**2022년도 졸업프로젝트[Back-end]**

**팀이름: 황제펭귄**

주제: TLS를 구현을 통한 홈페이지 제작 및 보안 채널 로그인 기능 구현

|  |  |  |  |
| --- | --- | --- | --- |
| **작성자** | 이유민 | **작성일자** | 2022.10.06 |

* **금일 진행 상황**

|  |  |
| --- | --- |
| **진행 상황** | **비고** |
| **Openssl을 이용한 3단 체인 인증서 제작(Full Certificate Chain)**   1. **목차** 2. **인증서 구성.** 3. **RootCA .crt(certification) 제작하기.** 4. **SubCA .crt(certification) 제작하기.** 5. **Server .crt(certification) 제작하기..** 6. **체인구조 완성하기.** 7. **인증서 구성**   **-결과사진**     1. Text     Description automatically generated**RootCA .crt(certification) 제작하기**   **(1)RootCA의 PrivateKey와 .crt파일 생성**  **openssl req -x509 -newkey rsa:4096 -keyout EmperorPenguin\_key.key -out EmperorPenguin\_crt.crt -sha256 -days 3650**  -RSA 4096비트 키를 생성해서 sha245으로 암호화,유효기간이 3650일인 인증서 발급   |  |  | | --- | --- | | req | 요청(생성). | | -x509 | 공개키 인증서와 인증 알고리즘을 사용하기 위한 PKI 표준. | | -newkey | 새로운 키를 만든다. | | rsa:4096 | 4096비트의 rsa키 생성 | | -keyout | EmperorPenguin\_key.key의 이름으로 키생성. | | -out | EmperorPenguin\_crt.crt를 생성. | | -sha256 | 생성한 키를 sha256으로 해시 | | -days 3650 | 유효기간:3650 |   **(2)Challenge Password 및 인증서 정보입력:**  (1)번 명령을 실행하면 챌린지 패스워드와 인증서 정보를 입력하게 된다. 챌린지 패스워드는 기억해두자.  **Challenge Password[]:136987**  **Country Name (2 letter code) []:KR**  **State or Province Name (full name) []:Seoul**  **Locality Name (eg, city) []:Jongno-Gu**  **Organization Name (eg, company) []:EmperorPenguinRootCA.Inc**  **Organizational Unit Name (eg, section) []:EmperorPenguinRootCA**  **Common Name (eg, fully qualified host name) []:EmpeorPenguin**  **Email Address []:** 정보입력을 완료하면 .key파일과 .crt 파일을 얻는다.     1. Text     Description automatically generated**SubCA .crt(certification) 제작하기.**   **(1) SubCA PrivateKey 생성**  **openssl genrsa -out SecurityGraduation\_sub\_key.key 4096**  RSA 4096bit의 SubCA 개인키를 생성한다**.**   |  |  | | --- | --- | | genrsa | 키를 RSA알고리즘으로 제작한다. |   **(2).csr(인증요청서) 생성**  **openssl req -new -key SecurityGraduation\_sub\_key.key -out SecurityGraduation\_sub\_csr.csr -sha256**  생성한 개인키를 sha256으로 해시하여 .csr을 발급받는다.  **(3)인증서 정보 입력 및 ChallengePassword 입력**  **Country Name (2 letter code) []:KR**  **State or Province Name (full name) []:Seoul**  **Locality Name (eg, city) []:Jongno-Gu**  **Organization Name (eg, company) []:SecurityGraduationSubCA.Inc**  **Organizational Unit Name (eg, section) []:SecurityGraduationSubCA**  **Common Name (eg, fully qualified host name) []:SecurityGraduation**  **A challenge password []:136987**  **(4).crt파일 생성**  Text  Description automatically generated  **openssl x509 -req -CA EmperorPenguin\_crt.crt -CAkey EmperorPenguin\_key.key -days 3650 -CAcreateserial -in SecurityGraduation\_Sub\_csr.csr -out SecurityGraduation\_Sub\_crt.crt -sha256**  SubCA의 상위기관인 RootCA의 .crt 파일과 .key 파일을 소명하고, SubCA의 CSR을 넣어 sha256으로 해시된 유효기간 3650일의 SubCA의 .crt 파일 생성.  위의 명령이 실행되면 RootCA로부터 서명이 완료된 .crt파일을 얻을 수 있다.   |  |  | | --- | --- | | -CA | 상위기괸 .crt파일 | | -CAkey | 상위기관 .key파일 | | -CAcreateserial | 시리얼 자동 지정 및 시리얼 파일 생성(.srl) |  1. **Server .crt(certification) 제작하기** 2. **Server PrivateKey 생성**Graphical user interface, text     Description automatically generated   **openssl genrsa -out SangmyungBank\_key.key 4096**  RSA 4096bit의 Server 개인키를 생성한다**.**  **(2) .csr(인증요청서) 생성**  **openssl req -new -key SangmyungBank\_key.key -out SangmyungBank\_csr.csr -sha256**  생성한 개인키를 sha256으로 해시하고 .csr을 발급받는다.  **(3)인증서 정보 입력 및 ChallengePassword 입력**  **Country Name (2 letter code) []:KR**  **State or Province Name (full name) []:Seoul**  **Locality Name (eg, city) []:Jongno-Gu**  **Organization Name (eg, company) []:SangmyungBank**  **Organizational Unit Name (eg, section) []:SangmyungBank.Security**  **Common Name (eg, fully qualified host name) []:SangmyungBank**  **Email Address []:**  **A challenge password []:136987**  **(4) .crt 파일 생성**  **openssl x509 -req -CA SecurityGraduation\_Sub\_crt.crt -CAkey SecurityGraduation\_sub\_key.key -days 1825 -set\_serial 01 -in SangmyungBank\_csr.csr -out SangmyungBank\_crt.crt -sha256**  Server의 상위기관인 SubCA의 .crt 파일과 .key 파일을 소명하고, SubCA의 CSR을 넣어 sha256으로 해시된 유효기간 1825일의 SubCA의 .crt 파일 생성.     1. **체인구조 완성하기.**   **(1).crt 파일들 엮기**  **cat Server.crt SubCA.crt RootCA.crt > chain.pem**  cat 명령어를 사용해 Server->SubCA->RootCa순으로 입력해서 pem파일을 추출한다.  **(2).keystore 파일로 변환하기**  **Openssl pkcs12 -export -name 발급사이트 -in chain.pem -inkey Server.key out keystore.p12**  #발급사이트 부분엔 다른문자를 넣어도 발급가능하다.  위에 추출한 pem파일과 server 키를 넣어서 .keystore 파일을 만든다.  **(3)위의 .keystore파일을 적용하면 1페이지의 결과를 얻을 수 있다.** |  |
| * **특이사항 / 협업 사항**   **Challenge Password: RootCA에서 설정하는 일종의 공개키이다. CSR을 발급받을 때 입력하며,이 패스워드로 검증해 서명을 한다.**  **SSC(Self-Signed-Certificate):본인이 직접 서명한 인증서로 상위기관이 없는 RootCA이다.**  **CSR(certificate signing request):상위기관에 서명을 요청하기 위해 사용하는 인증요청서이다.**  **CSR을 통해 인증서에 서명을 받고 CRT를 발급받는다.**  **CRT(Certificate):인증서** | |