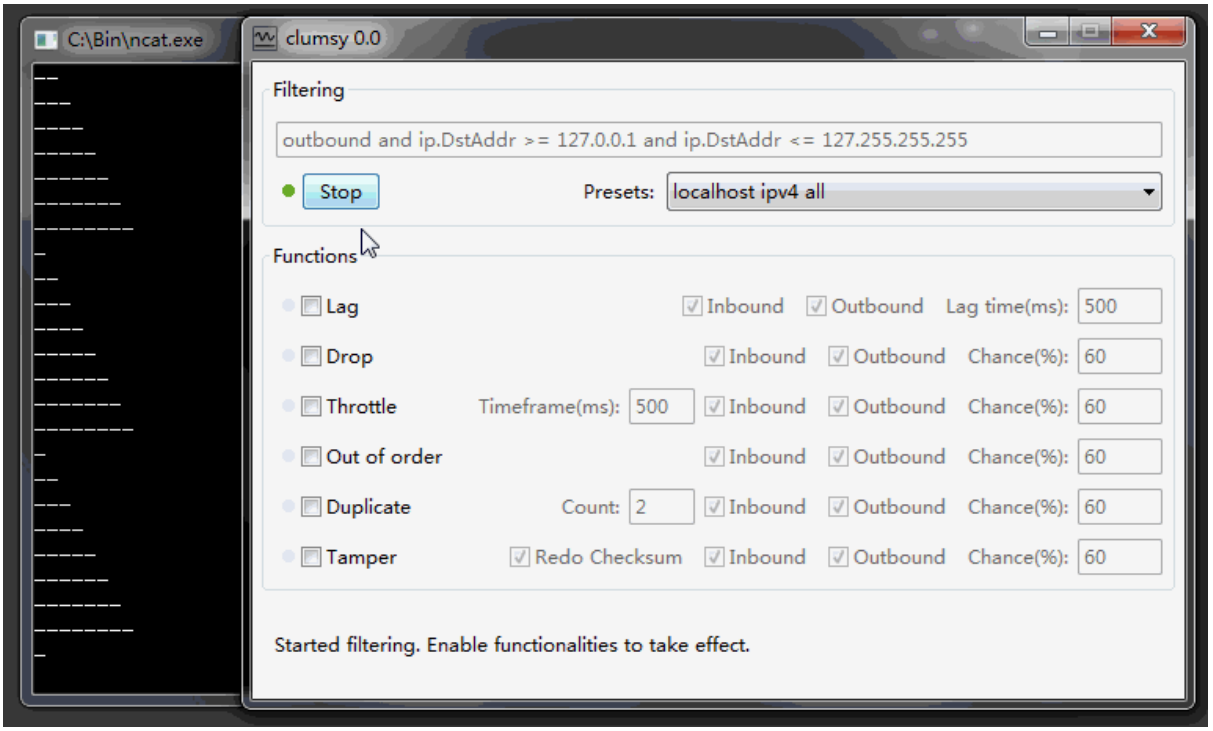
공식 사이트: http://jagt.github.io/clumsy/index.html
Windows만 지원. Windows Server 2008 이상 지원.
Windows OS에서 패킷 드랍율이나 지연을 만들어 주는 네트워크 상태 시뮬레이션 툴.

소개

WinDivert 라이브러리를 활용하여 clumsy는 네트워크 패킷을 정지시켜서, 이 패킷들을 캡쳐하고, 필요에 따라서 패킷을 지연시키거나, 드랍 시키거나, 개조하여 패킷을보낸다. 이상이 있는 네트워크에 관련된 기묘한 버그를 추적할 때나 접속 불량 애플리케이션을 평가할 때 등에 유용하다.

- 설치 불필요.
- 프록시 설정이나 애플리케이션 코드 변경을 하지 않는다.
- 시스템 전체의 네트워크 캡쳐는 어떤 애플리케이션에서도 동작하는 것을 의미한다.
- HTTP뿐만이 아닌 TCP/IP를 베이스로한 다양한 프로토콜을 지원하고 있다.
- 오프라인(로컬 호스트에서 로컬 호스트로 접속 등)에도 동작한다.
- 네트워크가 어느 정도 나쁜지를 인터렉티브하게 제어하고, 무엇이 일어나고 있는지를 전달하기에 충분한 시각적 피드백이 있다.



상세

clumsy는 주어진 필터에 의해 어떤 패킷을 캡처할 것인지를 선택한다. clumsy가 시작되면 필터에 기반한 패킷만 캡처하고 다른 패킷은 그대로 유지된다.

패킷이 캡처된 후, 당시는 예상하는 네트워크 상태를 악화시키기 위해 제공되는 기능을 활성화할 수 있다.

1. 래그(Lag): 네트워크의 지연을 에뮬레이트 하기 위해 짧은 시간 동안 패킷을 유지한다.
2. 드롭(Drop): 패킷을 무작위로 버린다.
3. Throttle: 트래픽을 특정 시간대에 차단한 후, 일괄 전송한다.
4. Duplicate: 복제된 패킷을 원래 패킷 직후에 전송한다.
5. Out of order: 패킷의 순서를 변경한다.
6. Tamper: 패킷 컨텐츠의 비트를 넛지한다.

요즘에는 모두가 고속 광대역 인터넷 연결을 가지고 있는 것처럼 보이지만, 네트워크 전송이 항상 신뢰할 수 있는 것은 아니라는 사실을 직면하는 것이 중요하다. 중복된 UDP 패킷이 응용 프로그램을 충돌시키는 것을 피하기를 원한다. 이를 적절하게 처리하려면 일반적으로 프로젝트에 코드를 추가해야 하지만 항상 쉽지 않으며 가능하지 않다. clumsy가 바쁜 개발자들에게 쉽고 고통스럽지 않은(최적은 아니지만) 옵션을 제공해주기를 바란다.

프로젝트의 리포지토리는 github에 있다.
실제로 사용하기 전에 반드시 매뉴얼을 읽고 기능이나 한계를 알아두자.

제한 사항

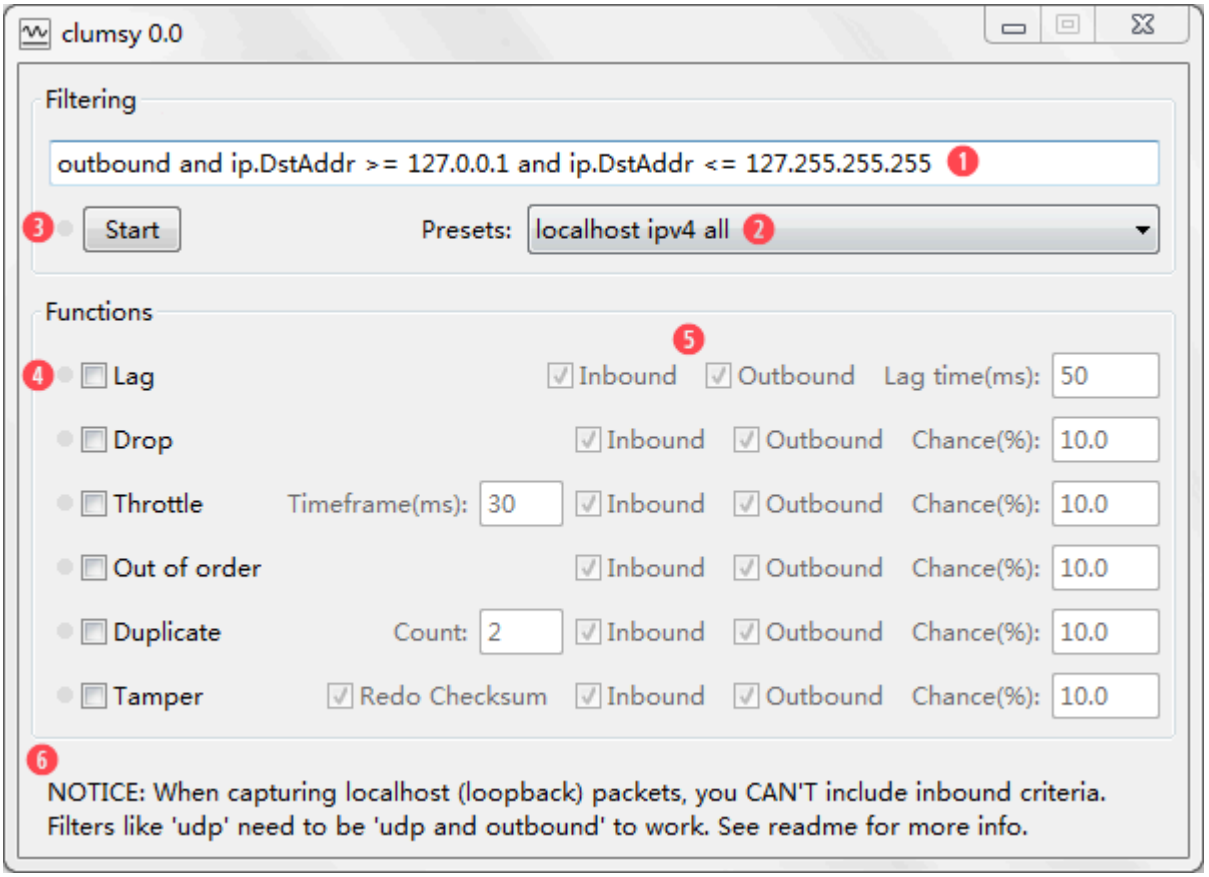
1. 루프백의 인바운드 패킷을 캡쳐하거나 재 인젝트 할 수는 없다.
컴퓨터에서 자기 자신에게 패킷을 보낼 때 이것이 인바운드인지 아웃바운드인지 구별하기 어렵다. 실제로 기초가 되고 있는 Windows Filtering Platform은 루프백 패킷을 모두 아웃바운드로 분류하고 있다.생각해보면 루프백 패킷에서 처리하고 있을 때 필터에 “인바운드”를 넣는 것은 불가능하다.
라우터에 따라서 할당된 인트라넷 IP 처럼 컴퓨터에 127.0.0.1 이외의 IP가 조잰하는 가능성이 있다는 것을 알아 두는 것은 중요하다.이들도 루프백 패킷로 본다.

2. 루프백 패킷은 2회 캡쳐 할 수 없다.

3. 인바운드 패킷의 캡쳐가 언제나 동작하지 않는다.
앞서 언급한 바와 같이 루프백한 인바운드 패킷을 다시 인젝트 할 수 없다. 문제는 어떤 경우에는 대상 IP가 자신의 컴퓨터가 아니더라도 일부 패킷이 인바운드 패킷으로 분류 될 수 있다는 것이다. 이것은 루프백 이외의 패킷에만 영향을 미친다. 로컬호스트에서만 작동하는 경우에는 문제가 없다. 향후 출시의 목표는 이 원인을 진단하고 해결책을 제공하는 것이다.

4. 프로세스에 기초한 필터링이 불가능하다.

사용 방법



1. 필터 입력. 자세한 구문은 다음 절에서 설명한다. 그러나 대부분의 프로그래밍 언어의 bool 공식과 유사하다. 패킷은 이 필터를 기반으로 캡처한다.

2. 사전 설정. 내장된 사전 설정 목록을 준비하고, 필터가 어떤 것이 될지 눈으로 볼 수 있다. 그리고 만약 여러분이 엉뚱한 것에 익숙해지면, 실행 파일에 동봉된 config.txt에 자체 필터를 추가할 수 있다.

3. 컨트롤 버튼.이를 클릭하면 캡처를 시작한다.
일부 장면에서는 실패할 수 있다. 예를 들어 필터의 구문이 잘못된 경우. 때로는 다른 오류로 실패할 수도 있다. 자세한 내용은 FAQ를 참조한다.
시작하면 텍스트가 "Stop"으로 변경된다. 왼쪽에 작은 아이콘이 있다. 패킷을 캡처하면 녹색이 된다. 실행 거부로 실패하면 아이콘이 빨간색이 된다. 이러한 일이 일어나는 것은 위의 제한이 있는 경우가 대부분이다. 그에 따라 필터를 변경해야 한다.

4. 기능 제어. 체크박스에 체크를 넣으면 기능이 활성화된다.제어의 왼쪽에도 아이콘이 있다.이 기능이 활성화 되면 녹색이 된다. 이 체크 박스에 체크를 넣으면 처리 도중 언제라도 on/off 전환이 가능하다.

5. 매개변수 제어. 각 기능에는 조정할 수 있는 추가 제어가 있다. 가장 일반적인 것은 아래와 같다.
- 인바운드/아웃바운드: 인바운드/아웃바운드 패킷을 처리할지 여부. 이들은 필터와 독립적이며 언제든지 변경할 수 있다.

- 확률: 이 함수가 처리할 확률. 분명히 항상 패킷을 떨어뜨리고 싶지는 않을 것이기 때문에, 확률은 합당한 값으로 설정하는 것이 좋을 것이다.

6. 상태: 현재 상태에 대한 유익한 문자 메시지를 표시한다.

필터 구문

필터링된 텍스트는 WinDivert에 직접 피드된다. 구문은 여기서 자세하게 설명한다. 어떤 언어라도 프로그래밍한 적이 있다면, 이것은 if 조건에 넣은 것과 매우 유사하다.
논리 공식을 구성하기 위해 and, or, not 및 괄호를 사용할 수 있으며 =, =, =, , ',' 와 같은 조작도 제공한다.WinDivert 문서에서 사용할 수 있는 필드 목록을 아래에 첨부한다. 또 예를 들면 사전 설정을 참조할 수도 있다.

outbound	Is outbound?
inbound	Is inbound?
ifIdx	Interface index
subIfIdx	Sub-interface index
ip	Is IPv4?
ipv6	Is IPv6?
icmp	Is ICMP?
icmpv6	Is ICMPv6?
tcp	Is TCP?
udp	Is UDP?
ip.*	IPv4 fields (see DIVERT_IPHDR)
ipv6.*	IPv6 fields (see DIVERT_IPV6HDR)
icmp.*	ICMP fields (see DIVERT_ICMPHDR)
icmpv6.*	ICMPV6 fields (see DIVERT_ICMPV6HDR)
tcp.*	TCP fields (see DIVERT_TCPHDR)
tcp.PayloadLength	The TCP payload length

udp.*	UDP fields (see DIVERT_UDPHDR)
udp.PayloadLength	The UDP payload length

사용 예

특정 포트로 보내는 패킷을 딜레이 시키기.

리모트에서 32451로 보내는 패킷은 2000밀리세컨드로 딜레이 되서 받는다.