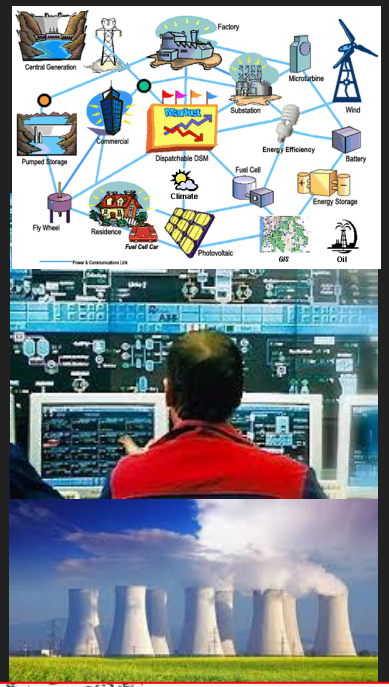
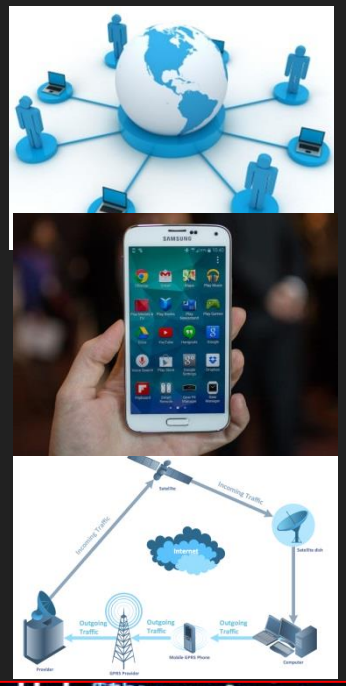
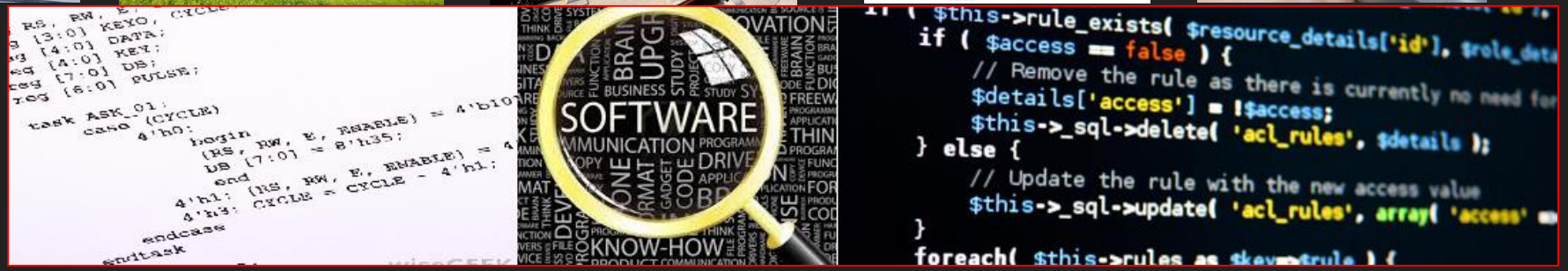
소프트웨어 보안을 위한 바이너리 분석 기법 동향

동국대학교 손윤식

4차산업혁명 시대: 소프트웨어 중심 사회



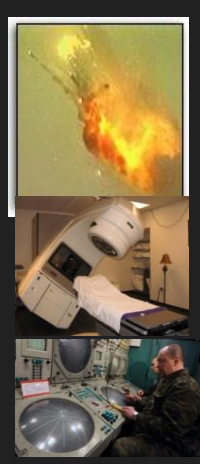




4차산업혁명 시대의 위협: 소프트웨어 버그/에러/오류

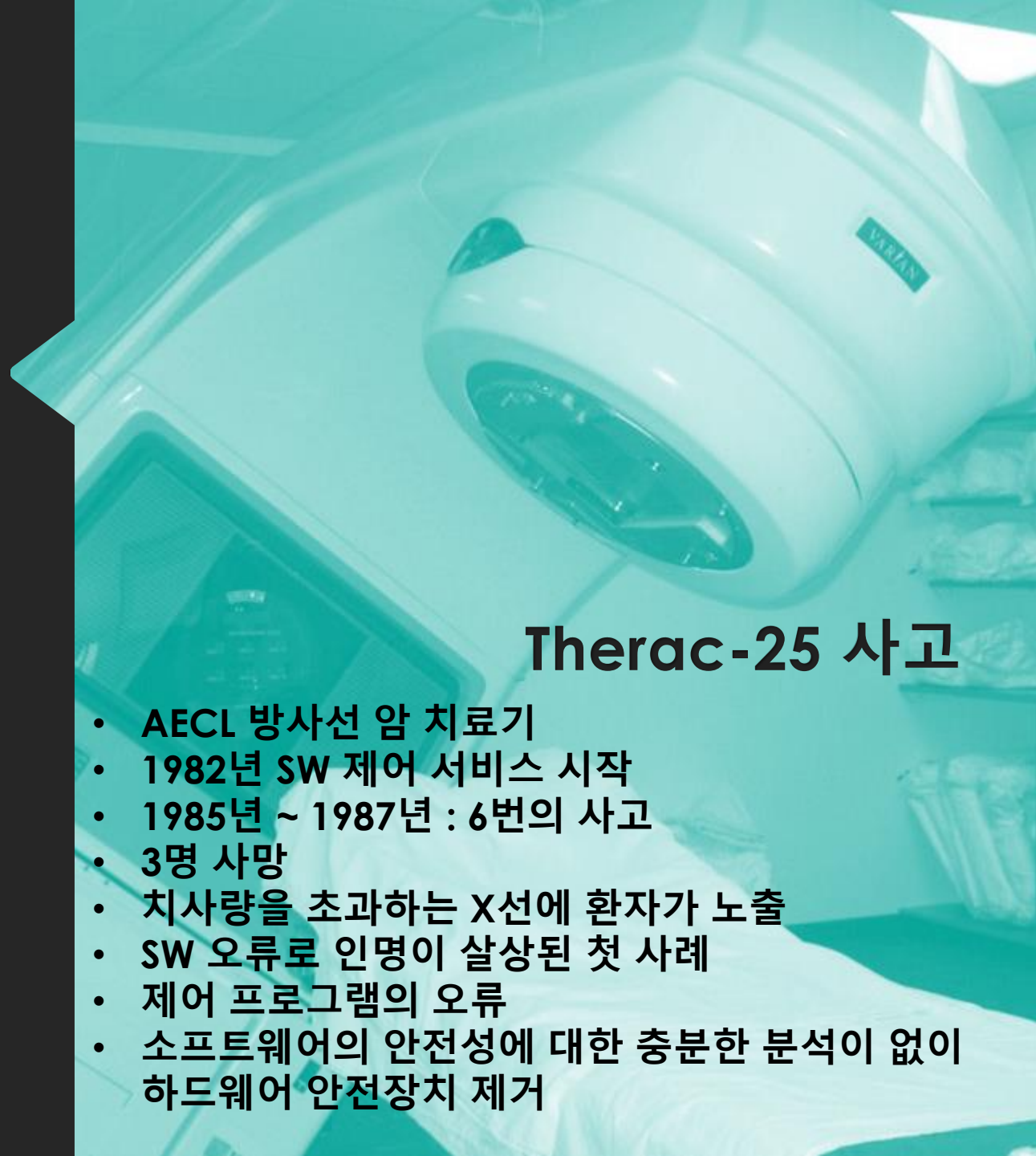
- (일반적인) 버그 :
  - 정상적 작동 중 설계/코딩 등의 잘못으로 발생하는 버그
  - SW 품질, 안전성, 신뢰성 등 저하

- 취약점 :
  - 정상적 작동 중 공격자의 의도로 발생하는 보안 약점
  - 사이버 공격에 사용



소프트웨어 안전성 (Safety) 탄생

- SW 공학에 Safety 개념 도입
- SW 신뢰성 ≠ SW 안전성
- IEC 61508 제정
- Functional safety of electrical/electronic/programmable electronic safety-related systems



Therac-25 사고

- AECL 방사선 암 치료기
- 1982년 SW 제어 서비스 시작
- 1985년 ~ 1987년 : 6번의 사고
- 3명 사망
- 치사량을 초과하는 X선에 환자가 노출
- SW 오류로 인명이 살상된 첫 사례
- 제어 프로그램의 오류
- 소프트웨어의 안전성에 대한 충분한 분석이 없이 하드웨어 안전장치 제거

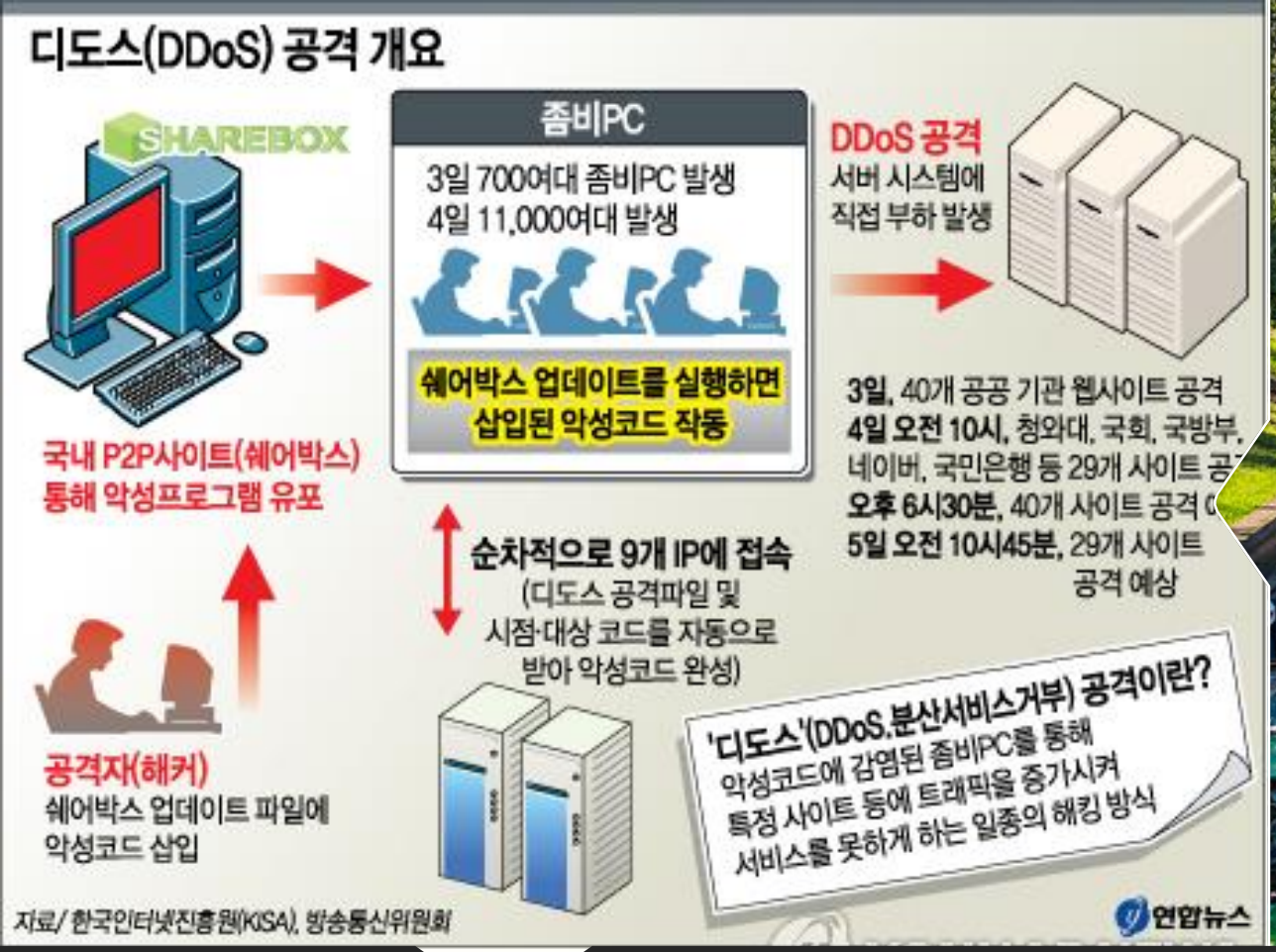
소프트웨어 안전성

- 소프트웨어 안전성 정의
  - **시스템이 재앙(인명의 손상, 시스템의 파괴 등)과 같은 실패 없이 동작하는 능력**
  - The ability of the system to operate without catastrophic failure.
  - ※ Software Engineering 10th edition, Sommerville, 2016

- Safety-critical (안전필수, 고안전, 안전중심) 정의
  - A term applied to a condition, process or item of whose proper recognition, control performance, or tolerance is essential to safe operation or use; e.q., safety, critical function,, safety critical path ..
  - ※ Air Force System Safety Handbook, 2000

산업분야별 기능 안전 표준

표준	DO-178C (12)	IEC 62304 (06)	ISO 26262 (11)	IEC 61508 (98)	EN50128(01)
개정 일자	2012	2006	2011	1998	2001
분야	항공	의료기기	자동차	전기/전자/임베디드 시스템	철도
특징	-안전 수준 5등급	-class 3등급	-ASIL 4등급 -IEC 61508 기반	-SIL 4등급 -다른 SIL 사용 표준의 모태	-SIL 5등급 -IEC 61508 기반
테스팅	-정적 분석 -동적 분석	-정적 분석 -동적 분석	-정적 분석 -동적 분석	-정적 분석 -동적 분석	-정적 분석 -동적 분석
정형기법	-정형기법 권고 Level A 필수	-정적 분석	-정적 분석	-정형기법 권고 SIL 4 필수	-정형기법 권고 SIL 4 권고



SW 보안 (Security) 탄생, 사이버 사고 (1): DDoS

SW 보안 (Security) 탄생, 사이버 사고 (2): 해킹

SW 보안 (Security) 탄생, 사이버 사고 (3): 사이버전쟁

소프트웨어 보안성 (Security)

- **실수 또는 고의의 외부 침입으로부터 시스템을 보호할 수 있는 능력**
- The ability of the system to protect itself against accidental or deliberate intrusion.

※ Software Engineering 10th edition, Sommerville, 2016.

- 소프트웨어 보안(Software Security)의 다른 정의
  - 소프트웨어가 포함하고 있는 취약점(보안약점)의 밀도

소프트웨어 보안의 어려운 점

- 프로그래밍 언어의 의미론이 명확하지 않아, 모르는 보안약점이 존재함.
- 제로데이 취약점을 줄여야 함.
- 취약점의 원인이 되는 보안약점을 줄여야 함.

- 명세에 정의 된 영역의 입력이 아닌 정의가 안된 범위의 입력을 사용
  - 예) SQL 삽입공격, 버퍼오버플로 공격 등
- 즉, 일반적인 테스트 방법 과 다른 방법이 활용

소프트웨어 보안의 어려운 점

- 공격자는 새로운 보안약점을 지속적으로 찾아내고, 이를 활용하여 제로데이 취약점 (사이버 무기)를 개발하고 있음.

- Automatic Exploit
- Patch-based Exploit

소프트웨어중심 사회 소프트웨어 특징(1)

- 신뢰성/안전성/보안성 한 속성 만 중요한 것이 아니다.
- 신뢰성 (정확성) 과 안전성도 함께 중요하고,
- 또는 신뢰성 (정확성), 안전성, 보안성도 중요하다.

- 미 국토안보부 : Secure Software
  - 속성:
    - Dependable (기능 신뢰성)
    - Trustworthy (보안 신뢰성)
    - Resilient/Survivality (생존성, 지속성)
  - 소프트웨어 및 소프트웨어 공급망 보증 (Software and Supply Chain Assurance, SSCA) 강화
  - Build Security in 포탈을 통한 자료 공개, Secure Coding 및 S-SDLC, BSIMM 지속적 발전

소프트웨어중심 사회 소프트웨어 특징(2)

- 미 국방부에서 2011년 US RQ-170 사건 이후 SW 개념의 변경

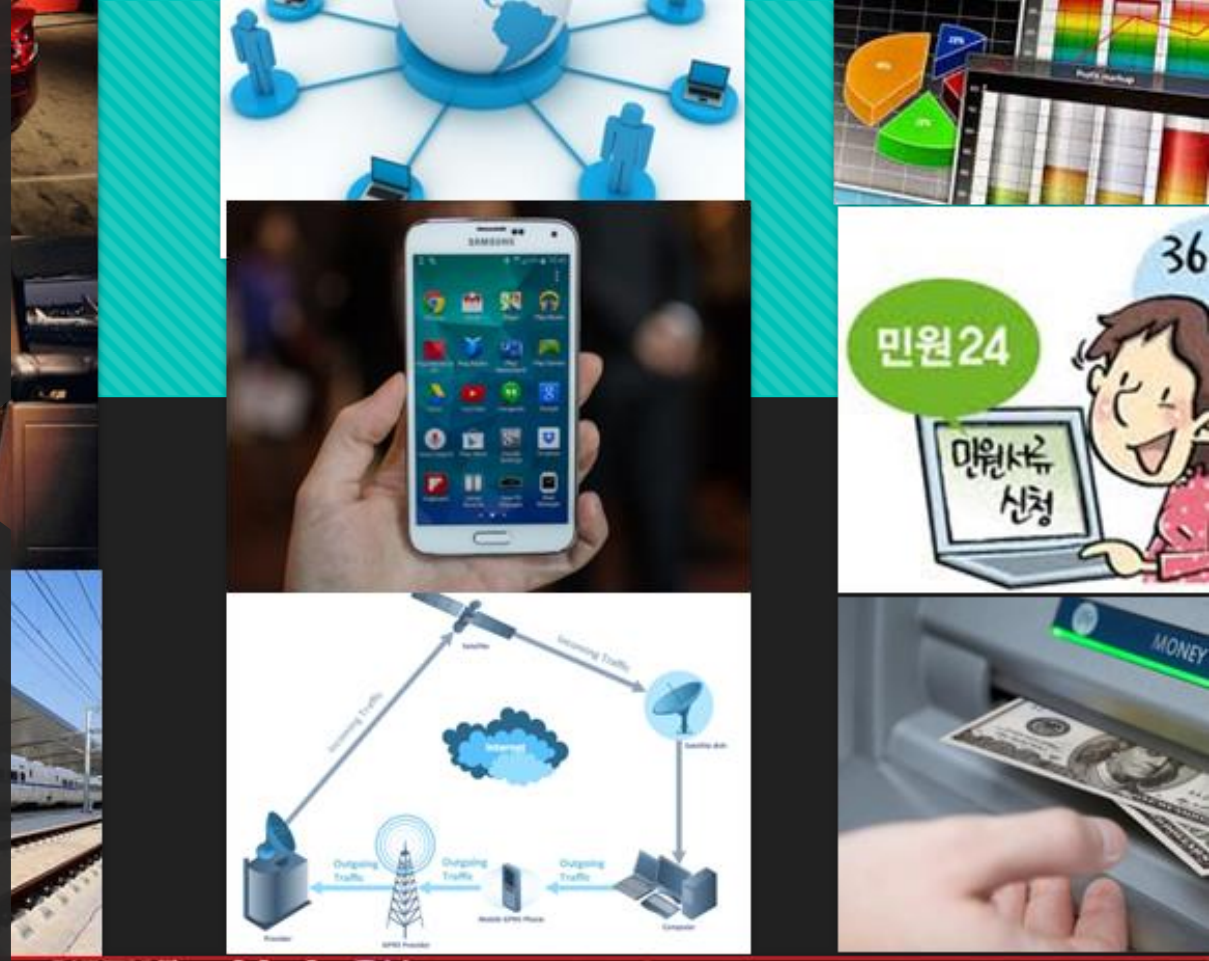
- **High-Assurance Software**
  - Correctness (기능 정확성)
  - Security (보안성)
  - Safety (안전성)

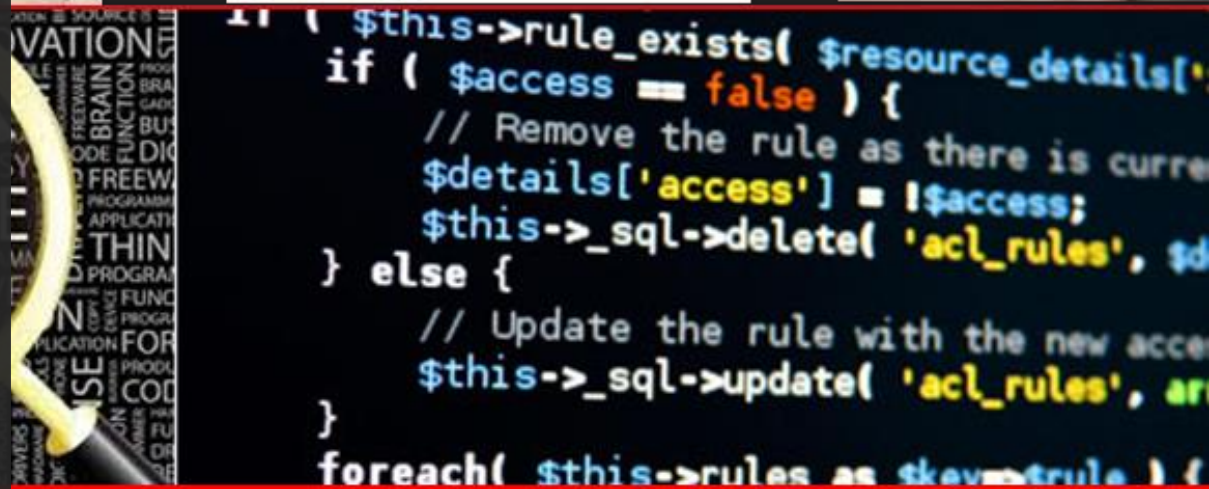
- 2014년 5월 미 국방성에서 'Hack-Proof' Drone 기술 발표 및 실용화
  - 수학적으로 증명/보증된 커널 소프트웨어 개발
  - 18개월의 성공적 테스팅
  - 실전 배치 준비 중

소프트웨어중심 사회 소프트웨어 특징(3)

- **High Confidence Software and Systems**
  - 미 NTRND
  - "… complex and networked, distributed computing systems and CPS… life-, safety-, and mission-critical applications."
  - 시스템의 설계 및 구현에 있어 규모나 구성에 관계없이 그 속성을 평가할 수 있는 시스템 디자인 혁신 등이 시도
  - NSA, NASA, 의료 등 다양한 분야에서 연구되고 있음.

- 소프트웨어 중심 사회도래
- 소프트웨어로 인한 많은 사고 발생
- 문제가 뭐 지??
- 무엇이 필요하지??
- 소프트웨어 보안을 위한 공감대 형성, 프로세스, 분석방법론 개발
- 소프트웨어를 잘 만들자!!



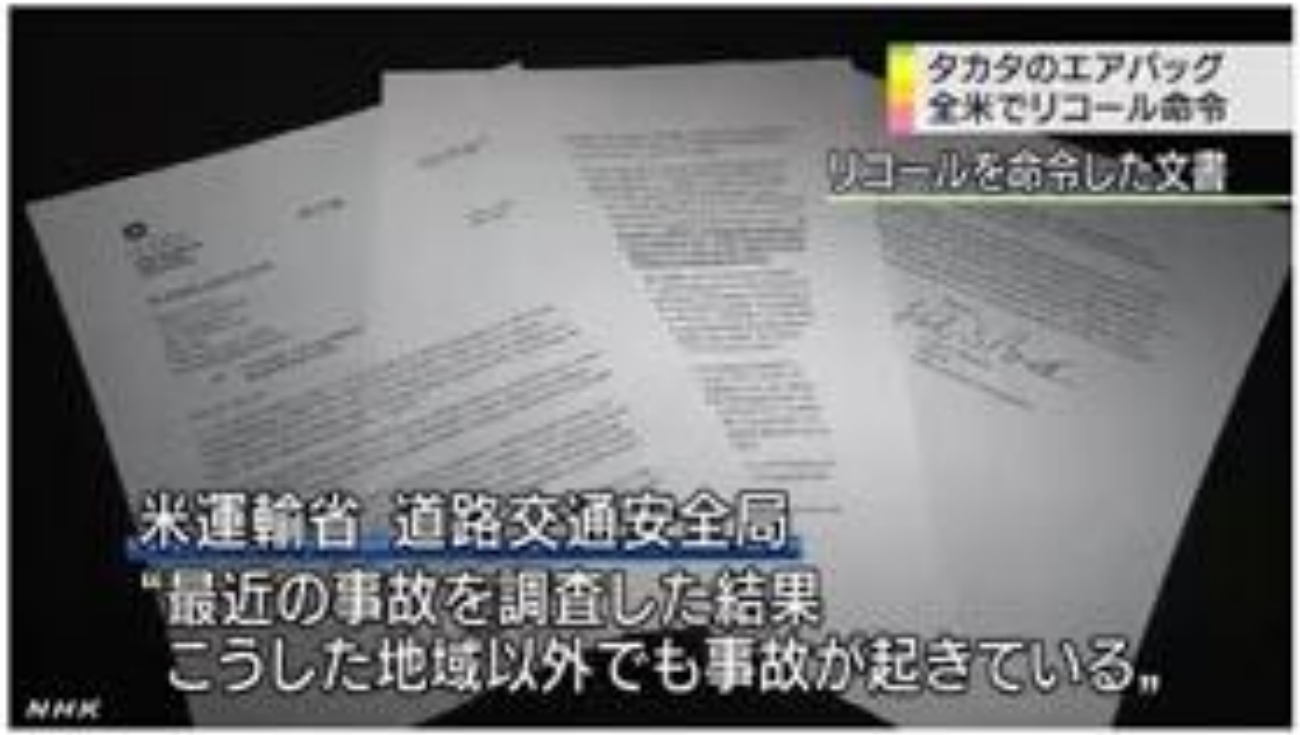


그러나 현실은…

- SW개발사 역량, 개발 비용
- 제도적인 문제
- 개발자와 해커의 관점이 다름
- SW의 복잡도 증가

米運輸省 タカタのエアバッグでリコール命令

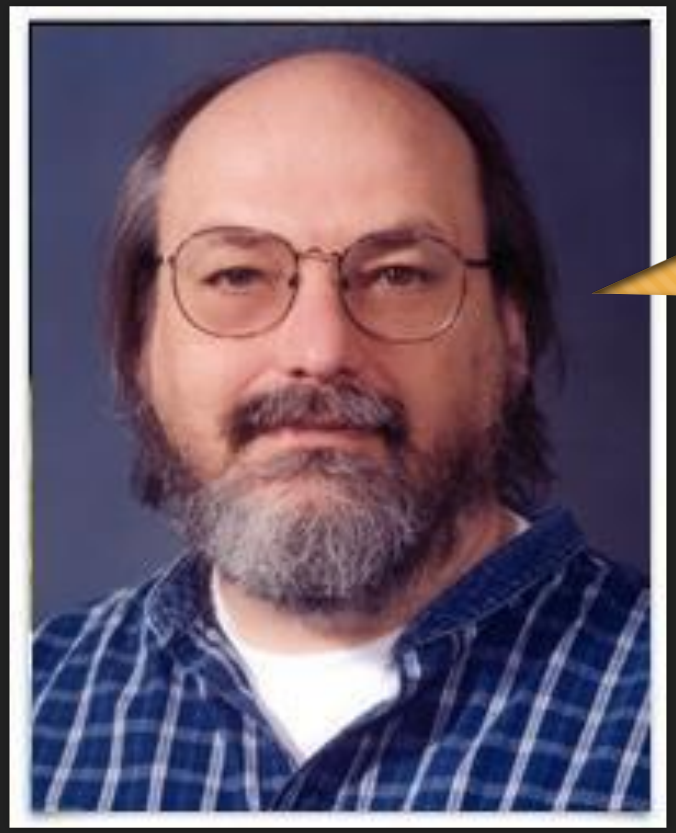
11月27日 7時13分



アメリカ運輸省の道路交通安全局は、日本の自動車部品メーカー「タカタ」が製造したエアバッグの不具合が確認された車を全米でリコールするよう タ ホ
動車メーカー5社 し
日本の部 メー
したエア 低っ 最大の高い地域で作動した場合 が飛び散って

HONDA Ford mazda BMW CHRYSLER

미국 도로교통안전국, 일본 다카타 에어백 결함 리콜 명령



You can't trust code that you did not totally create yourself. No amount of source-level verification or scrutiny will protect you from using untrusted code.

Ken Thompson
Reflections on Trusting Trust
CACM 1984

SW 보안의 어려움

- 소스코드 분석 기술만으로 프로그램의 보안성을 보장할 수 없음
- SW는 수많은 라이브러리(바이너리)와의 결합으로 구현됨
- 특히 내가 직접 개발하지 않은…

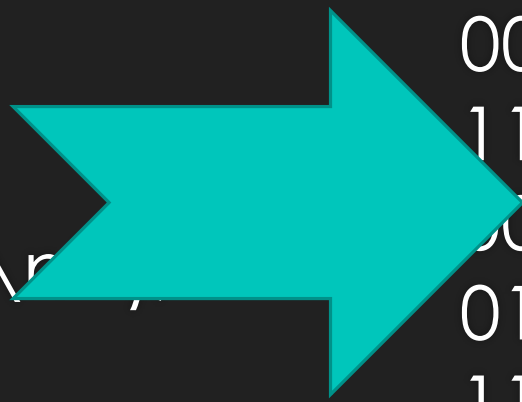
40% 의 임베디드 소프트웨어는 3rd-party library를 사용 중
2015 VDC Report: Software Quality and Security Challenges from Rapid Rise of Third-Party Code

소프트웨어 보안 분석기법의 다변화 필요

- 특정 SW의 보안성 수준
  = 해당 SW의 가장 약한 컴포넌트의 보안성 수준
- SW는 다양한 라이브러리, COTS와의 결합으로 구성
- 소스 코드가 존재하지 않는 경우도 매우 빈번

우리가 보는 소스코드가 실행되는 것이 아님

```
#include <stdio.h>
int main (void)
{
    printf( “hello world!\n”);
}
```



```
01010101010111111010
10101010101010010001
00100100011111111010
11111010010101000010
00101101010001011010
01010010010100101111
11101010100000001010
10110000010000001011
```

바이너리 분석에서의 고려사항

- 바이너리 파일 획득 방법
  - ISP, jTAG, UART, Flash Memory, …
- 바이너리 분석 파일 분석 이슈

•하드웨어: Intel x86/x64, ARM, MIPS 등
•운영체제: Windows, VxWorks, Linux 등
•실행파일 포맷: PE 32/64, COFF, ELF 등
•파일 타입: 실행파일, 라이브러리
•프로그래밍 언어: C/C++, Ada, Java 등
•컴파일러: MS C/C++ 컴파일러, GCC 등
•최적화 유무

기존 바이너리 분석기법

- 전통적인 역공학기법
  - 파일구조 분석
  - 디스어셈블러
  - 제어흐름생성
- 패턴매칭
- 동적테스팅
- Manual approach

바이너리 분석 기법

- 퍼징
  - 보안 취약점 유발 입력 데이터 자동 생성 기술
  - API 퍼징
    - beSTORM, peach, AxMan, COMRaider, Hamachi, MangleMe
    - Machine learning 기법과 결합 추세 – 모델링 자동화, 탐색공간 축소

바이너리 분석 기법

- 사후 디버깅
  - 메모리 덤프 분석 + 결함 위치 추정
  - 결함 위치 추정
    - AI, Model traversing, Fault propagation
    - 커버리지, 모델, 프로그램 상태, 프로그램 스펙트럼, 기계학습 기반 방법론

바이너리 분석 기법

- Exploit 자동 추출 기술
  - 취약점을 식별, 제한적인 조건하에서 exploit을 자동으로 생성
  - APEG(Automatic Patch-based Exploit Generation)
    - Proof of Vulnerability, Shell을 실행시키지 않음
  - AEG(Automatic Exploit Generation)
    - Symbolic Execution, Fuzzing -> 코드 커버리지 확대
    - 사후 디버깅 -> 취약점 식별
    - Exploit 공식 -> Exploit 자동 추출
      - Buffer Overflow, Format String Bug

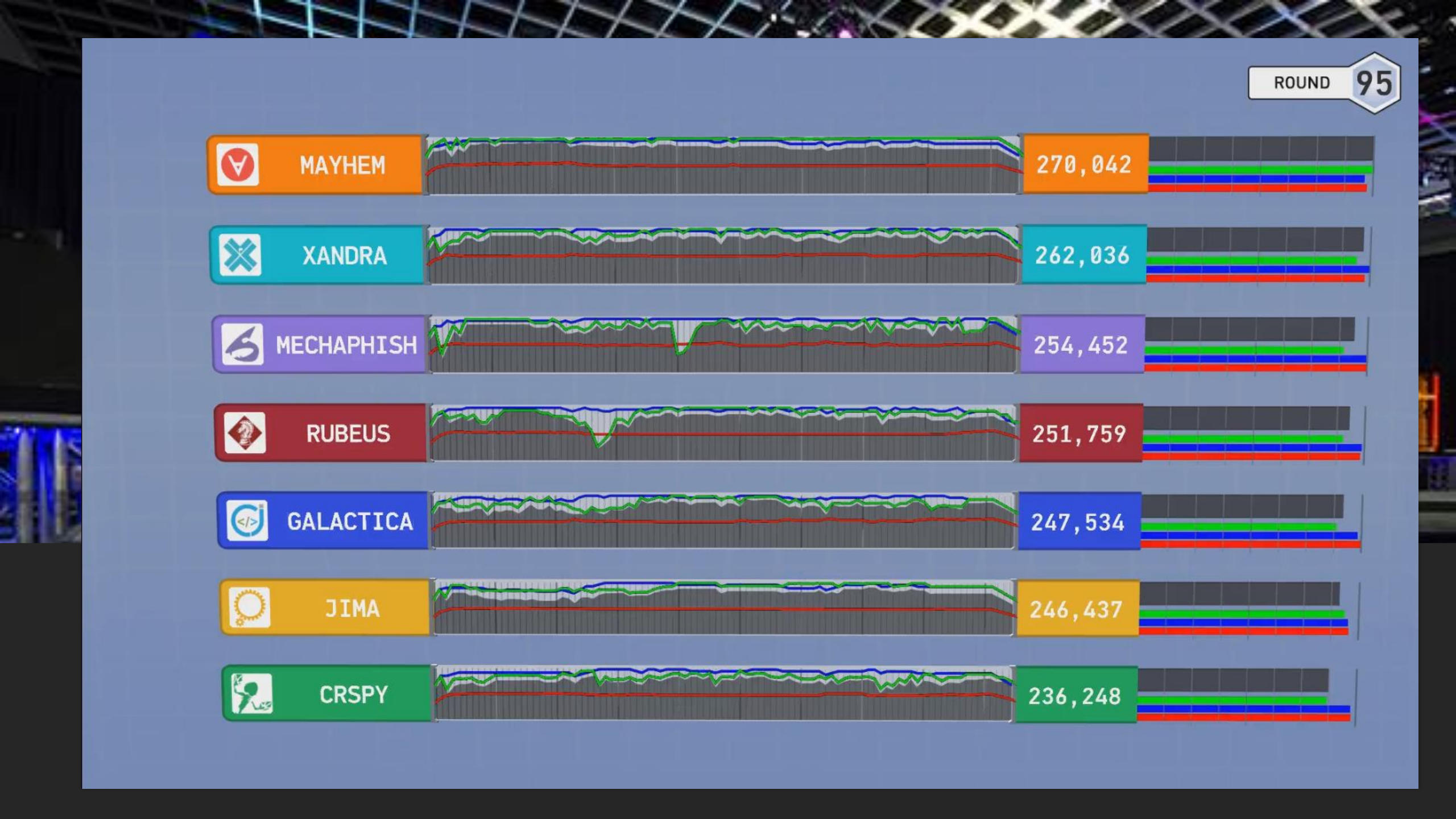
최근 연구 동향 및 이슈

- 바이너리 – 중간언어 변환
- Program abstraction 정보 재구성 정밀화 (타입, 자료구조, 객체)
- 소스코드 정적 분석기법과의 결합
- 인공지능 기법과의 결합
- 바이너리 분석자동화 Automatic Exploit Generation

Static Analysis
Symbolic Execution
SMT solver

바이너리 보안약점 분석 자동화 기술 목표

- 바이너리 파일에 존재하는 보안약점을 자동으로 검출
- 검출된 보안약점이 실제로 공격가능한지 확인
- 검출된 보안약점의 패치 자동화


ROUND 95
MAYHEM
270,042
XANDRA
262,036
MECHAPHISH
254,452
RUBEUS
251,759
GALACTICA
247,534
JIMA
246,437
CRSPY
236,248

바이너리 보안약점 분석 자동화 기술의 방향

- 전세계적인 관심 증폭
  - 싱가폴 NUS 에서는 70억규모의 대형 프로젝트가 진행중
  - MS에서도 바이너리 분석 툴을 상용화하려고 계획중
- SW보안, 자동화된 해킹-보호 기법의 핵심
  - 정적 분석 기법을 통한 취약점 자동 탐지
  - 기호실행을 통한 공격코드 생성
  - 취약점 보호를 위한 패치 자동 생성
- 인공지능 기술과 결합
  - 하나 이상의 취약점의 연계, 다양한 취약점 공격에 대한 자동화
  - 취약점 탐지의 효율향상을 위한 탐색 전략 개발

Question?