

```
# loopback
assign net ::/127 noc
assign net 127.0.0.1/32 noc
# abilene nmslan (observatory systems)
assign net 2001:464:0::/40 noc
assign net 198.32.10.1/23 noc
assign net 10.0.0.0/16 hostile
```

이 예제는 어떻게 loopback 인터페이스로부터 서버까지 어떻게 연결이 noc 제한계층과 연관된 제한을 배정되는지 보여준다. 추가적으로, nmslan 시스템은 같은 제한계층으로 배정되어 있다.

이 예제는 또한 주어진 서브넷으로부터의 온 모든 연결들이 인증되지 않으면 모두 거부당하는지 어떻게 확인할 수 있는지 알려준다(10.0.0.0/16를 보라). 그 *hostile* 제한계층은 allow_open_mode를 설정하지 않는다. 그래서, 열린 모드 통신들은 이 주소 영역으로부터 수락되지 않을 것이다. 그러나, 이 서브넷의 사용자들은 아직 사용자이름/AES 키 인증을 사용할 수 있다. (만약 아무런 통신이 주어진 서브넷에 요구되지 않는다면, 그 기능성은 방화벽 응용과 함께 보다 좋게 제공된다.)

넷 마스크 배정은 너무 강하게 믿어지면 안된다. Loopback은 정당하고 충분히 “local” 망 하지만 그외 모델을 확장하기 전에 많은 염려가 되어야 한다.

다음 예제는 주어진 사용자로부터 연결을 분류하기 위해 사용자이름 배정을 어떻게 사용할지 보여준다.

```
# network admins
        assign user joe root
        assign user jim root
        assign user bob root
```

```
# measurement geeks
        assign user boote noc
```

owampd 서버는 주어진 사용자가 어떤 이들을 얘기하는지 인증할 수 있을 필요가 있다. 이는 128-bit 공유키를 사용함 으로써 이루어진다. 키 연관으로의 사용자 이름은 아래와 같이 기술된 owampd.keys 파일을 사용하는 owampd에게 알려진다. 사용자는 owampd.limits 파일에 사용되기 위해 반드시 owampd.keys 파일 안에 포함되어 있어야 한다. 만약 사용자가 owampd.limits에 포함되어있고, owampd.keys에 포함되어 있지 않다면 그 owampd 프로세스는 시작을 거부할 것이다.

bwctld.keys 설정

owampd.keys 파일은 owampd가 사용자를 인증하도록 필요한 두 개의 동일성/AES 키들을 담아두기 위해 사용된다. 이 파일의 형식은 aespasswd(1) 매뉴얼 페이지에 기술되어 있다. owampd.keys 파일의 위치는 owampd의 c 옵션에 의해 제어된다.

owampd는 인증을 위해 대칭 AES 키들을 사용한다. 그러므로, owping 클라이언트는 AES 동작에 의해 인증을 위해서 정확히 동일한 AES 키를 통해 접근해야만 할 것이다. 대부분, 사용자는 단순히 처음에 만들어진 AES 키를 통과절차로만 알고 있을 것이다. 추가적으로, 그 키는 타협된 것이 아니라는 것을 단말 사용자와 시스템 관리자가 확신하는 것 또한 중요하다.

만약 owping 클라이언트가 나타난 AES 키와 ID를 사용해서 인증 할 수 있다면, owampd는 이 연결에 정책적 제한을 조사하기 위해 owampd.limits 파일에서 발견되는 지시사항을 사용할 것이다.

사용자 이름과 AES 키 규칙:

- 사용자 이름은 16 글자로 제한한다.
- AES 키는 128 bit 세션 키이다.
- AES 키는 key 파일에서 암호화 되지 않는다. 그리고 그를 보호하기 위해 유닉스 허용방안(permission)을 이용한다.
- AES 키를 발생시키기 위해 암호(passphrase)를 사용한다.
- 암호 발생 키를 key 파일에 추가시키기 위해 aespasswd를 사용한다.
- 클라이언트: 응용은 암호를 위해 사용자에게 촉구한다.

일반 유닉스 보호 방법에서는 특정 사용자 혹은 그룹 허용방안(permission)과 함께 데몬을 실행 시키려 할 것이다. 그때 그 허용방안은 key 파일을 읽을 수 있도록 허용한다. 하지만,

그에게 접근하는 사용하는 제한한다. 예제 key 파일은 다음과 같을 것이다.

joe a0167ac6101b360d2f4dd164abba2337
bob 2d2d2a31b3e2f4af54524423f43c2323
sam 2234c35d522a4d34aae43ef4323ffa52b

이것은 단순히 16진수로 표현된 128 bit 값이다. 지금까지 owampd.keys 파일을 생성하고 유지하는 방법을 알아온 것은 aespasswd 응용을 이용하기 위한 것이다.

aespasswd

이는 htpasswd (apache 웹서버)와 비슷하다. Key 파일에 추가될 ID를 구체화하고 암호를 위해 프롬프트 상태로 한다. 사용자는 128 bit 양을 기억하는데 쉽지 않기 때문에 용이하다. 이는 암호를 128-bit 16진수 키로 변환시키는데 사용된다. 보다 많은 정보를 원한다면, http://e2epi.internet2.edu/owamp/aespasswd.man.html을 보라. 이런 같은 응용은 OWAMP와 BWCTL을 위한 key 파일을 관리하기 위해 사용된다. 그리고 bwctl는 수정되는 파일을 세심히 관찰한다.

새로운 key 파일을 생성하기 위해 ‘ n’ 옵션을 사용한다:

% aespasswd n f bwctld.keys demo

추가적으로 사용자 이름은 ‘ n’ 을 통해 추가될 수 있다:

% aespasswd f bwctld.keys joe

보다 자세한 정보를 원한다면 다음을 참고한다.
http://e2epi.internet2.edu/owamp/owampd.keys.man.html,
http://e2epi.internet2.edu/owamp/aespasswd.man.html, 그리고
http://e2epi.internet2.edu/owamp/owampd.limits.man.html.

OWAMP를 이용하는 단방향 시간지연 측정기 실행하기

owampd 시작하기

시험하는 동안 전면으로 데몬을 가동한다.

% /ami/bin/owampd c /ami/etc/ Z

많은 명령어 라인 옵션들은 config 파일 파라메터를 겹쳐서 사용된다. 예를 들어, 데몬이 config 파일로부터 시작되지 않는다면 ‘ c’ 옵션은 항상 사용될 것이다. 더 많은 정보를 필요로 한다면, 다음을 참고한다. http://e2epi.internet2.edu/owamp/owampd.man.html.

테스트 (owping)

우선, owampd로부터 출력을 관찰하고 간단한 지역호스트 테스트를 시도한다. 이는 지역호스트에게 시간격차 이슈가 없기 때문에 좋은 테스트라 하겠다.

% /ami/bin/owping localhost

처음으로 지역호스트에게 테스트하는 것은 동작 설정을 검사할 수 있도록 도와준다. 방화벽 문제기반으로 호스트를 위해 테스트할 지역호스트 인터페이스 대신 인터페이스를 위한 IP 주소를 사용한다.

현재 호스트들중 하나를 테스트한다. (이 호스트는 단지 Network Performance Workshop 동한 사용 가능하도록 보장된다. 우리는 때때로 여기에서 새로운 것들을 시도하려 한다.)

% /ami/bin/owping nmsy-aami.abilene.ucaid.edu (Internet2 test host)

현재 또 다른 호스트를 테스트한다. 좋은 후보는 피어 망 중 하나 혹은 글로벌 PMP 리스트 중 뭔가 선택적으로 안 되는 것 중에서 새로운 배치가 될 수도 있다. (http://e2epi.internet2.edu/pipes/pmp/pmp-dir.html).

% /ami/bin/owping *otherhost*

보다 많은 정보를 위해, http://e2epi.internet2.edu/owamp/owping.man.html.

일단, 설치가 검점되면, rc 스크립트를 설치하도록 권장한다. 그는 호스트가 부팅되었을 때 자동적으로 owampd를 시작하도록 해준다. 일반적으로 데몬의 옵션중 Z는 사용하지 말아야한다. 하지만, 매우 좋은 디버깅 도구이다.

문제해결

다음은 가장 자주 일어나는 문제들이다.

1. No control connection (제어 연결 없음)

- 데몬 실행 안 할 경우
- 방화벽에 의해 제어 포트(4824)이 닫힐 경우

2. Control connection denied (제어 연결 거부)
   - 부적당한 설정 때문
   - 유효하지 않은 증명서 때문
3. Control connection works all test packet lost (제어 연결 성공 모든 패킷 손실)
   - 방화벽 UCP 포트의 범위를 열어주고, owampd.conf 내에서 열어진 포트들의 범위를 owampd가 사용하도록 테스트 포트도 열어준다.
   - 시계 옵셋(offset) 만약 두 시스템 사이에서 시계 옵셋이 크다면, 모든 *timeout* 패킷들이 손실되었다고 선언될 것이다. (*timeout* 정의를 위해 owping의 L 옵션을 보라.) 이를 테스트 하기 위해 L을 위한 큰 값을 사용한다.(시스템들 사이에서 가능한 시계 격차보다 큰 값)
4. Negative delay value for results
   - 시계 옵셋 가능하다면 시스템들 사이에 시간 격차를 알아보기 위해 NTP를 사용한다.