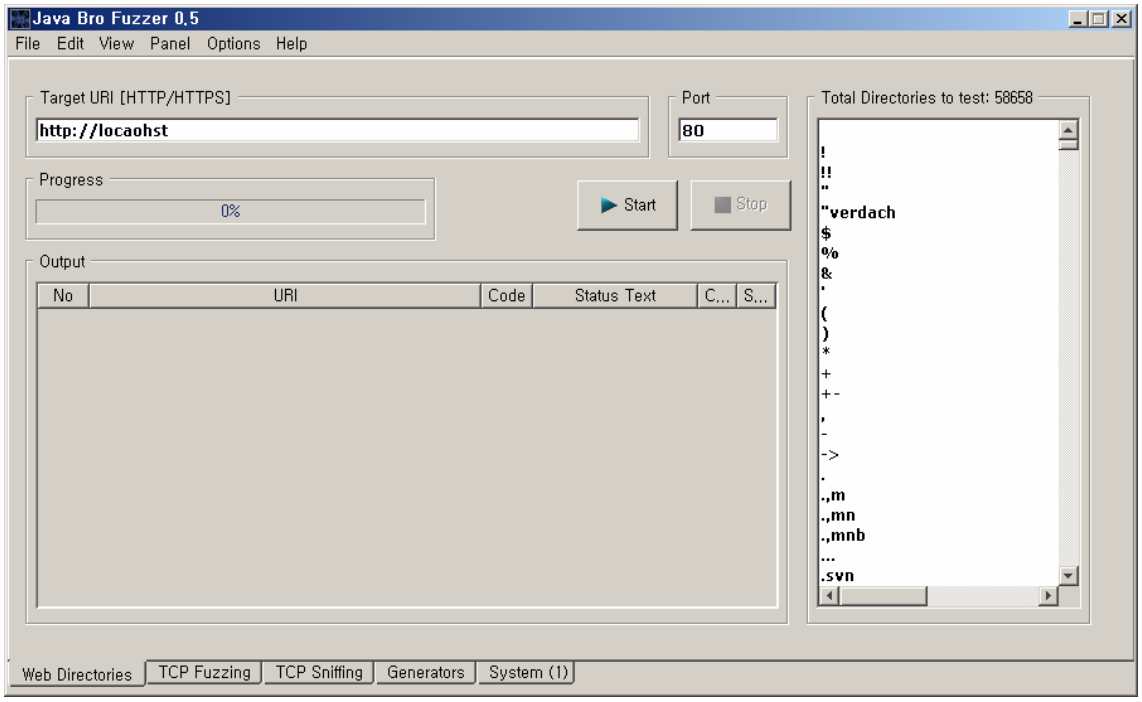
With Back|Trɛck 2

BT2에 포함되어 있는 툴에 대해 하나씩 사용해 보자. BT2가 아닌 windows 로 porting 된 툴이 있으면 windows 용 툴을 사용 하겠다.

1. JBROFUZZ

Fuzzer tool 이다. Spi에서 나온 상용 fuzzer tool 보다는 기능적인 면에서 다소 미흡하지만 사용하기가 쉽고 편리하다.



처음 실행하면 위와 같은 화면이 뜬다. GUI 환경이라 어려워 보이지 않고 또한 간단한 메뉴로 되어 있어 사용하기 매우 편리하다.

그럼 이 툴에 가지고 있는 몇가지 기능에 대해 알아보자.

아래 부분을 보면 5가지 tab이 있다

Web directories ➔ 웹 루트부터 디렉토리를 찾는 기능이다.

Tcp Fuzzing ➔ 이 툴의 핵심 기능으로 tcp 세션에 Fuzzing을 할 수 있는 기능 이다.

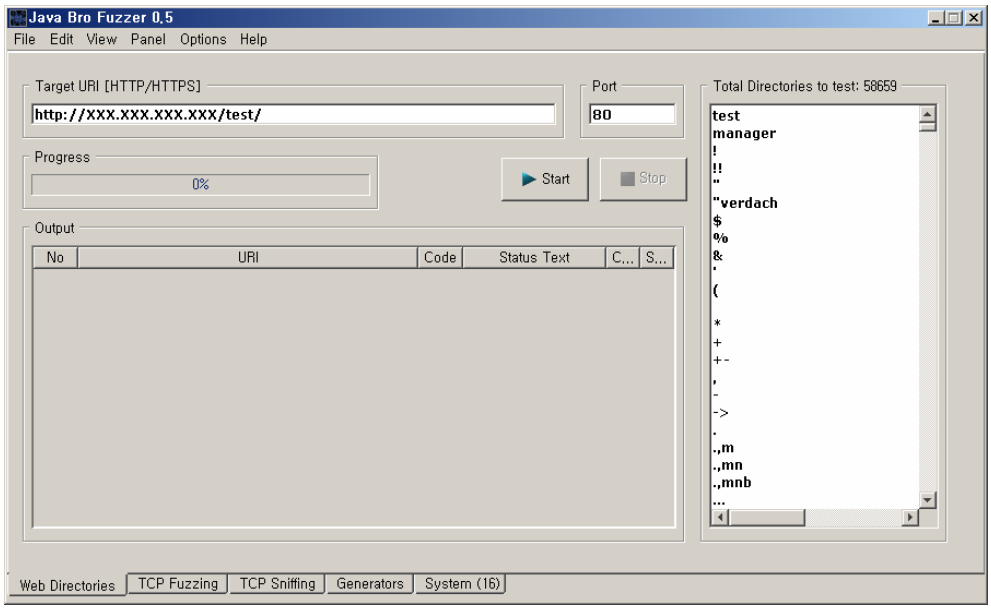
Tcp Sniffing ➔ 말 그대로 proxy 기능을 사용하여 스니핑을 통해 request//response 정보를 확인 할 수 있다.

Generators ➔ 각각의 Fuzzing 요소들의 실제 대입되는 값을 확인 할 수 있다.

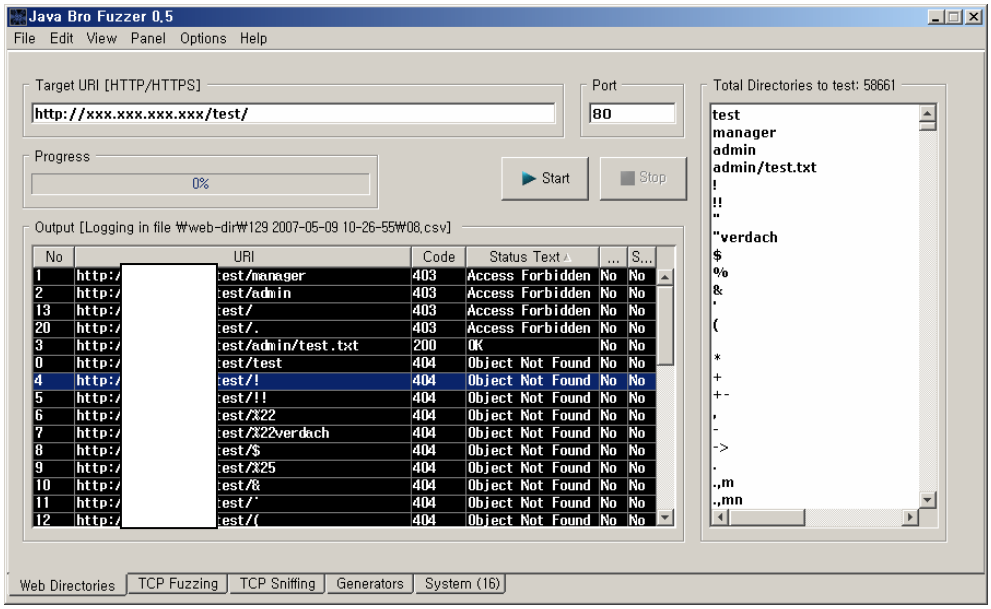
System ➔ jbrofuzz 툴에 대한 로그을 남기는 듯 하다.

(1) Web directories

해보자….. 아주 쉽다..



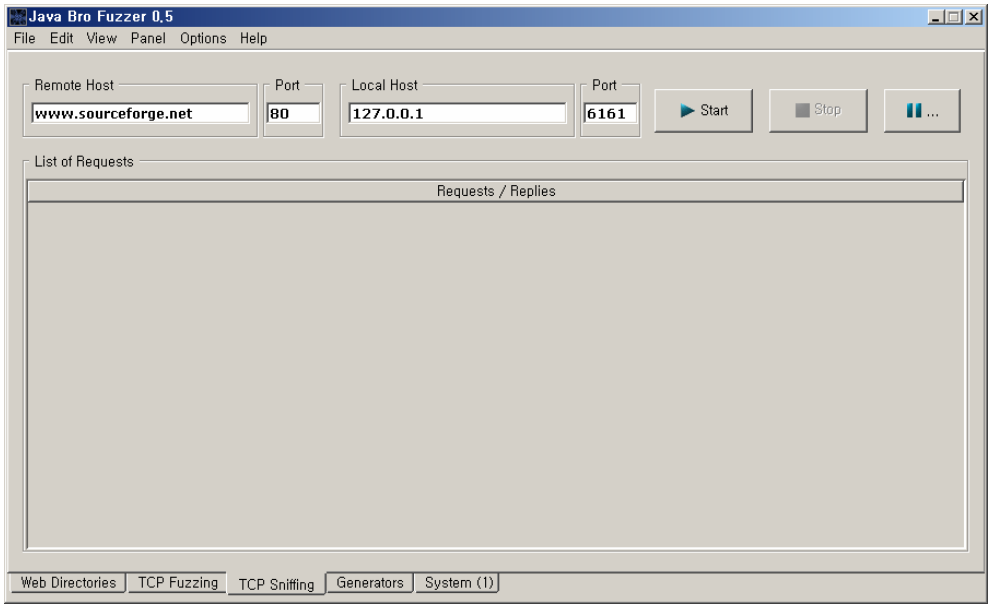
Target URI 부분에 써주면 된다. Start 버튼을 클릭하면 된다. 특별히 포함하고 싶은 디렉토리 명이 있으면 오른쪽 창에 입력하면 리스트에 추가 된다.



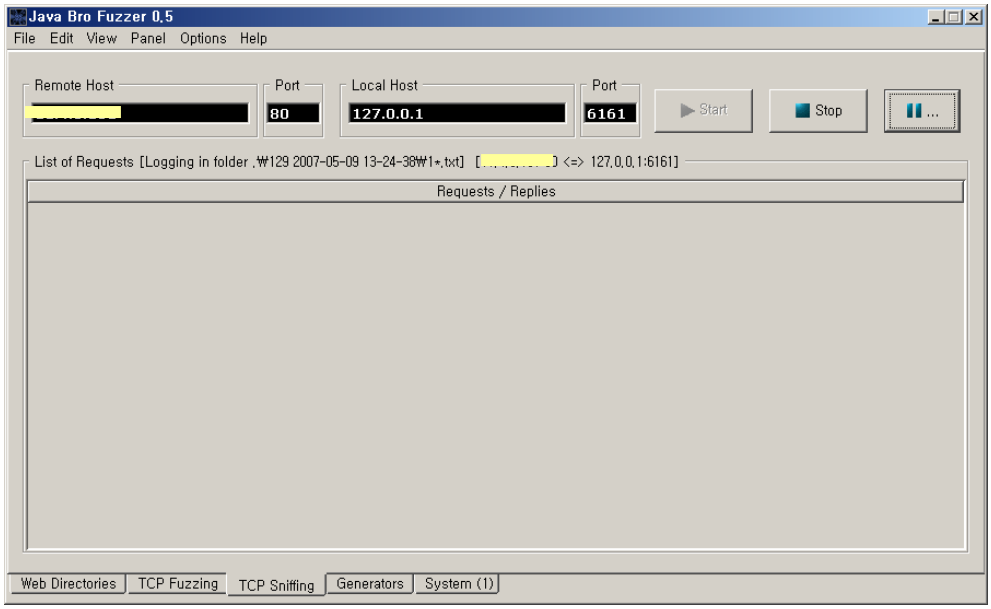
테스트 환경을 구축하여 실제로 테스트 한 화면 이다. 웹 Fuzzer tool의 특징인 반자동을 이용하면 더 편리하다. 예를 들면 디렉토리를 찾는 기능이지만 리스트에 파일까지 써주면 파일을 찾는다. 위 결과는 디렉토리가 존재하면 403 상태 코드를 파일이 존재하면 200 코드를 응답한 것을 쉽게 표로 보여 주고 있다. 만일 디렉토리 리스팅이 허용되어 있다면 200 코드를 디렉토리에서 응답한 것을 보여 줄것이다. 이를 통해 웹 디렉토리 중에 관리자 페이지와 같은 중요한 디렉토리들의 이름에 대한 점검을 할 수 있다.

(2) TCP Fuzzing

TCP Fuzzing 이 툴의 핵심 기능이다. 이 기능에 대해 보기전에 우선 tcp sniffing 에 대해 확인해 보도록 하자.

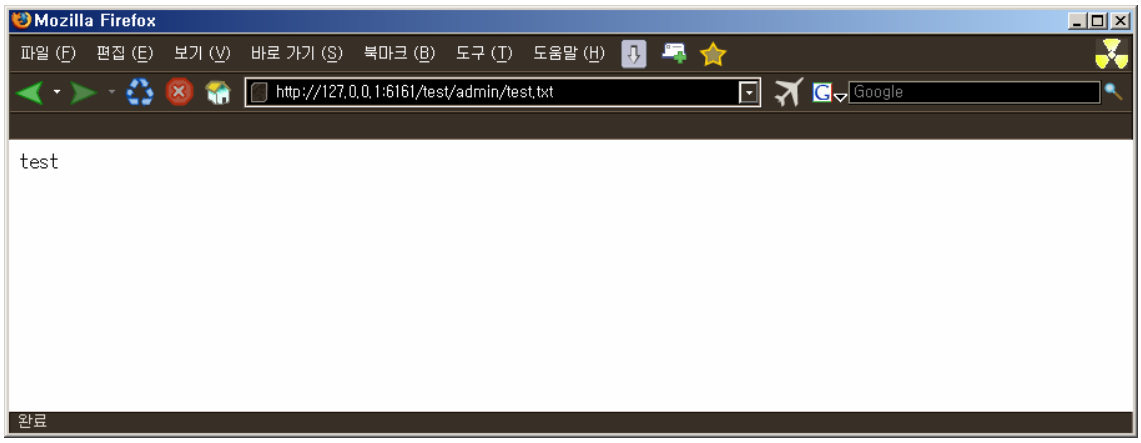


초기 화면은 위의 그림과 같다. 이 상태에서 start 버튼을 누르면 localhost의 6161 포트로 들어오는 컨넥션에 대해선 www.sourceforge.net 으로 포워딩 해준다. 이종의 proxy 라고 할 수 있다. 실제로 사용해 보자.

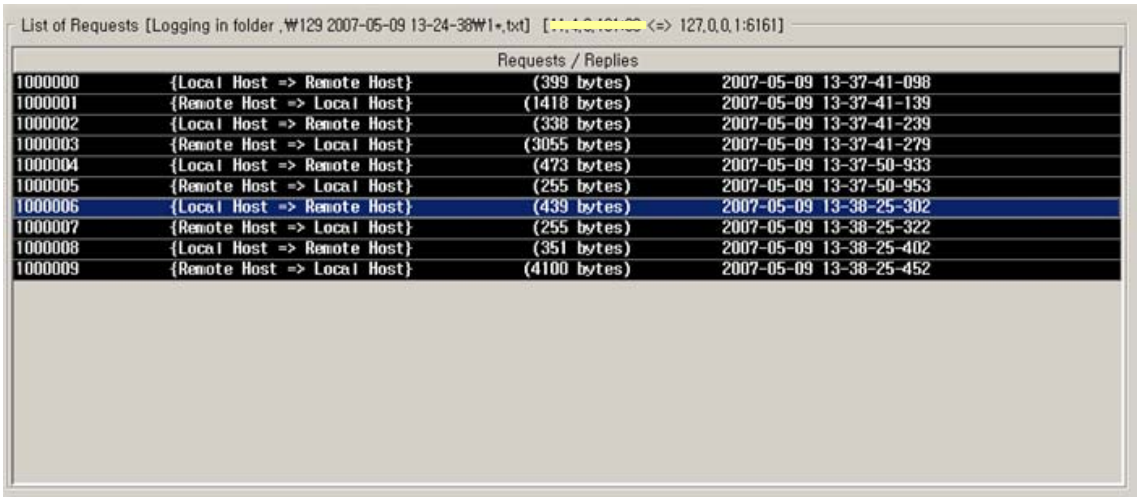


이제 웹 브라우져을 통해 6161로 접속 해보겠다. Start 버튼을 누르면 localhost에 6161포트가 열린다.

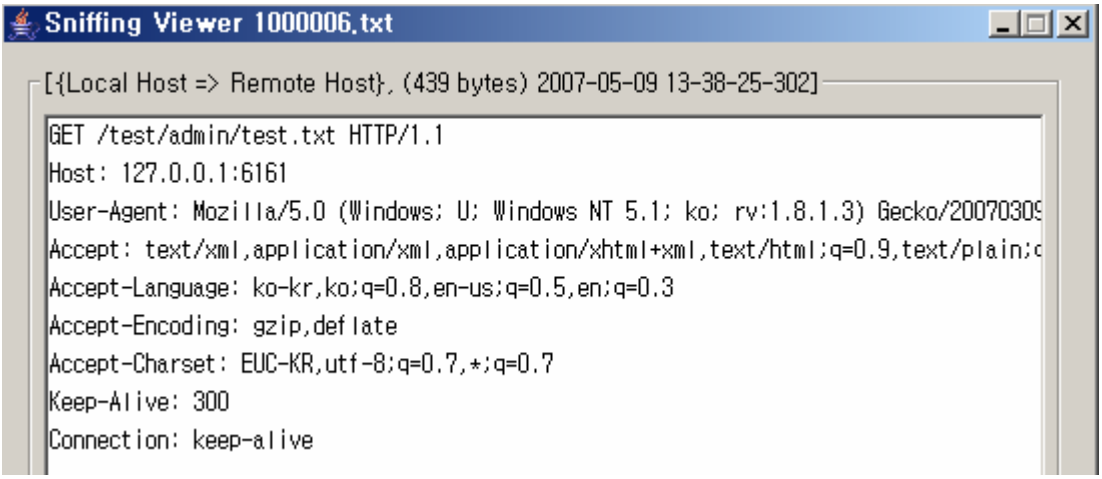
Remote host에 설정된 서버의 /test/admin/test.txt



Jbrofuzz을 보면 접속정보가 logging 되고 있다.

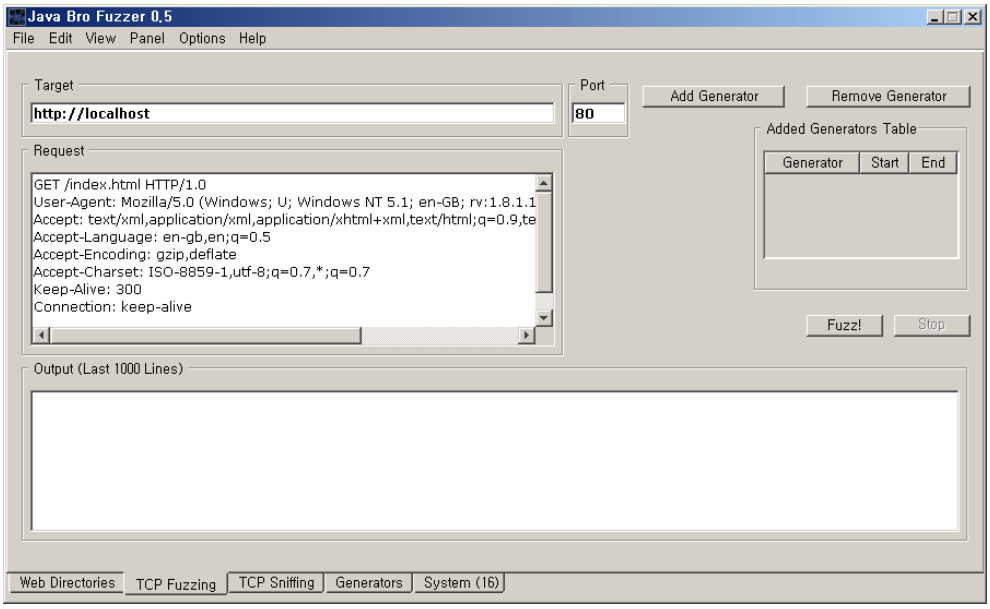


각각의 접속 정보에 대해 상세한 내용을 볼수 있다.

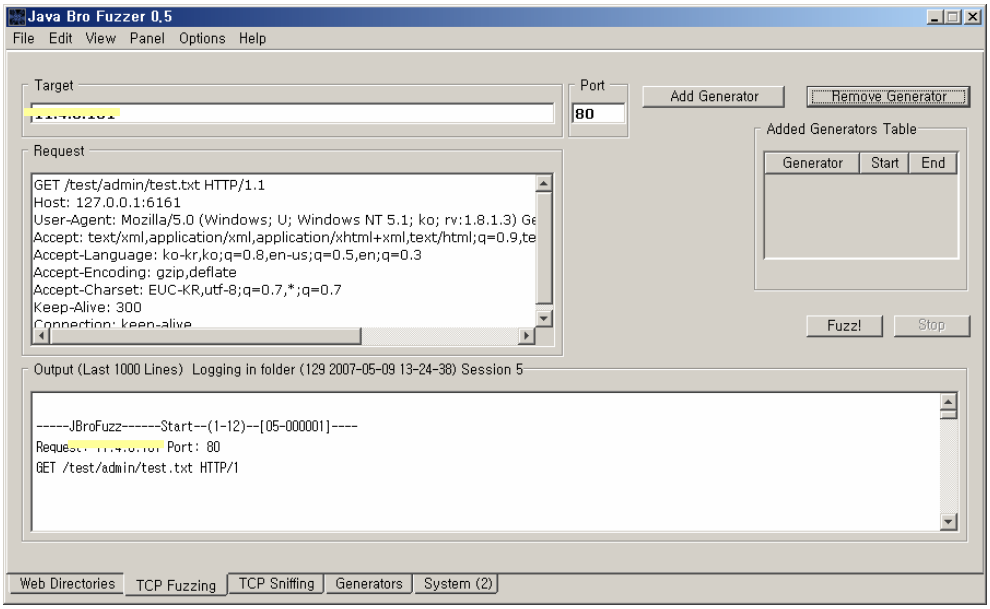


이 기능은 다른 여러가지 용도로 사용할 수 있으나 여기서는 tcp fuzzing에서 사용할 http 헤더로 사용하겠다.

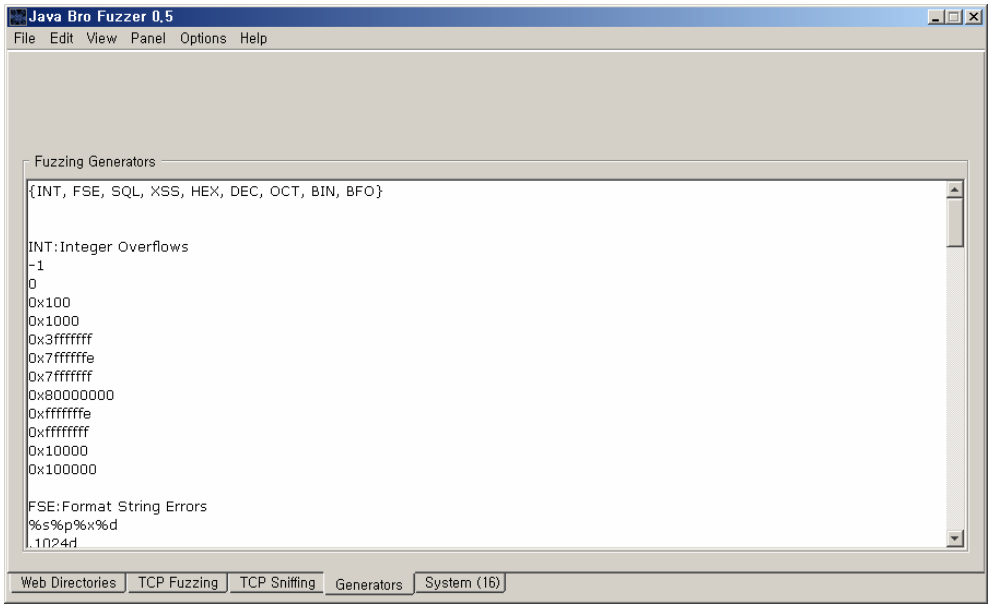
이제 fuzzing 기능을 사용해 보자.



처음 상태는 위의 그림과 같다. 이제 fuzzing 할 부분을 정하고 generator를 설정하기만 하면 된다. 이것 또한 아주 쉽게 할 수 있다. 우선 위에서 확인한 헤더 정보를 request를 복사에 넣는다. 그러나 가끔 Tcp sniffing 탭에서 복사해온 값은 문장 끝에 carriage return 없기 때문에 fuzzer 가 동작을 하지 않는 경우가 있다. 사실 fuzzer 가 동작하지 않는 것이 아니라 웹서버에서 응답을 하지 않는다. http 헤더가 엉망으로 들어오기 때문에 웹서버에서는 응답을 해주지 않는다. 이럴때는 헤더를 추가하기 위해서는 마지막에 carriage return(*%0d%0a*)을 하나씩 붙여 주면 잘 작동 된다.



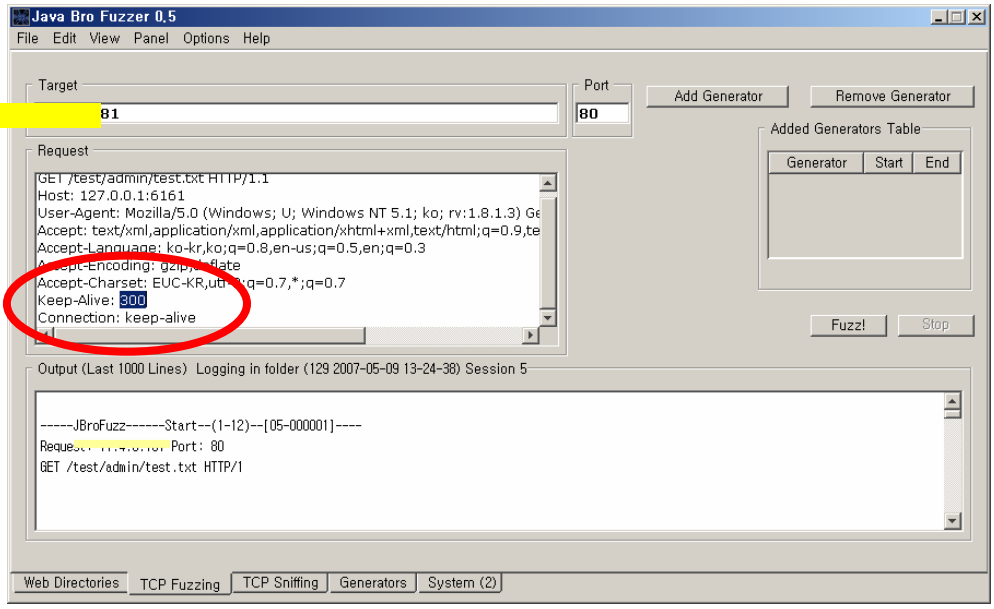
그 다음으로 generator 설정을 위해 generator에는 어떤 것 들이 있는지 확인 해보자. Generators 탭으로 이동하면 다음과 같은 그림이 나온다.



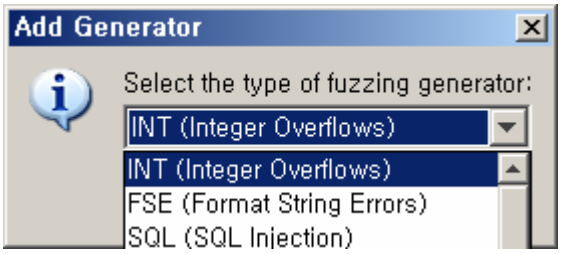
각각의 generator가 입력할 값들을 보여 주고 있다.

Generator 탭에 나오는 것은 jbrofuzz 폴더에 “jbrofuzz-generators” 파일에 있다. jbrofuzz-generators 파일을 수정하여 새로운 generator 도 만들 수 있다. 이 부분 또한 어렵지 않으며 파일에 자세히 설명되어 있으니 필요한 Generator 가 있다면 만들면 되겠다. 하지만 아쉬운 점은 generator에 random 기능을 이용해서 생성하는 부분이 없다.

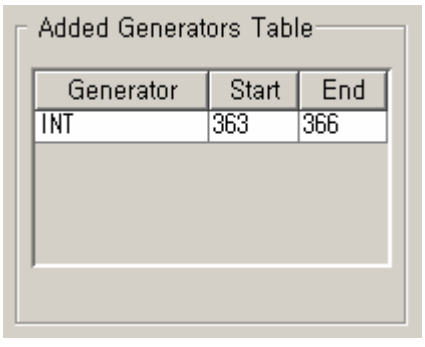
이제 실제로 generator를 설정해 보자.



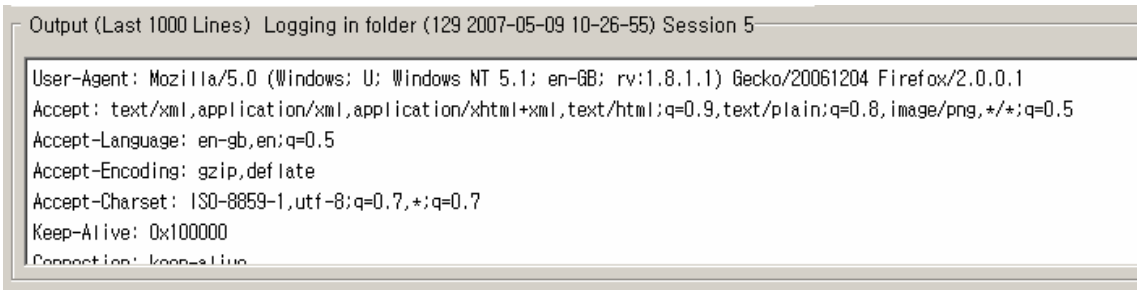
간단하다. Keep-Alive 헤더 필드의 값처럼 마우스로 드래그 후 Add Generator 버튼을 클릭해 주면 된다.



Integer overflows를 체크하는 항목을 선택한 후 확인 버튼을 누르면 설정 완료이다.



오른쪽에 있는 table 박스에 표시 된다. INT는 fuzzing generator 이름이다.
Start는 request 박스에서 글자수를 세어 우리가 정한 fuzzing 범위 중 첫번째 위치이고 end는 끝나는 위치이다. 이제 fuzz! 버튼을 누르면 fuzzing test 를 시작 한다. 결과 값은 output 박스에 출력이 되며 이 내용은 jbrofuzz 설치 폴더의 jbrofuzz₩fuzzing 안에 파일로 저장 된다.



Output 박스를 보면 keep-Alive 헤더 필드에 0x100000로 입력이 되었으며 서버로부터의 응답을 보고 취약점의 존재 유무를 확인 할 수 있다.

Web fuzzing 툴들은 여러가지로 쓸모있는 것들이 많은 것 같다.
Nc(netcat)를 두고 스위스의 군용 칼(맥가이버칼)이라고 부르기도 한다. Web fuzzing 툴도 그와 비슷하다는 생각이 든다. 기능이 단순하기에 응용할 수 있는 분야도 많으리라 생각된다.