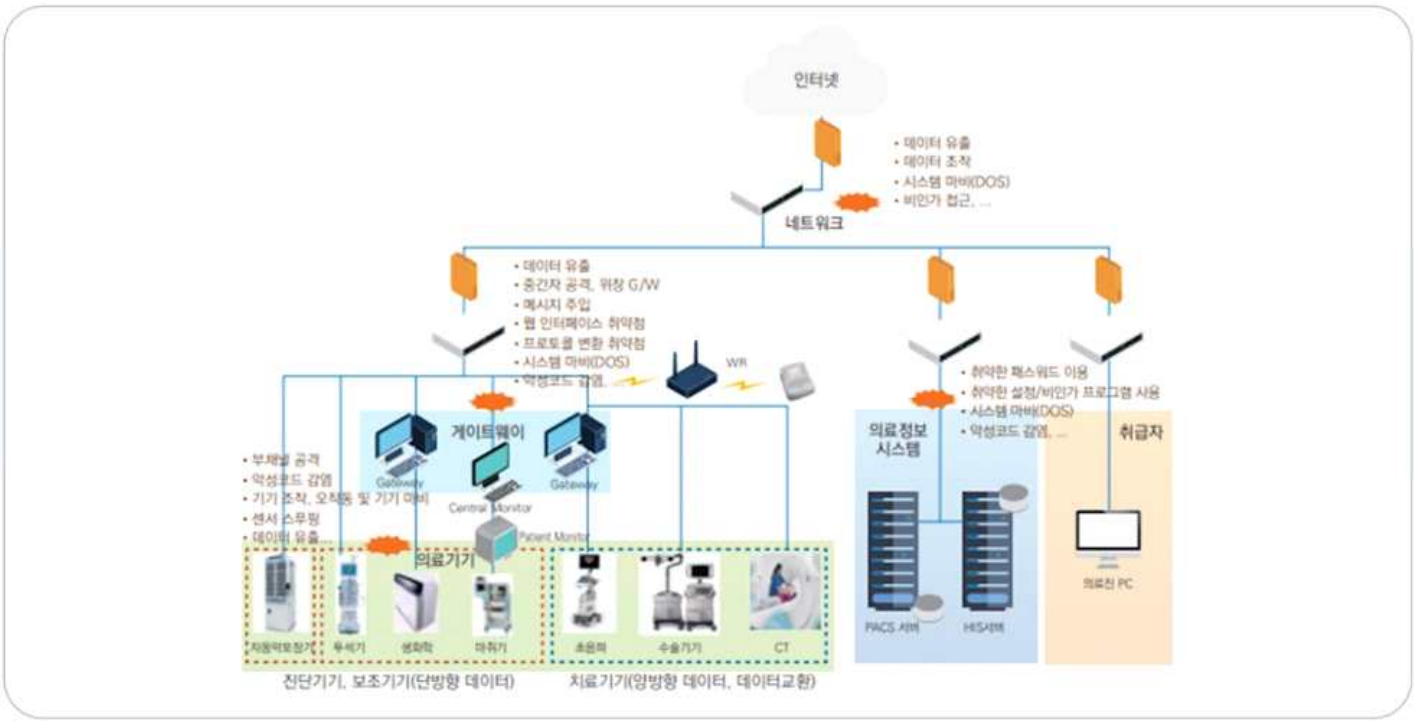
스마트의료 분야의 공격대상 별 보안위협



스마트의료 기기 보안위협의 종류

- 디버그 포트를 이용한 펌웨어 획득 : 개발 시 사용된 디버그 포트를 제거하지 않아 펌웨어 등을 획득하는 공격으로 공격자가 내부 소스코드 및 구조를 파악 할 수 있으며, 이를 기반으로 알려지지 않은 취약점을 확인하거나 특정 부분을 변조하여 주입공격 등을 할 수 있다.
- 부채널 공격 : 전송되는 정보에 대한 암호 알고리즘이 작동할 때 전기 소모량, 전자기 신호량 등 을 분석해서 암호키 등을 유추 할 수 있다.
- USB를 통한 악성코드 감염: USB 포트를 통한 악성코드 유포 또는 정보 유출
- 센서 스푸핑: 인증체계를 적용하지 않은 센서에 스푸핑 공격 등으로 데이터 감지를 방해함으로써 의료기기 오작동 유발

[답가지]

① 스마트의료 기기 보안위협
② 스마트의료 게이트웨이 보안위협
③ 스마트의료 네트워크 보안위협
④ 스마트의료 정보시스템 보안위협

[피싱해설]

- **피싱(Phishing)**: 개인정보 (Private data)와 낚시 (Fishing)의 합성어로 인터넷 이용자들에게 유명 회사 (특히 은행)를 사칭하는 e-메일을 보내고, 위장된 홈페이지에 접속하도록 유도하여 계좌번호, 주민등록번호, 로그인 비밀번호, 인증서 암호 등의 개인정보를 입력하도록 함으로써 얻은 이들 정보를 이용해 금융사기를 일으키는 공격기법

[파밍해설]

- **파밍(Pharming)**: 피싱(Phishing)과 조작(Farming)의 합성어로 합법적으로 소유하고 있던 사용자의 도메인을 탈취하거나 도메인 네임 시스템(DNS) 이름을 속여 사용자들이 진짜 사이트로 오인하도록 유도하여 개인정보를 훔치는 공격기법

[랜섬웨어 해설]

- **랜섬웨어(Ransomware)**: 몸값을 뜻하는 Ransom과 Software(소프트웨어)가 더하여진 합성어로, 컴퓨터 시스템을 감염시켜 접근을 제한하고 일종의 몸값을 요구하는 악성 소프트웨어의 한 종류. 일반적으로 내부 파일을 암호화하여 암호해독 키제공에 대한 금전을 요구함.

[DDoS 해설]

- **DoS (Denial of Service)**: 과도한 트래픽을 유발하여 시스템의 중요자원을 완전 점거하고 불능상태로 만들어 기본적으로 사용자의 가용성을 저해하는 공격방법

- **DDoS (Distributed DoS)**: 분산된 많은 수의 좀비 PC, 봇넷을 이용하여 타겟 서버의 성능저하 및 시스템 마비를 일으키는 공격

- **DrDoS (Distributed Reflection DoS)**: 공격자가 정상적인 서버에 연결요청을 보내고, 정상적인 서버들의 응답을 victim에게 하도록 하여 반사와 증폭을 이용한 서비스 거부 공격

A3

무선랜의 보안 취약점 해설 (물리, 기술, 관리)

- 무선랜의 물리적인 보안 취약점
  - 도난 및 파손
  - 구성설정 초기화
  - 전원 차단
  - LAN 차단
  - SSID 및 암호 유출

A3

무선랜의 보안 취약점 해설 (물리, 기술, 관리)

- 무선랜의 기술적인 보안 취약점
- 암호화 하지 않은 통신 데이터에 대한 도청 (WEP > WPA > WPA2)
- 무선전파 전송 장비에 대한 서비스 거부 공격 (SSID를 포함한 Probe Request)
- 불법 AP를 통한 전송 데이터 수집

A3

무선랜의 보안 취약점 해설 (물리, 기술, 관리)

- 무선랜의 관리적 보안 취약점
- 무선 AP 장비에 대한 관리 미흡 : 파손, 도난, 개수, 불법설치 등에 대한 파악 부족
- 사용자의 보안의식 부족: AP의 보안기능 미설정, AP 관리 페이지의 Default 비밀번호 사용 등
- 전파 출력 관리 부족: 워드라이빙으로 전파 수신, 중심 주파수 기준 3개 채널까지 전파 간섭 발생 등

[답]

다음 중 무선 랜(Wireless LAN)의 기술적 보안 취약점에 해당하지 않는 것은?

[답가지]

① 도청

② 서비스 거부

③ 불법 AP(Rouge AP)

④ 전파관리 수준의 미흡

답: 망분리

- **물리적 망분리**

물리적 망분리

- **패쇄망**: 업무망의 컴퓨터에 인터넷망과의 연결점을 제거하여 개인정보유출 경로를 차단하는 방법
- **물리적 PC 분리 방식**: 별도의 패쇄망을 구축하기 어려운 경우에 망을 분리할 수 있는 방법으로 망은 논리적으로 분리하지만 개인정보취급자의 PC는 물리적으로 분리하는 방법

외부 인터넷망 차단조치 안내서, KISA, 2013

- **논리적 망분리**

논리적 망분리

- **서버기반 논리적 망분리**: 인터넷망 접근을 위한 가상화를 지원하는 서버팜과 업무용 사용자 컴퓨터를 분리하는 방법으로, 사용자 컴퓨터에서는 업무망에만 접속이 가능하며 서버팜과 사용자 컴퓨터는 이벤트 값과 화면 값만을 송수신하여 업무망과 인터넷망을 분리
- **PC 기반 논리적 망분리**: 접속 컴퓨터의 운영체제 및 응용프로그램을 가상화 하여 사용자 컴퓨터에서 인터넷에 접속하기 위한 가상화 영역을 구성하는 방법으로, 인터넷을 통해 악성코드가 유입되더라도 가상화 영역에서 실제 영역으로 전달되지 않으며, 가상화 영역 초기화를 통해 악성코드 제거가 용이

외부 인터넷망 차단조치 안내서, KISA, 2013

- **망분리시 고려사항**

망분리시 고려사항

- 방화벽: 종류, 위치
- PC 보안 및 PMS 시스템
- 접근제어 (NAC) 시스템
- 보조기억매체 관리 시스템 (보안 USB 및 Agent SW)
- 보안 메일 시스템
- 이메일 악성코드 차단시스템

A5 시큐어 코딩

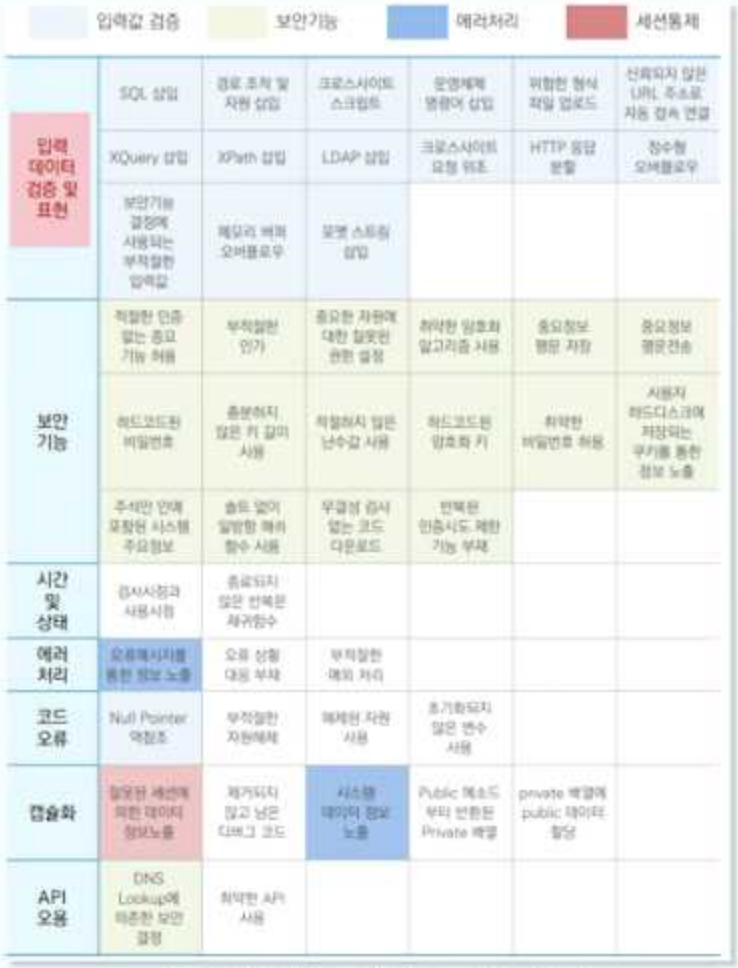
- 시큐어코딩 개념

안전한 소프트웨어 개발을 위해 소스 코드 등에 존재할 수 있는 잠재적인 보안 취약점을 제거하고, 보안을 고려하여 기능을 설계 및 구현하는 등 소프트웨어 개발 과정에서 지켜야 할 일련의 보안 활동 KISA에서 '행정기관 및 공공기관 정보시스템 구축. 운영 지침(행정안전부 고시 제 2018-21호) '에 따라 정보화 사업 수행 시 안전한 SW 개발을 위한 시큐어코딩 기법 제시

- 시큐어코딩 가이드 종류

소프트웨어 개발 보안 가이드
JAVA 시큐어코딩 가이드
C 시큐어코딩 가이드
Android-JAVA 시큐어코딩 가이드

보안취약점 47개



소프트웨어 개발 보안 가이드, KISA, 2019

상대 디렉토리 경로 조작 해설

- 상대 디렉토리 경로 조작

외부의 입력을 통하여 "디렉토리 경로 문자열" 생성이 필요한 경우, 외부 입력 값에 대해 경로 조작에 사용될 수 있는 문자열을 필터링하지 않으면, 예상 밖의 접근 제한 영역에 대한 경로 문자열 구성이 가능해져서 시스템 정보누출, 서비스 장애를 유발할 수 있는 취약점

- 안전한 코딩 기법

외부의 입력이 직접 파일 이름을 생성하는 것을 막음

불가피할 경우 다른 디렉토리의 파일 접근을 할 수 없도록 replaceAll() 등의 메소드를 사용하여 위험 문자열(", /, ₩)을 제거

외부 입력을 받아들이되, 내부적인 처리는 미리 정의한 데이터를 사용하도록 하며, 미리 정의된 케이스를 제외하고는 모두 무시

경로 순환 문자열을 이용하여 서버의 시스템 파일에 접근

상대 디렉토리 경로 조작 예

안전하지 않은 코드의 예 - JAVA

```
1:  ……
2:  public void accessFile(Properties request)
3:  {
4:      ……
5:      String name = request.getProperty("filename");
6:      if( name != null )
7:      {
8:          File file = new File("/usr/local/tmp/" + name);
9:          file.delete();
10:     }
11:     ……
12: }
```

안전한 코드의 예 - JAVA

```
1:  ……
2:  public void accessFile(Properties request)
3:  {
4:      ……
5:      String name = request.getProperty("user");
6:      if ( name != null && !"".equals(name) )
7:      {
8:          name = name.replaceAll("/", "");
9:          name = name.replaceAll("\\", "");
10:         name = name.replaceAll(".", "");
11:         name = name.replaceAll("&", "");
12:         name = name + "-report";
13:         File file = new File("/usr/local/tmp/" + name);
14:         if (file != null) file.delete();
15:     }
16:     ……
17: }
```

Null 포인터 역참조

- **Null 포인터 역참조(코드오류)**

  Null 포인터 역참조는 '일반적으로 그 객체가 Null이 될 수 없다'라는 가정을 위반할 때 발생하는 취약점으로 그 결과로 발생하는 예외 사항을 추후 공격을 계획하는데 사용됨

- **안전한 코딩 기법**

  Null이 될 수 있는 레퍼런스는 참조하기 전에 널 값인지 검사하여 안전한 경우만 사용

  C 언어에서는 포인터를 이용하는 경우 반드시 해당 포인터가 null인지 검사

안전하지 않은 코드의 예 – JAVA

```
1:  ……
2:  public void checknull()
3:  {
4:      String cmd = System.getProperty("cmd");
5:      // cmd가 널 인지 체크하지 않았다.
6:      cmd = cmd.trim();
7:      System.out.println(cmd);
8:  ……
```

안전한 코드의 예 – JAVA

```
1:  ……
2:  public void checknull()
3:  {
4:      String cmd = System.getProperty("cmd");
5:      // cmd가 null인지 체크하여야 한다.
6:      if (cmd != null)
7:      {
```

(보기 1)답

- **6번 라인:** 외부로부터 입력된 파일 명(name)이 검증없이 삭제할 파일의 경로 설정에 사용되고 있어 의도하지 않았던 파일이 삭제되어 시스템에 악영향을 줄 수 있다.
- **7번 라인:** 삭제하려는 해당 파일이 존재하는지 즉, null 체크를 하지 않아 Null Pointer Exception이 발생할 수 있다

```
[보기 1]
1: ……
2: public void f(Properties request) {
3:     ……
4:     String name = request.getProperty("filename");
5:     if(name != null && !"".equals(name)) {
6:         File file = new File("/usr/local/tmp/" + name);
7:         file.delete();
8:     }
9:     ……
10: }
```

(보기 2)답

- **6번 라인**: 문제에서 지정한 문자 "/" 에 대해 상대경로 설정에 이용되지 않도록 replace() 메소드 또는 replaceAll() 메소드를 이용하여 특수 문자를 제거
- **8번 라인**: 파일 삭제에 앞서 해당 파일이 존재하는지 null 체크를 하여 Null Pointer Exception이 발생하지 않도록 if 구문으로 확인

```
1: ......
2: public void f(Properties request) {
3: ......
4:     String name = request.getProperty("filename");
5:     if (name != null && !"".equals(name)) {
6:         name = name.replace("/", " ") ;
7 :        File file = new File("/usr/local/tmp/" + name);
8:         if (file != null) file.delete() ;
9:     }
10: ......
11: }
```