

SNMP 트랩 패턴 검사기 구현

제출 기한: 12월 2일 밤 11시 59분

제출 방법: xeraph@nchovy.com 메일 전송

(바이너리 포함 시 자동 필터링 되므로 압축 후 확장자에 .rename을 붙여서 보내기 바랍니다.)

제출 파일: 주석(영문)이 잘 달린 소스 코드, 실행 가능한 바이너리

(자바의 경우 JAR로 패키징하고 Main-Class를 지정해서 java -jar로 바로 실행할 수 있도록 패키징 하시기 바랍니다. Maven이나 Ant로 개발 환경 셋팅한 경우는 추가 점수 있음.)

프로그래밍 언어: Java, C, C++, Python 중 택일

(단, 윈도우 개발 환경이어야 하며, C/C++의 경우 Visual Studio 2008로 빌드 가능하도록 솔루션 파일과 프로젝트 파일을 같이 보내주시기 바랍니다.)

과제 내용:

1. 콘솔 프로그램 실행 매개변수로 파일 경로를 입력받고, 아래 예시와 같이 줄 단위로 패턴 식별자(3만 이하의 양수)와 16진수 바이트 패턴을 읽어서 Aho-Corasick 유한상태기계 (FSM)를 구축
1 6E 63 68 6F 76 79
10 48 45
5 48 45 52 53
7 48 49 53
2. UDP 162번 포트로 임의의 SNMP Trap v1 패킷을 수신
3. Variable Bindings 부분을 OID(키)와 값으로 파싱하고, 그 중에서 모든 OctetString 타입의 값에 대하여 Aho-Corasick 알고리즘으로 바이트 패턴 매칭을 수행
 - A. [RFC1155](#): Structure and identification of management information for TCP/IP-based internets 참조
 - B. [X.690](#): ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)
4. 일치하는 모든 패턴의 식별 번호를 OID와 함께 출력
 - A. 위의 예시와 같은 패턴 설정을 사용했을 때 .1.3.6.1.4.1.2854 키에 대한 OctetString 값이 SHERS인 경우 아래 포맷으로 출력
.1.3.6.1.4.1.2854 -> 10 5
 - B. 테스트 시 [trapgen](#) 사용

기타 궁금한 사항이 있으면 메일로 질문 바랍니다.