시험 윤리 서약서

소속	
학번	
성명	

위 본인은 아래와 같이 부정한 방법으로 시험에 응시하거나 결과물을 제출하는 경우 시험 성적이 취소 또는 조정될 수 있으며, 필요한 경우 징계 등 엄중하게 처벌받을 수 있음에 동의합니다.

1. 대리시험을 의뢰하거나 대리로 시험에 응시하는 행위
2. 타인의 답안을 복사 또는 표절하는 행위
3. 2인 이상이 시험 내용을 공유하거나 상의하는 행위
4. 기타 시험 관련 부정으로 간주될 수 있는 행위

20__년 __월 __일

서약자 _____ (서명)

배경 설명

1. 서비스 시작

```
vagrant@vagrant-env:~/project/service$ ./server.sh start
```

2. 서비스 설명

각각의 서비스는 사용자의 요청에 따라 특정 리소스를 접근, 그 결과를 사용자에게 제공한다.

서비스 이름	서비스 주소
service_md5	127.0.0.1:8001
service_sha1	127.0.0.1:8002
service_sha256	127.0.0.1:8003
service_sha512	127.0.0.1:8004

참고로, 위 서비스들은 실제 서비스와 리소스들이 모두 구현되어 있다는 가정하에, 서비스에 대한 리소스 요청과 응답을 단순화한 것으로, 내용 자체는 무시해도 좋다.

3. 서비스 이용 방법

사용자는 특정 서비스를 통해서 해당 서비스에서 제공하는 리소스 중 하나에 접근한다.

```
vagrant@vagrant-env:~/project/service$ ./client.py
Usage: ./client.py {md5|sha1|sha256|sha512} [resource]
```

4. 서비스 이용 예시

```
vagrant@vagrant-env:~/project/service$ ./client.py md5 test
"resource":"test";"content":"098f6bcd4621d373cade4e832627b4f6";
```

```
vagrant@vagrant-env:~/project/service$ ./client.py sha1 sw
"resource":"sw";"content":"8d4c780fcfdc41841e5070f4c43da8958ba6aec0";
```

```
vagrant@vagrant-env:~/project/service$ ./client.py sha256 app
"resource":"app";"content":"a172cedcae47474b615c54d510a5d84a8dea3032e958587430b413538be3f333";
```

```
vagrant@vagrant-env:~/project/service$ ./client.py sha512 midterm
"resource":"midterm";"content":"34dab65b9ace351fe5e8cec66436753504505d7e25c2aa622c51cc47ee8c543259a3899cbfcd50288dc1165b1f2239ba3f234e182783fd5a9ecd61f53290f66a";
```

중간 실습 시험

1. 프로그램 설명

- 사용자들이 서비스들을 통해 특정 리소스들에 접근할 때, 본 시스템은 중간에서 접근하는 리소스들에 대한 통계 정보를 수집한다.

- 이와 함께, 통계 수집 과정에서 특정 룰에 매칭되는 리소스 접근이 있는 경우 경고 메시지를 띄운다.

2. 구현 사항

- Makefile

  make 명령어를 통해 시스템을 컴파일 한다.

- Command-line arguments (context.c / context.h)

짧은 옵션	긴 옵션	[입력]
-i	--interface	[인터페이스 이름] / 필수, 정의되지 않았다면 종료
-r	--rule	[룰 파일 이름] / 필수, 정의되지 않았다면 종료
-d	--debug	파라미터 없음

```
vagrant@vagrant-env:~/project$ sudo ./stats_monitor -i lo
Missing option: -r
Failed to initialize context
vagrant@vagrant-env:~/project$ sudo ./stats_monitor -r rules.txt
Missing option: -i
Failed to initialize context
```

- File I/O (rule.c / rule.h)

  룰 파일이 존재하지 않을 경우, 에러 메시지와 함께 프로그램을 종료한다.

```
agrant@vagrant-env:~/project$ sudo ./stats_monitor -i lo -r empty
Initialized rules
Failed to open 'empty'
Failed to load rules
```

  룰 파일을 읽어 필드별로 파싱한 후 struct 형태로 변환, linked-list 형태의 pointer array로 유지한다.

  rule → service, resource, message, *next
  service → {“md5” | “sha1” | “sha256” | “sha512”}
  위 4 경우 외 다른 값이 들어오면 에러 메시지와 함께 프로그램을 종료한다.

  모든 룰을 처리한 후 전체 룰에 대해서 출력한다.

== rules.txt ==

```
# service:"service name"; resource:"resource name"; message:"message";

service: "md5"; resource: "dku"; message: "someone accessed 'dku' in md5";
service: "sha1"; resource: "sw"; message: "someone accessed 'sw' in sha1";
service: "sha256"; resource: "app"; message: "someone accessed 'app' in sha256";
service: "sha512"; resource: "midterm"; message: "someone … 'midterm' in sha512";
```

== 실행 예시 ==

```
vagrant@vagrant-env:~/project$ sudo ./stats_monitor -i lo -r rules.txt
```

```
vagrant@vagrant-env:~/project$ sudo ./stats_monitor -i lo -r rules.txt
Initialized rules
Rule #01
  service: sha512
  resource: midterm
  message: someone accessed 'midterm' in sha512
Rule #02
  service: sha256
  resource: app
  message: someone accessed 'app' in sha256
Rule #03
  service: sha1
  resource: sw
  message: someone accessed 'sw' in sha1
Rule #04
  service: md5
  resource: dku
  message: someone accessed 'dku' in md5
3 rules are loaded
```

- Libpcap (capture.c / capture.h)

  입력 받은 인터페이스 이름이 시스템 네트워크 인터페이스 목록에 없다면,
  에러 메시지와 함께 프로그램을 종료한다.

```
vagrant@vagrant-env:~/project$ sudo ./stats_monitor -i xxx -r rules.txt
Initialized rules
Rule #01
  service: sha512
  resource: midterm
…
  resource: dku
  message: someone accessed 'dku' in md5
3 rules are loaded
Failed to find 'xxx'
Failed to capture packets
```

  옵션으로 부터 전달받은 인터페이스를 통해 패킷을 수집한다.

- Protocol parsing (decode.c / decode.h)

  수집된 패킷 중 프로토콜 파싱을 통해 서비스 관련 패킷들만 추출한다.
  -d | --debug 옵션이 켜지면 서비스로의 리소스 요청 및 응답에 대한 패킷들을 출력한다.

```
vagrant@vagrant-env:~/project$ sudo ./stats_monitor -i lo -r rules.txt -d
Initialized rules
Rule #01
  service: sha512
  resource: midterm
…
  resource: dku
  message: someone accessed 'dku' in md5
3 rules are loaded
Start to capture packets
127.0.0.1:50910 -> 127.0.0.1:8002
GET /abc HTTP/1.1..Host: 127.0.0.1:8002..User-Agent:
python-requests/2.22.0..Accept-Encoding: gzip, deflate..Accept: */*..Connection:
keep-alive....
127.0.0.1:8002 -> 127.0.0.1:50910
HTTP/1.1 200 OK..Server: Werkzeug/2.2.2 Python/3.8.10..Date: Thu, 13 Oct 2022
20:33:13 GMT..Content-Type: text/html; charset=utf-8..Content-Length: 70..Connection:
close....
127.0.0.1:8002 -> 127.0.0.1:50910
"resource":"abc","content":"a9993e364706816aba3e25717850c26c9cd0d89d";
```

- String match (match.c / match.h)

  수집된 패킷 중 응답에 해당하는 패킷 내용에 대해서 매칭된 룰이 있는 경우,
  (src ip, dst ip, 접근한 서비스명, 매칭된 룰 정보) 를 출력한다.

```
vagrant@vagrant-env:~/project$ sudo ./stats_monitor -i lo -r rules.txt
Initialized rules
Rule #01
  service: sha512
  resource: midterm
…
  resource: dku
  message: someone accessed 'dku' in md5
3 rules are loaded
Start to capture packets
127.0.0.1:52972 -> 127.0.0.1:8001 (message: someone accessed 'dku' in md5, service:
md5, resource: dku, content: 9c95bb2bb0b8f1c011ed0c2317e5069e)
127.0.0.1:43836 -> 127.0.0.1:8003 (message: someone accessed 'app' in sha256,
service: sha256, resource: app, content:
a172cedcae47474b615c54d510a5d84a8dea3032e958587430b413538be3f333)
127.0.0.1:57014 -> 127.0.0.1:8002 (message: someone accessed 'sw' in sha1, service:
sha1, resource: sw, content: 8d4c780fcfdc41841e5070f4c43da8958ba6aec0)
```

- Pointer array (stats.c / stats.h)

  수집된 패킷 중 응답에 해당하는 패킷 내용 (resource, context) 에 대해서 서비스 별 pointer array에 linked-list 형태로 entry (service, resource, context, counter=1)를 추가한다.

  이미 추가한 entry가 중복해서 나타난다면 카운터를 증가한다.

- signal handler (main.c)

  Ctrl+C가 입력되면, 현재까지 수집된 통계정보를 서비스와 요청별로 출력한다.

```
…
127.0.0.1:47944 -> 127.0.0.1:8001 (message: someone accessed 'dku' in md5, service:
md5, resource: dku, content: 9c95bb2bb0b8f1c011ed0c2317e5069e)
127.0.0.1:44538 -> 127.0.0.1:8002 (message: someone accessed 'sw' in sha1, service:
sha1, resource: sw, content: 8d4c780fcfdc41841e5070f4c43da8958ba6aec0)
127.0.0.1:44540 -> 127.0.0.1:8002 (message: someone accessed 'sw' in sha1, service:
sha1, resource: sw, content: 8d4c780fcfdc41841e5070f4c43da8958ba6aec0)
127.0.0.1:59228 -> 127.0.0.1:8003 (message: someone accessed 'app' in sha256,
service: sha256, resource: app, content:
a172cedcae47474b615c54d510a5d84a8dea3032e958587430b413538be3f333)
^CService: md5
  [#1]
  Resource: dku
  Content: 9c95bb2bb0b8f1c011ed0c2317e5069e
  Count: 3
Service: sha1
  [#1]
  Resource: sw
  Content: 8d4c780fcfdc41841e5070f4c43da8958ba6aec0
  Count: 2
Service: sha256
  [#1]
  Resource: app
  Content: a172cedcae47474b615c54d510a5d84a8dea3032e958587430b413538be3f333
  Count: 1
Service: sha512
Destroyed stats
Destroyed the rules
```

  == 실행한 명령어들 ==

```
vagrant@vagrant-env:~/project/service$ ./client.py md5 dku
"resource":"dku","content":"9c95bb2bb0b8f1c011ed0c2317e5069e";
vagrant@vagrant-env:~/project/service$ ./client.py md5 dku
"resource":"dku","content":"9c95bb2bb0b8f1c011ed0c2317e5069e";
vagrant@vagrant-env:~/project/service$ ./client.py md5 dku
"resource":"dku","content":"9c95bb2bb0b8f1c011ed0c2317e5069e";
vagrant@vagrant-env:~/project/service$ ./client.py sha1 sw
"resource":"sw","content":"8d4c780fcfdc41841e5070f4c43da8958ba6aec0";
vagrant@vagrant-env:~/project/service$ ./client.py sha1 sw
"resource":"sw","content":"8d4c780fcfdc41841e5070f4c43da8958ba6aec0";
vagrant@vagrant-env:~/project/service$ ./client.py sha256 app
"resource":"app","content":"a172cedcae47474b615c54d510a5d84a8dea3032e958…
```

3. 구현 완료

다음 항목 중 구현이 완료된 항목에 대해서 체크한다.

- [05점] Makefile [ ]
- [05점] Command-line arguments [ ]
- [15점] File I/O [ ]
- [15점] Libpcap [ ]
- [15점] Protocol parsing [ ]
- [15점] String match [ ]
- [15점] Pointer array [ ]
- [15점] Signal handler [ ]

4. 제출 방법

- project 디렉토리를 압축한 후 파일 이름을 학번_이름.zip으로 변경한다.
- 해당 압축파일을 이메일로 제출한다.

  이메일 제목: [2022-02 고급프로그래밍실습] 학번 이름
  이메일 주소: jaehyun.nam@dankook.ac.kr

- 이메일 제출 확인 후 퇴실하기 바랍니다.