

**Introduction to Software Security**

# **Software Risks**

**(Bugs, Flaws, Vulnerabilities)**

**Seong-je Cho**

**Computer Security & Operating Systems Lab,  
Dept. Software Science, DKU**

# Sources / References

---

- *Information Security: Principles and Practice*, 2<sup>nd</sup> edition by Mark Stamp, Wiley, 2011
- Common Weakness Enumeration (CWE)
  - <https://cwe.mitre.org/>
- Software Security: Principles, Policies, and Protection, Matias Payer, Apr. 2019.
- N. Vljajic, CSE 3482: Introduction to Computer Security, Yorku
- Nicholas Weaver, Computer Science 161: Computer Security, Berkeley
- Myrto Arapinis, Computer Security: INFRA10067, University of Edinburgh
- Lecture 12 Program Security, CS 450/650 Lecture

**Please do not duplicate and distribute**

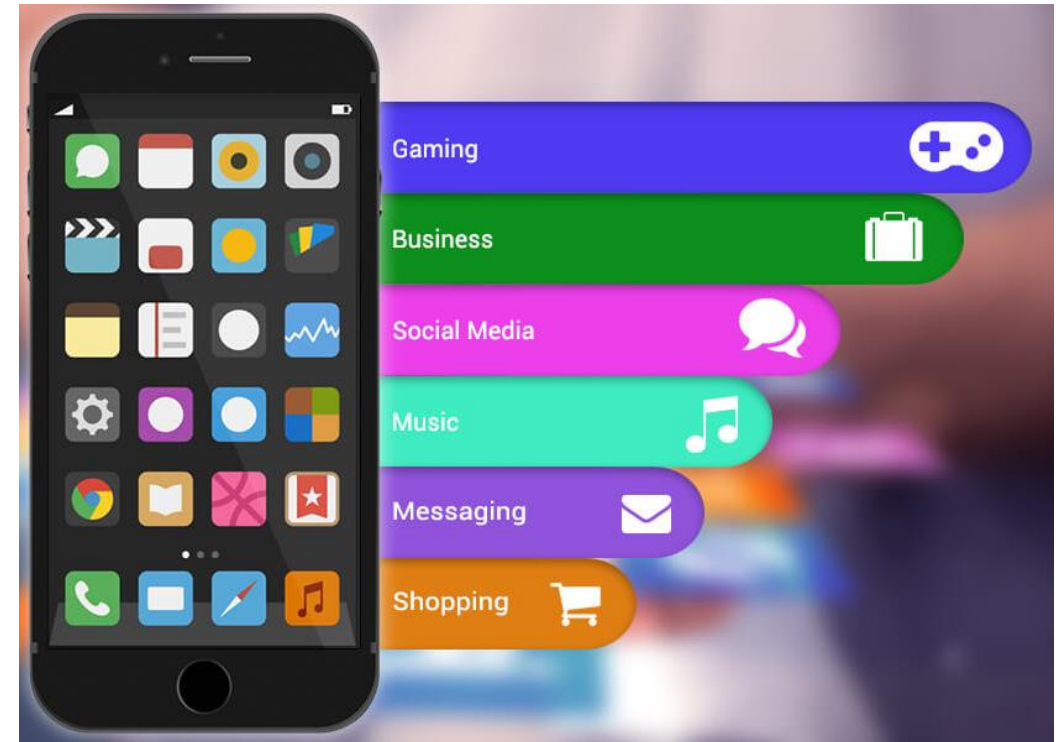
# Contents

---

- **Why is Software Security critical?**
- **Examples Software Bugs/Flaws**
  - Improper initialization
  - Side effects
  - Scoping
  - Control flows
  - Integer security
  - Null pointer dereference
  - Operator precedence logic error
- **Buggy software is dangerous**
- **Bug Bounty**

# Software is ubiquitous

- **Systems software** : OS, compiler, loader
- **Business software** : Payroll, accounting
- **Scientific and engineering software**
  - Computer-aided design, simulation, weather prediction, ...
- **Internet software:**
  - B2C: business-to-customer (e.g., amazon.com)
  - Facebook, Google Chrome, ...
- **PC software** : Spreadsheets, word processing, games, ...
- **Embedded software**
  - Cars, microwave ovens, cable boxes, light switches,
  - “smart dust”, ...
- **Mobile applications**
- **SaaS**



Source: <https://theme-vision.com/top-5-ideas-to-sell-your-mobile-app-better/>

# Software

## Software is:

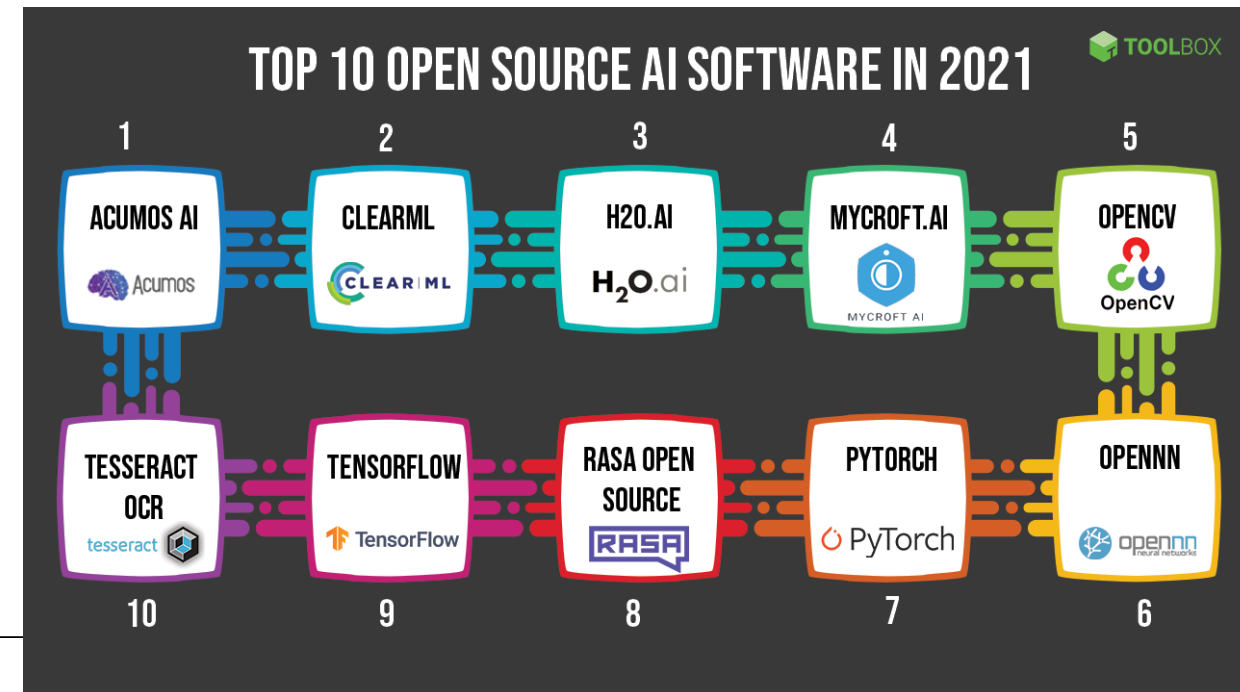
- Executable programs
- Source code, Library, ...
- Documents: user requirements, design documents, user/programmer guides, etc.

**"Software is a critical part of digital transformation,"** said Janet Kennedy

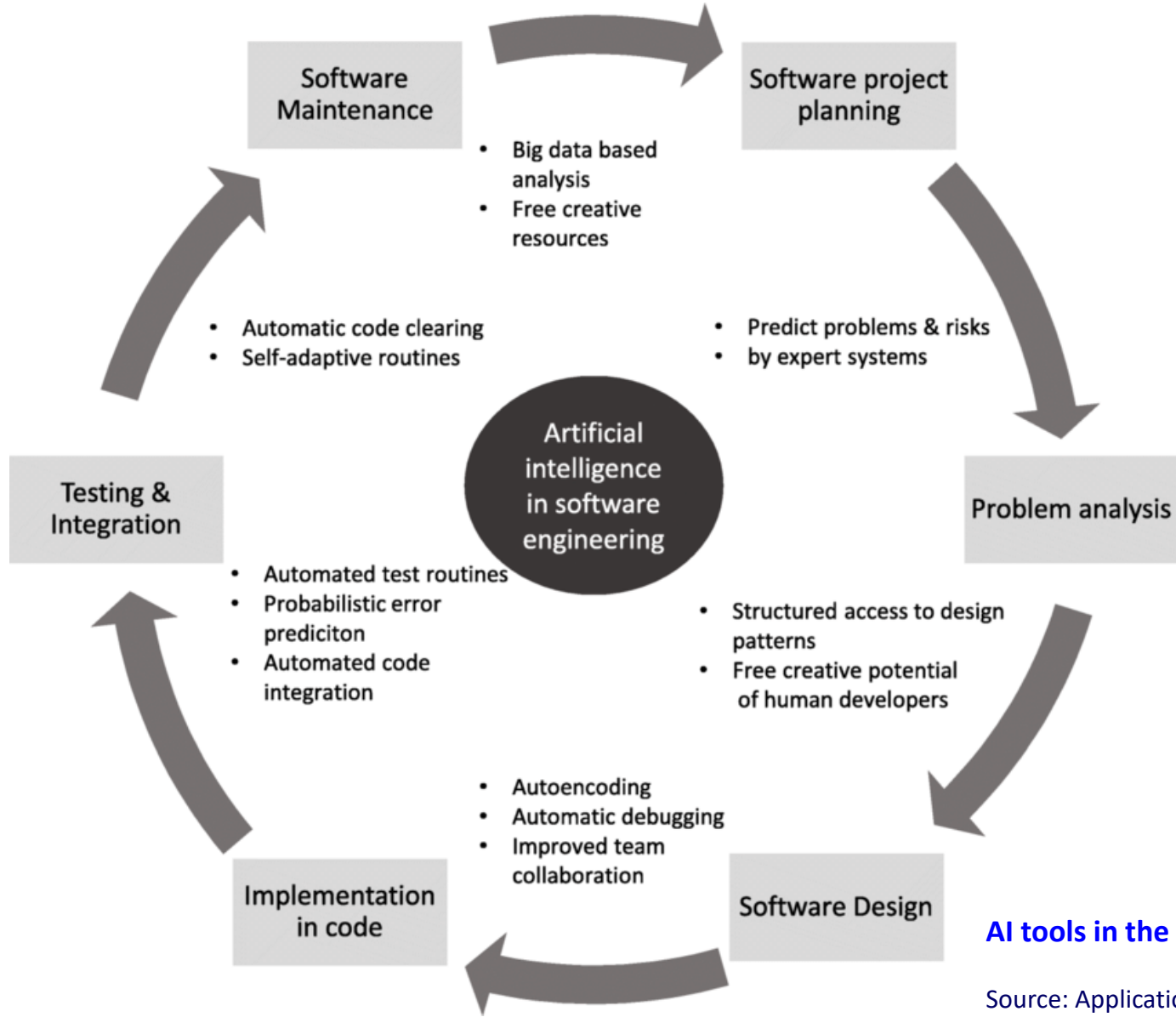
## Software plays a key role

- Process, transfer, and save data
- Produces, manages, and presents information
- In information society
  - Next step after industrial society

**Software-oriented society**, Software-centric society, ...



# SW is still eating the world



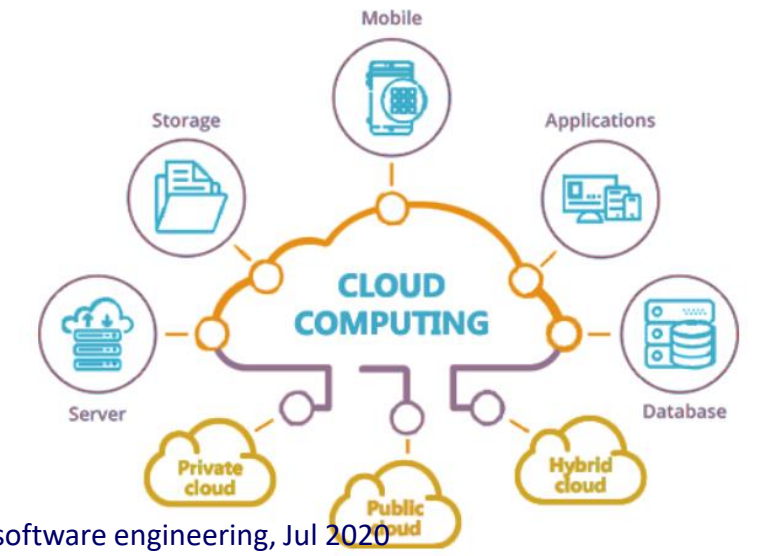
AI tools in the SDLC

Source: Applications of AI in classical software engineering, Jul 2020

## Open Source Big Data Tools



Source: Big Data Software Market Report on Global and USA Industry Forecasts described in a new market report, 2020/08/18



# Software Crisis

---

**Software Crisis It was in late 1960's Many software projects failed**

**Is was in late 1960's**

- Many SW projects failed
- Many SW projects late, over budget, providing unreliable SW that is expensive to maintain.
- Many SW projects produced SW which did not satisfy the requirements of the customer
- Complexities of software projects increased as HW capability increased
- Larger SW system is more difficult and expensive to maintain
- Demand of new SW increased faster than ability to generate new SW

All the above attributes of what was called a 'Software Crisis'.

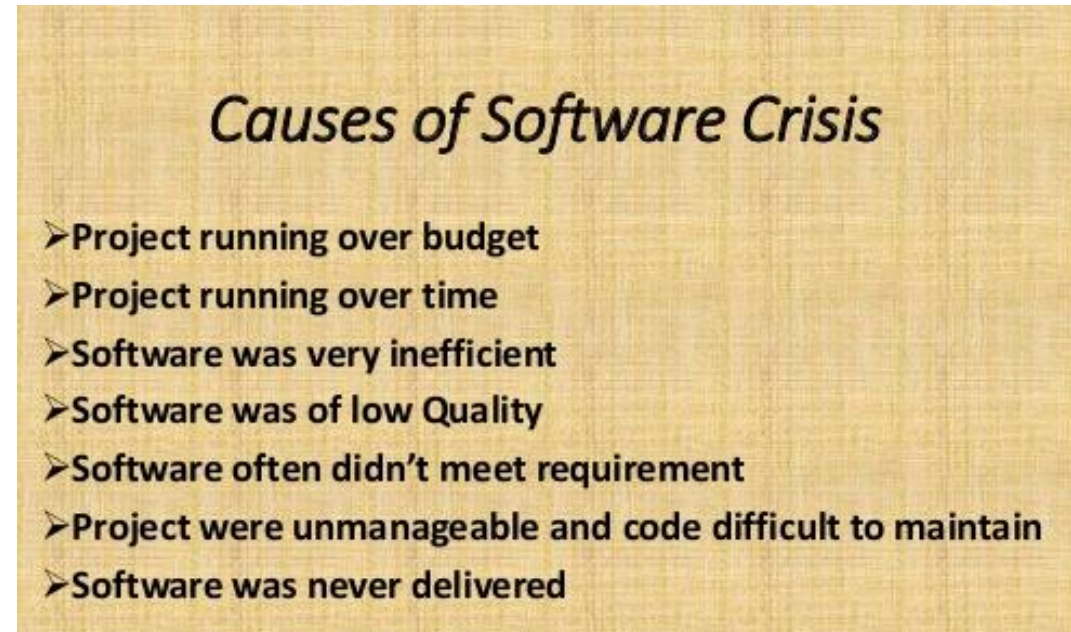
So the term 'Software Engineering' first introduced at a conference in late 1960's to discuss the software crisis.



# Software Crisis?

- People that design and build software have to deal with many problems
- **Software crisis** for the last 30 years?
  - In reality, things are not that bad
    - Many more successes than failures
    - But problems are persistent
- **The Software Engineering field is still immature**
  - e.g., comparing with civil engineering, etc.

- 👉 Source: (from an old material)
- 198:431 Software Engineering
  - [Fall 2006](#), Prof Barbara G. Ryder



Source: A presentation on Software Crisis (prepared by: Chandan Sharma)



# 1965-1985: The Software Crisis

---

- SW engineering was spurred by the so-called **software crisis** of the 1960s, 1970s, and 1980s, which identified many of the problems of software development.  
The software crisis was originally defined in terms of productivity, but evolved to emphasize **quality**.  
Some used the term software crisis to refer to their inability to hire enough qualified programmers.
- Over budget and schedule
- **Property damage**
  - **Software defects** can cause property damage.
  - Poor software security allows hackers to steal identities, costing time, money, and reputations.
- **Life and Death** (Some embedded systems used in radiotherapy machines failed so catastrophically that they administered lethal doses of radiation to patients. The most famous of these failures is the Therac-25 incident.)

Source: CS302: Software Engineering, [saylor.org](https://saylor.org).

## 1985-1989: Silver Bullets is There?

---

- For decades, solving the software crisis was paramount to researchers and companies producing software tools. They trumpeted every new technology and practice from the 1970s to the 1990s as a "silver bullet" to solve the software crisis.
  - Tools, discipline, formal methods, process, and professionalism were touted as silver bullets:
- In 1986, Fred Brooks published his *No Silver Bullet* article, arguing that no individual technology or practice would ever make a 10-fold improvement in productivity within 10 years.
  - SW 프로젝트에서 특별한 묘책[특효약]은 없다.
  - 어떤 도구나 어떤 관행, 그 하나만으로 모든 것이 해결되지 않는다.
  - 기술이든 관리 기법이든 한쪽으로만 이루어진 개발은 없으며 그 자체로 10년 안에 생산성, 신뢰성, 단순성 면에서 점진적인 개선만을 약속한다

Source: CS302: Software Engineering, saylor.org.

# Why Software Security?

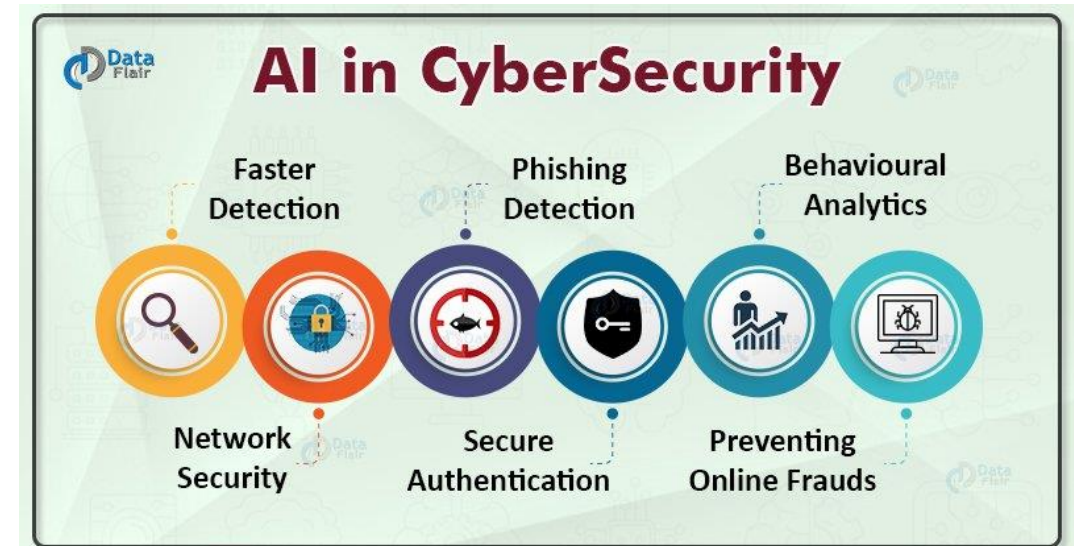
---

- Why is software as important to security as crypto, access control and protocols?
  - Virtually all of computer security is implemented in software
- If your software is subject to attack, your security is broken
  - Regardless of strength of crypto, access control or protocols
- Software can be a poor foundation for security
  - Software Vulnerability
  - Bug, Flaw, Defect, Weakness, ...

# Security and Software (Security Software)



- Code review tools: Codebrag, Gerrit, Reviewable, ...
- Open SSL (The OpenSSL Project)
- Microsoft Defender
- Anti-virus products (V3-pro, ALYac, Kaspersky, ...)
- Software Firewall, Wireshark, ...
- Security Onion (Snort), Suricata
- SolarWins Security Event Manager
- Kismet (an open-source, wireless IDS)
- OpenVPN, SoftEther VPN, ...



Source: <https://www.acmetek.com/endpoint-security-products-features/>

Source: Using Artificial Intelligence in Cybersecurity (Ultimate Guide), SOCRadar, 2022/01/11

**SW is NOT secure**  
**SW bugs can prove deadly**

# Buggy Software is Everywhere!

- **Software Architecture: Therac-25 the killer radiation machine** (in 1985)
- Therac-25는 암 환자용 방사선 치료기로서, 높은 에너지 방사광선을 빠르게 집중적으로 조사해 악성 종양을 파괴하는 장비이다.
  - 간호사와 방사선사가 환자에게 맞는 치료방식을 설정하기 위해, 기계를 조작하는 과정에서 operator가 (구성 관련) 입력 실수를 유발한 경우 sw가 이를 바로 잡아야 했는데, 실제로는 operator의 오류가 UI로 전달되어 방사선 치료가 계속되는 쪽으로 기계가 오 작동되었다.
  - 6명의 사망자가 발생하였으며, 이 사고는 인터페이스 설계 부실과, 기계 조작에 대한 방사선사의 교육이 제대로 이루어지지 않은 데 원인이 있었다.
- The main problems in the development of this software had been the following:
  1. The programmers did not assume that the operators could make a mistake in the configuration and would have to reconfigure the machine. They assumed that the operator would always take the right path of configuration.
  2. No tests of any kind were developed, that is, the software was only tested by them in the cases of successes when they were developing it.
- 출처: 소프트웨어 개발 프로세스에서의 안전성 분석 및 관리 활동의 적용방안, 중소기업융합학회 논문지, 6(1), 2016.  
Software Architecture: Therac-25 the killer radiation machine, Carlos Caballero, 2019.05.09

# Buggy Software is Everywhere!

---

- **NASA's Mars Lander (cost \$125 million)** (1999.09.30)
  - Mars Climate Obiter (화성 기후 궤도 탐사선)
  - Error in converting English (yard) and metric (meter) units of measure
    - a Lockheed Martin engineering team used English units of measurement while the agency's team used the more conventional metric system for a key spacecraft operation
  
- **Toyota unintended acceleration recall** (2009-2010)
  - Toyota's problem with unintended acceleration has been blamed on everything from the position of the floor mats to the shape of the accelerator pedal to glitches in the cars' software.
  - 도요타는 2009년에 일어난 렉서스 자동차 급발진 사고로 인해 천문학적인 손해를 보았다.  
탑승자 전원이 사망하면서 900만 대의 차량이 리콜되며 5조 원 이상의 경제적 손실을 입은 것.  
2013년에, 급발진 사태의 원인은 다름 아닌 전자제어장치(ECU)에 내장된 sw의 결함 때문인 것으로 밝혀졌다.
  - Toyota's killer firmware: Bad design and its consequences, by Michael Dunn, 2012.10.28



# SW bugs can prove deadly

---

## Bad software is everywhere!

- **Gamers May Fight Deadly Software Bugs in US Military Weapons** [Jan. 23 2012, Technews]
  - A lesson learned when a buggy Patriot missile defense system failed to intercept a Scud missile that killed 28 American soldiers during the first Gulf War in 1991.
  - To prevent such weapons disasters, the U.S. military wants to transform dull bug-hunting tasks into fun problem-solving games that attract swarms of online players

# SW bugs can prove deadly

- **Bug can cause deadly failures when anesthesia device is connected to cell phones** [Apr. 23 2014, arse-technica]
  - Federal safety officials have issued an urgent warning about **software defects** in an anesthesia delivery system that can cause life-threatening failures at unexpected times, including when a cellphone or other device is plugged into one of its USB ports.
  - Spacelabs Healthcare is recalling the ARKON Anesthesia System with Version 2.0 Software due to a **software defect**.

의료기관이 사용하는 마취기는 보통 마취기계 본체 외에도 인공호흡기와 모니터 등이 세트로 이뤄져 있다. 그런데 마취기 소프트웨어에 취약점이 발견됐다고 한다.

문제가 된 마취기 모니터는 태블릿 같은 형태로 생겼고 USB 포트가 있다. 여기에 USB 케이블을 이용해 스마트폰 같은 기기를 마취기에 연결하게 되면 환자의 생명에 위협을 가할 수 있는 사태가 발생한다는 것이다.

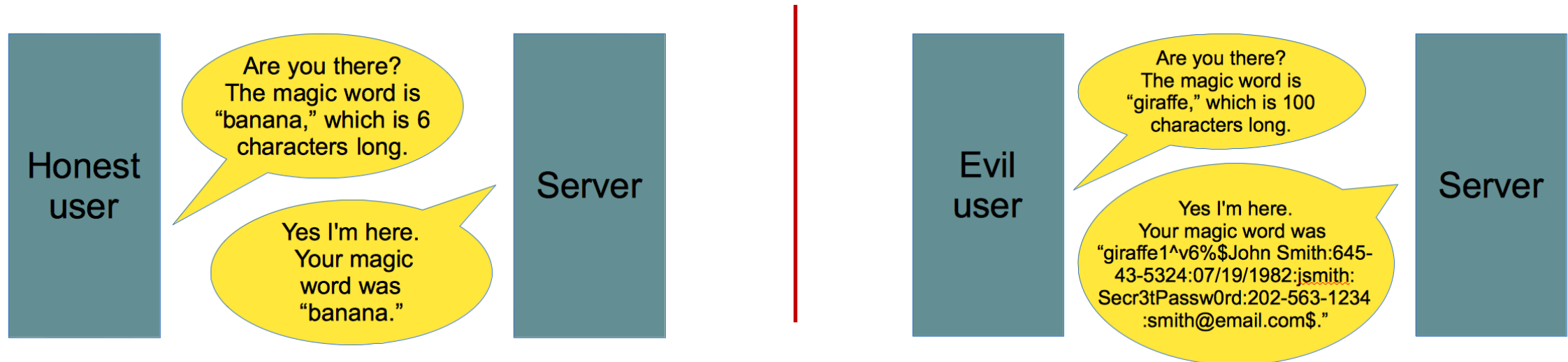
미 식품의약안전국 FDA 역시 의료기간에 리콜에 따라 경고를 하고 있다. 이에 따르면 버그로 인해 마취기가 멈출 수 있으며 마취기에 붙어 있는 USB 포트에 디지털기기를 충전하면 작동이 중지된다는 것이다. 이에 따라 사망을 초래할 수 있는 심각한 결함이지만 다행스럽게도 아직까지 사망자는 보고되고 있지 않다.

anesthesia: (의학) 마취, 무감각증,

# The Heartbleed Bug



- CVE-2014-0160 (source: <https://heartbleed.com/>)
- This weakness allows stealing the information protected, under normal conditions, by the SSL/TLS encryption used to secure the Internet.
- The Heartbleed bug allows anyone on the Internet to read the memory of the systems protected by the vulnerable versions of the OpenSSL software.
  - Without using any privileged information or credentials, it was possible to steal the secret keys used for X.509 certificates, user names and passwords, instant messages, emails and business critical documents and communication.



Source: The Heartbleed Bug, explained, by Timothy B. Lee at Vox.com

# Heartbleed and Shellshock

- **美병원 노린 개인정보유출에 '하트블리드' 악용** (ZDnet Korea, 2014/08/21)
  - 미국 병원 네트워크인 '커뮤니티 헬스 시스템(CHS)'에서 450만명의 환자 개인정보가 유출된 사건에 '하트블리드' 취약점이 악용된 것으로 추정되고 있다.
  - CHS가 해킹되는 과정에서 초기 단계에 하트블리드 취약점이 악용됐다고 보도했다. 이 취약점은 암호화 통신을 위한 프로토콜인 오픈SSL에서 발견된 것으로 서버 내 메모리에 저장된 정보를 임의로 빼내 그 중 해커가 유용한 정보를 얻을 수 있게 한다. 여기에는 서버에 접속할 수 있는 로그인 ID, 비밀번호 등이 포함될 수 있다.
  - 다시 말하면 서버에 관리자용으로 접근할 수 있는 ID, 비밀번호 등을 하트블리드 취약점을 악용해 알아낸 뒤 해당 네트워크에 VPN으로 접속해 정보를 유출시켰다는 것이다.
- **끝나지 않은 하트블리드...중소사이트 위험** (ZDnet Korea, 2014/06/23)
- **하트블리드보다 위험한 배시 취약점 '비상'** (ZDnet Korea, 2014/09/26)
  - 리눅스는 물론 애플 OS X와 같은 유닉스 기반 OS에서 사용되는 셸 프로그램인 '배시(Bash)'에서 발견된 취약점인 '셸쇼크(shellshock)'를 악용한 공격이 처음으로 발견됐다.
  - 셸쇼크는 직접 웹서버에 악성코드를 심거나 악성명령을 내릴 수 있다는 점에서 문제가 더 심각하다는 게 전문가들의 평가다. 해당 취약점은 'CVE-2014-6271'라는 이름으로 공식 분류됐다.
  - 해당 버그를 악용한 악성코드를 분석한 결과 분산서비스거부(DDoS) 공격을 위한 좀비PC를 양산하는 기능 뿐만 아니라 보안이 취약한 서버에 접근해 쉬운 비밀번호를 쓸 경우 바로 침투해 공격자가 마음대로 조정할 수 있게 하는 기능을 가졌다.
- **리눅스·OS X 셸에 중대 보안취약점 발견** (ZDnet Korea, 2014/09/25)
  - 리눅스와 유닉스 환경에서 많이 쓰이는 셸 프로그램 '배시(Bash)'에 중대한 보안취약점이 발견됐다. 셸 명령어를 실행하는데 사용되는 배시의 취약점을 이용해 해커가 관리자권한 인증없이 서버를 제어할 수 있다는 것이다.

## Shellshock (Bashdoor)

# Shellshock: Romanian hackers are accessing Yahoo servers, claims security expert



Hackers are exploiting Yahoo's bash bug vulnerability, according to technology consultant Jonathan Hall

Zachary Davies Boren • Monday 06 October 2014 18:50

## First Shellshock botnet attacks Akamai, US DoD networks

### CVE identifier(s)

- CVE-2014-6271 (initial),
- CVE-2014-6277,
- CVE-2014-6278,
- CVE-2014-7169,
- CVE-2014-7186,
- CVE-2014-7187

By Juha Saarinen  
Sep 26 2014  
7:18AM

7 Comments



RELATED ARTICLES

### Wopbot on the rampage.

Attackers have been quick to exploit the **Shellshock Bash command interpreter bug** disclosed yesterday by building a botnet that is currently trying to infect other servers, according to a security researcher.

The "wopbot" botnet is active and scanning the internet for vulnerable systems, including at the United States Department of Defence, chief executive of Italian security consultancy Tiger Security, Emanuele Gentili, told *iTnews*.

```
recv:
recv:
buf: PONG

recv:
buf: UDP Flooding 178.172.239.241:80 for 50 seconds.

recv:
buf: PONG
```

# Shellshock vulnerability

## Shellshock knowledge Prerequisites

- **bash** supports **environment variables**
- You can invoke existing ones or add new ones

```
tudor@ubuntu: ~  
tudor@ubuntu:~$ echo -e $USER'\n'$PATH  
tudor  
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games  
tudor@ubuntu:~$ export CONGRATS="Felicitari Simona Halep!"  
tudor@ubuntu:~$ echo $CONGRATS  
Felicitari Simona Halep!  
tudor@ubuntu:~$
```

- Let's talk about bash functions

- Can be used in .sh scripts
- Can be defined in "one-liners"

```
tudor@ubuntu: ~  
tudor@ubuntu:~$ welcome() { echo "Hi $USER, here's the date: "; date; }  
tudor@ubuntu:~$ welcome  
Hi tudor, here's the date:  
Thu Oct 23 02:35:46 PDT 2014  
tudor@ubuntu:~$
```

- Can also be defined in **environment variables**

```
tudor@ubuntu: ~  
tudor@ubuntu:~$ export bunvenit="() { echo \"Hi $USER, here's the date:\"; date; }"  
tudor@ubuntu:~$ bash -c 'bunvenit'  
Hi tudor, here's the date:  
Thu Oct 23 02:59:37 PDT 2014  
tudor@ubuntu:~$
```

Source: Shellshock Vulnerability, by Tudor Enache, OWASP

# Shellshock Vulnerability

- **셸 함수도 다른 bash 인스턴스로 Exporting 하는 것이 가능**

[환경변수]='() { return: };'

- Environment 를 통해 새롭게 추가된 함수정의를 전파하기 위해 함수이름으로 환경변수를 만들고 그 변수 안에 "() {" 로 시작하는 함수 내용을 정의

- **Shellshock 취약점이 있는 경우,**

- 취약한 bash 의 경우, 함수 정의 뒤에 임의의 명령을 추가하면 bash 에서 해당 환경변수를 임포트할 때 끝에 추가된 명령도 같이 실행.

[환경변수]='() { return: }; [임의의 명령]'

- This flaw is triggered when extra code is added to the end of these function definitions (inside the environment variable). Something like:

```
$ env x='() { :}; echo vulnerable' bash -c "echo this is a test"
vulnerable
this is a test
```

- **Shellshock 취약점이 패치된 경우,**

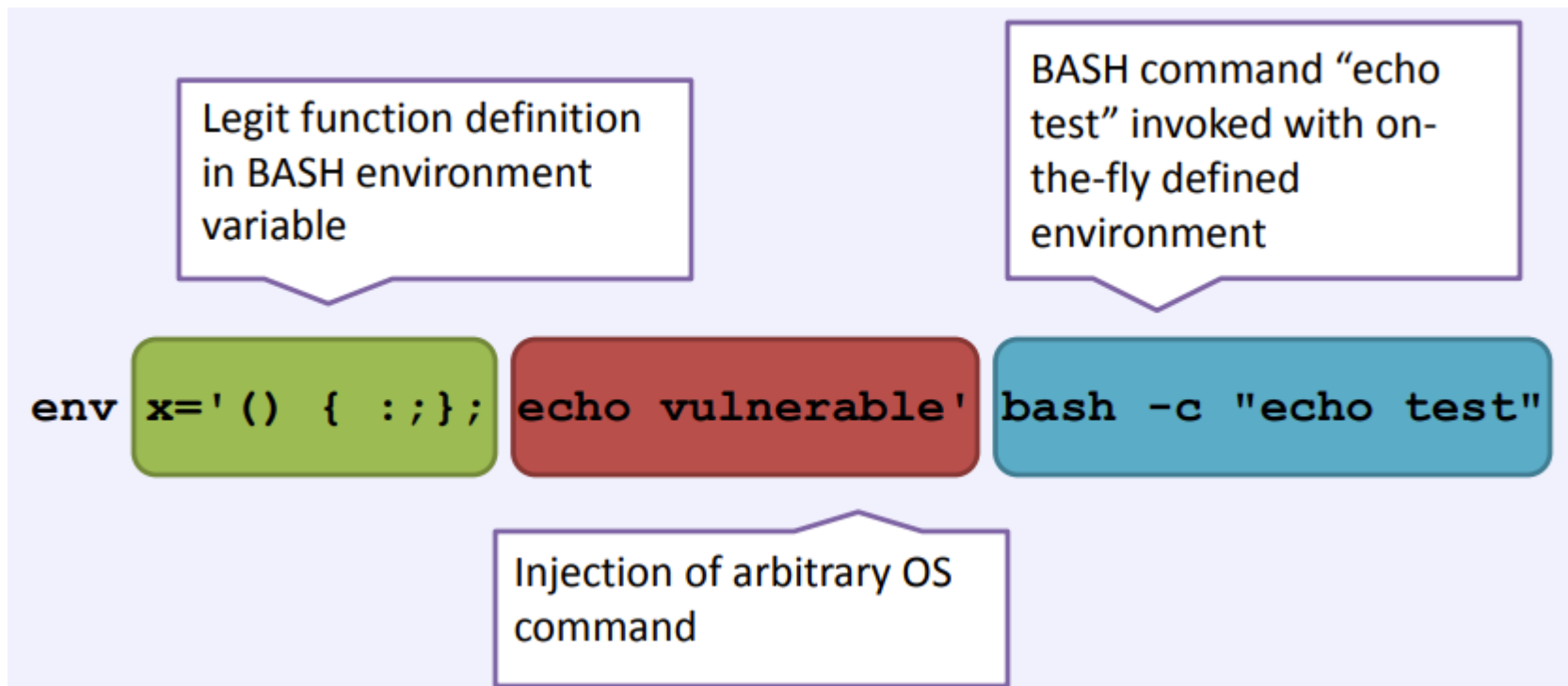
- The patch used to fix this flaw, ensures that no code is allowed after the end of a Bash function. So if you run the above example with the patched version of Bash, you should get an output similar to:

```
$ env x='() { :}; echo vulnerable' bash -c "echo this is a test"
bash: warning: x: ignoring function definition attempt
bash: error importing function definition for `x'
this is a test
```



# Shellshock vulnerability

- Shellshock is effectively a **Remote Command Execution** (RCE) vulnerability in BASH
  - REC via Apache with mod\_cgi, CGI Scripts, Python, Perl
  - OpenSSH RCE/Privilege escalation
- The vulnerability relies in the fact that BASH incorrectly executes trailing commands when it imports a function definition stored into an environment variable



## CVE-2014-7169

```
env X='() { (a)=>#} bash -c "echo date"; cat echo
```

On a vulnerable system, this would execute the command "date" unintentionally.

Source: Shellshock Vulnerability, by Tudor Enache, OWASP

# SW bugs can prove deadly

- HOW THE BOEING 737 MAX DISASTER LOOKS TO A SOFTWARE DEVELOPER (IEEE Spectrum, 18 Apr 2019)
- **Killer software**: 4 lessons from the deadly 737 MAX crashes (Fierce, Mar 3, 2020)
- [지식K] 보잉 737 MAX의 이유 있는 추락 (KBS News, 2019.04.10)
  - 보잉 737 MAX 기종을 보유하고 있는 인도네시아와 에디오피아의 여객기가 각각 지난해와 올해에 추락하면서 189명과 157명이 사망했다. 원인은 '받음각(AOA) 센서'의 **SW 오류**로 '조종특성향상시스템(MCAS)'이 가동되면서 사고가 난 것으로 드러났다.
  - 보잉 737 MAX의 디자이너들은 HW적인 문제를 SW적으로 풀어 보기 위해 MCAS를 장착한 것이다.
  - "MCAS의 설계와 운용도 문제였다."
    - MCAS가 단 하나의 받음각 센서에 의존하게 했다는 것도 문제였다고 USA TODAY는 6일 지적했다. 백업 시스템도 없었다는 얘기. 보잉은 하나의 센서가 고장이 났을 때, 오류를 일으켰을 때의 대안도 없이, 그 데이터만으로 MCAS가 작동하도록 설계했다.
    - MCAS가 문제를 일으켰을 때 조종사들이 끄는 방법도 제대로 배우지 못한 것으로 드러났다.
  - 보잉은 대책으로 우선 소프트웨어를 업그레이드하겠다고 밝혔다.
  - 이 소프트웨어 수정만으로, 추락사고의 우려가 완전히 해소될까요?  
앞으로 비행 테스트 결과를 지켜 봐야겠다. 현재 300대 이상의 보잉 737 MAX 운항이 중단된 상태이다.

# SW bugs can be dangerous

- 과기부 "KT 통신장애는 인재"...'exit' 명령어 하나 누락에 (매일경제, 2021.10.29)
  - ... 전국적으로 85분간 인터넷이 끊겼던 KT 인터넷 장애는 KT 협력업체 직원이 'exit' 명령어 하나를 제대로 안써서 발생한 것으로 드러났다. 거대 네트워크 시스템이 작업자의 사소한 실수로 전국 인터넷이 한순간에 멈춰질 수 있음을 여실히 드러낸 것이다. 과기정통부는 재발방지를 위해 통신작업 절차를 보다 명확히 하고, 사전에 명령어 오류를 잡아낼 수 있는 시뮬레이션 시스템을 구축하겠다고 답했다.
- 테슬라, **소프트웨어 결함**으로 美서 1만2천대 리콜 (연합뉴스, 2021.11.02)
  - 차량에 탑재된 소프트웨어 결함으로 전방 충돌 경고가 제대로 이뤄지지 않고 긴급 제동 장치가 갑자기 활성화되는 등 주행 시 문제가 발생할 위험이 있다는 것이 리콜 사유다.
  - 테슬라 측은 2017년부터 판매된 모델S·X·3과 2020년부터 판매된 모델 Y 중 지난달 23일 탑재된 소프트웨어가 이뤄진 차량이 리콜 대상이라고 밝혔다. 이번 리콜은 테슬라가 지난달 23일 완전자율주행(FSD) 베타 버전을 탑재한 일부 차량 대상으로 소프트웨어 업데이트를 실시한 이후에 이뤄진 것이다.
- 테슬라, **자율주행SW 결함**...미국서 차량 5만4천대 리콜 (Bloter, 2022.02.02)
  - 완전자율주행(FSD) 베타 버전 소프트웨어를 탑재한 미국 내 차량 약 5만4000대를 리콜한다. 정지 신호에서 차량이 완전히 멈추지 않는 결함 때문이다.
  - 리콜 대상은 2016~2022년 판매된 모델S·X, 2017~2022년 모델3, 2020~2022년 모델Y SUV로 총 5만4000대에 달한다. 테슬라는 소프트웨어 업데이트를 통해 롤링 스탑 기능을 해제할 예정이다.
- <역사 속의 소프트웨어 오류>에는 코드 한 줄을 빼먹어 막대한 피해로 이어진 소프트웨어 결함 사례가 가득하다.  
(출처: [유레카] 항공기 사고와 소프트웨어 사고 / 구본권)

# SW is remains insecure

## 워드프레스 SQL 인젝션 보안취약점 등 4건...주의

김민권 기자 | 승인 2022.01.10 14:24

## SK하이닉스, 조선일보 등 16개 사업자 개인정보 유출로 과징금 '철퇴'

곽중희 기자 | 승인 2022.03.25 14:32 | 댓글 0

이 외 5개 사업자의 유출 사고에 대한 해킹 방법은 에스큐엘(SQL) 인젝션, 웹셸 공격, 무작위 대입 공격으로 규명됐다. 강원도의사회 등 4개 사업자의 경우 업무상 실수로 개인정보가 외부에 공개되거나 내부 직원들에게 개인정보가 잘못 전달됐다.

취약점이 발견된 곳은 줌(Zoom) 클라이언트와, 온프레미스(on premise) 환경에서 음성과 영상 콘텐츠를 전송하는 MMR 서버인 것으로 밝혀졌다.

보안뉴스

고위험군 오류, 아파치 서버를 들끓게 하다

발견된 취약점은 CVE-2021-44790으로, 일종의 버퍼 오버플로우 취약점이며 원격 코드 실행 공격을 가능하게 한다. 그보다 조금 약한 CVE-2021-4224도...

2021. 12. 28.

보안뉴스

HP·삼성·제록스 프린터 드라이버 취약점 주의! 보안 업데이트 필수

버퍼 오버플로우로 인해 발생하는 권한상승 취약점...최신 드라이버 설치 필요  
코로나19로 재택근무 확대되면서 프린터 관련 취약점 자주 발견

2021. 7. 24.

보안뉴스

줌 클라이언트와 MMR 서버의 제로데이 취약점, 상세히 공개돼

















배경 : 문제의 취약점은 2개로 CVE-2021-34423은 CVSS 기준 9.8점짜리 버퍼 오버플로우 취약점이고, CVE-2021-34424는 CVSS 기준 7.5점짜리 메모리...

2022. 1. 21.

# Exercise (Hands-on Experience) at PentesterLab

## ● PentesterLab (<https://pentesterlab.com/exercises>)

- We Will Help You Get To The Next Level! – Try our **free exercises** or go pro

<input type="text" value="Search..."/> <span>FILTER &gt;</span>					
EXERCISE		AVERAGE TIME TO COMPLETE	DIFFICULTY	# OF USERS COMPLETED	TIER
	Python Snippet #03	< 1 Hr.		35	PRO
	Python Snippet #04	< 1 Hr.		27	PRO
	Python Snippet #05	< 1 Hr.		36	PRO
	CVE-2021-39x3x	< 1 Hr.		27	PRO
	CVE-2022-21724: JDBC RCE PostgreSQL	< 1 Hr.		7	PRO
	Recon 17	< 1 Hr.		1559	Free
	Recon 18	< 1 Hr.		1470	Free
	Recon 19	< 1 Hr.		1338	Free

# Exercise (Hands-on Experience) at PentesterLab

---

- **CVE-2014-6271/Shellshock** (<https://pentesterlab.com/exercises/cve-2014-6271/course> )
  - This exercise covers the exploitation of a Bash vulnerability through a CGI.
  - This course details the exploitation of the vulnerability CVE-2014-6271.
  
- **From SQL Injection to Shell: PostgreSQL edition** ([https://pentesterlab.com/exercises/from\\_sqli\\_to\\_shell\\_pg\\_edition/course](https://pentesterlab.com/exercises/from_sqli_to_shell_pg_edition/course) )
  - This course details the exploitation of SQL injection in a PHP based website and how an attacker can use it to gain access to the administration pages.
  
- **XSS and MySQL FILE** ([https://pentesterlab.com/exercises/xss\\_and\\_mysql\\_file/course](https://pentesterlab.com/exercises/xss_and_mysql_file/course) )
  - This exercise explains how you can use a Cross-Site Scripting vulnerability to get access to an administrator's cookies. Then how you can use his/her session to gain access to the administration to find a SQL injection and gain code execution using it.
    - This course details the exploitation of a Cross-Site Scripting in a PHP based website and how an attacker can use it to gain access to the administration pages. Then, using this access, the attacker will be able to gain code execution on the server using SQL injections.

# **Software Bugs**

**(Defects, Weaknesses, and Vulnerabilities)**



# Bugs, Defects, Weaknesses, and Vulnerabilities

---

- Improper initialization
- Side effects
- Scoping
- Operator precedence
- Divide-by Zero
- Infinite loop
- Type confusion (illegal downcasts)
- Deadlock
- Integer Overflow / Underflow
- Memory leak
  - Use-after-free
- Buffer overflow = Buffer overrun
- Time-of-check-to-time-of-use flaw
- Format string bug

# Examples of SW Bugs

```
typedef unsigned int uint;
int getmin(int *arr, uint len) {
    int min;
    for (int i=0; i<len; i++)
        min = (min < arr[i]) ? min : arr[i];
    return min;
}
```

## Improper Initialization

## Side Effects

bar = ?

baz = ?

```
if (foo == 12 || (bar = 13))
    baz == 12;
```

# Examples of SW Bugs

## Scoping

a = ?

```
int a;
void calc(int b) {
    int a = b*12;
    if (b + 24 == 96)
        a = b;
}
```

printf("a = %d\n", a);

```
for (i=0; i<numrows; i++)
    for (j=0; j<numcols; j++);
        pixels++;
```

## Control-flow (1)

If xlen=10, ylen=5,

pix[0] = ?   pix[1] = ?   pix[2] = ?

pix[3] = ?

After the 2<sup>nd</sup> for statement, x = ?, y = ?

```
int x,y;
for (x=0; x<xlen; x++)
    for (y=0; y<ylen; y++);
        pix[y*xlen + x] = x*y;
```

# Examples of SW Bugs

## Control flow (2)

```
if (isbad(cert))
    goto fail;
if (invalid(cert))
    goto fail;
goto fail;
```

...

```
L10: printf("Hello, world\n");
    goto L10;
```

## The Apple goto fail vulnerability

```
static OSStatus
SSLVerifySignedServerKeyExchange(SSLContext *ctx, bool isRsa, SSLBuffer
signedParams,
                                uint8_t *signature, UInt16 signatureLen)
{
    OSStatus      err;
    ...

    if ((err = SSLHashSHA1.update(&hashCtx, &serverRandom)) != 0)
        goto fail;
    if ((err = SSLHashSHA1.update(&hashCtx, &signedParams)) != 0)
        goto fail;
    goto fail;
    if ((err = SSLHashSHA1.final(&hashCtx, &hashOut)) != 0)
        goto fail;
    ...

fail:
    SSLFreeBuffer(&signedHashes);
    SSLFreeBuffer(&hashCtx);
    return err;
}
```

# Examples of SW Bugs

```
float x = 0.1;
while (x != 1.1) {
    x = x + 0.1;
    printf("x = %f\n", x);
}
```

```
int k = 1
int val = 0;

while (k = 10) {
    val++;
    k++;
}
printf ("k = %d, val = %d \n", k, val);
```

## Control-flow (2) -- Loop

How many times will this loop run?

# Examples of SW Bugs

## Null pointer dereference!

```
int *ptr = NULL;
printf("Value of ptr: %d\n", ptr);

int *p = 0;
*p = 1;

/* ----- */
int length;
char *buff;

scanf ("%d", &length);
buff = (char *) malloc(length+1); // always Not NULL?
strcpy(buff, "Hello World! Welcome!");
```

```
void Pointer(int *ptr) {
    *ptr = *ptr + 5;
}

main(void) {
    int num = 10;

    Pointer(&num);
    Pointer(NULL);
}
```

# Examples of SW Bugs

## Memory error

```
int i;  
char buffer[5] = "Great";  
int k;
```

## Off-by-one error

```
int i;  
int buffer[5];  
int k;  
  
for (i=0; i<=5; i++)  
    cin >> buffer[i];
```

Off-by-one error occurs when we have "<=" instead of "<" when we are checking the expression in the loop.

```
int array[] = new int[5];  
  
for (int i = 0; i <= 5; i++)  
    System.out.println(array[i]);
```



# Examples of SW Bugs

## Operator precedence Logic Error

```
node *find(node **curr, val) {  
    while (*curr != NULL)  
        if (*curr->val == val) return *curr;  
    else  
        *curr = *curr->next;  
}
```

(*\*curr*)->val

The arrow operator -> and the dot operator . bind more tightly than dereference \*, parenthesis would solve the problem.

```
int x, a, b, c, d, e, f;
```

```
a = 7; b = 6; c = 5; d = 4; e = 3; f = 2;
```

```
x = a & b + c * d && e ^ f == 7;
```

```
printf("x = %d\n", x);
```

# Examples of SW Bugs

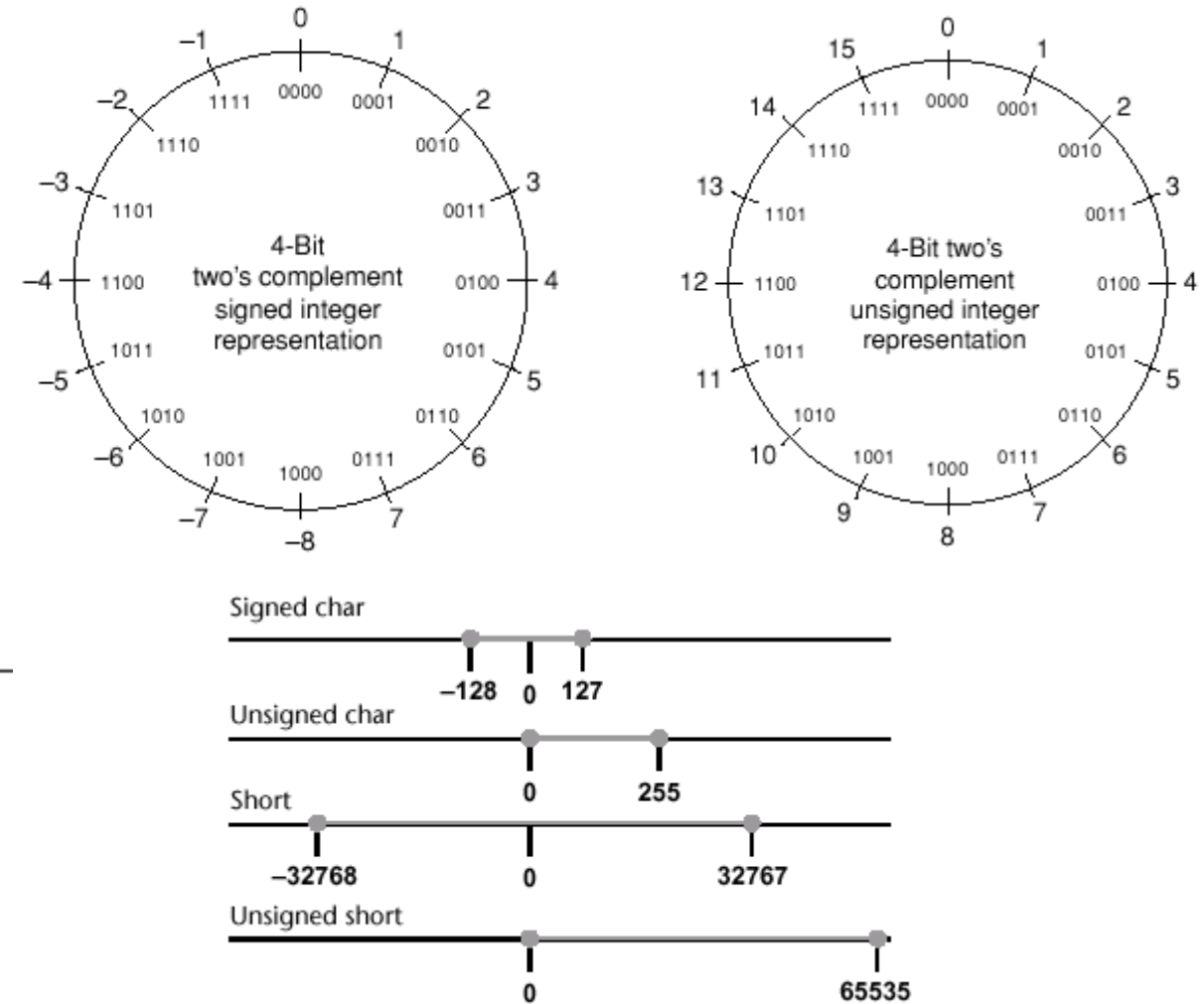
## Integer Security

```
char cresult, c1, c2, c3;
c1 = 100; c2 = 90; c3 = -120
cresult = c1 + c2 + c3;
printf("%c, %d, %c, %d\n", cresult, cresult, c3, c3);
```

```
short s1 = 32000, s2 = 1500;
s1 = s1 + s2;
printf("%h, %d\n", s1, s1);
```

```
1. unsigned int l = ULONG_MAX;
2. char c = -1;

3. if (c == l) {
4.     printf("Why is -1 = 4,294,967,295???\n");
5. }
```



Source: [https://www.codeguru.com/cpp/sample\\_chapter/article.php/c11111/Integer-Security.htm](https://www.codeguru.com/cpp/sample_chapter/article.php/c11111/Integer-Security.htm)

# Terminology

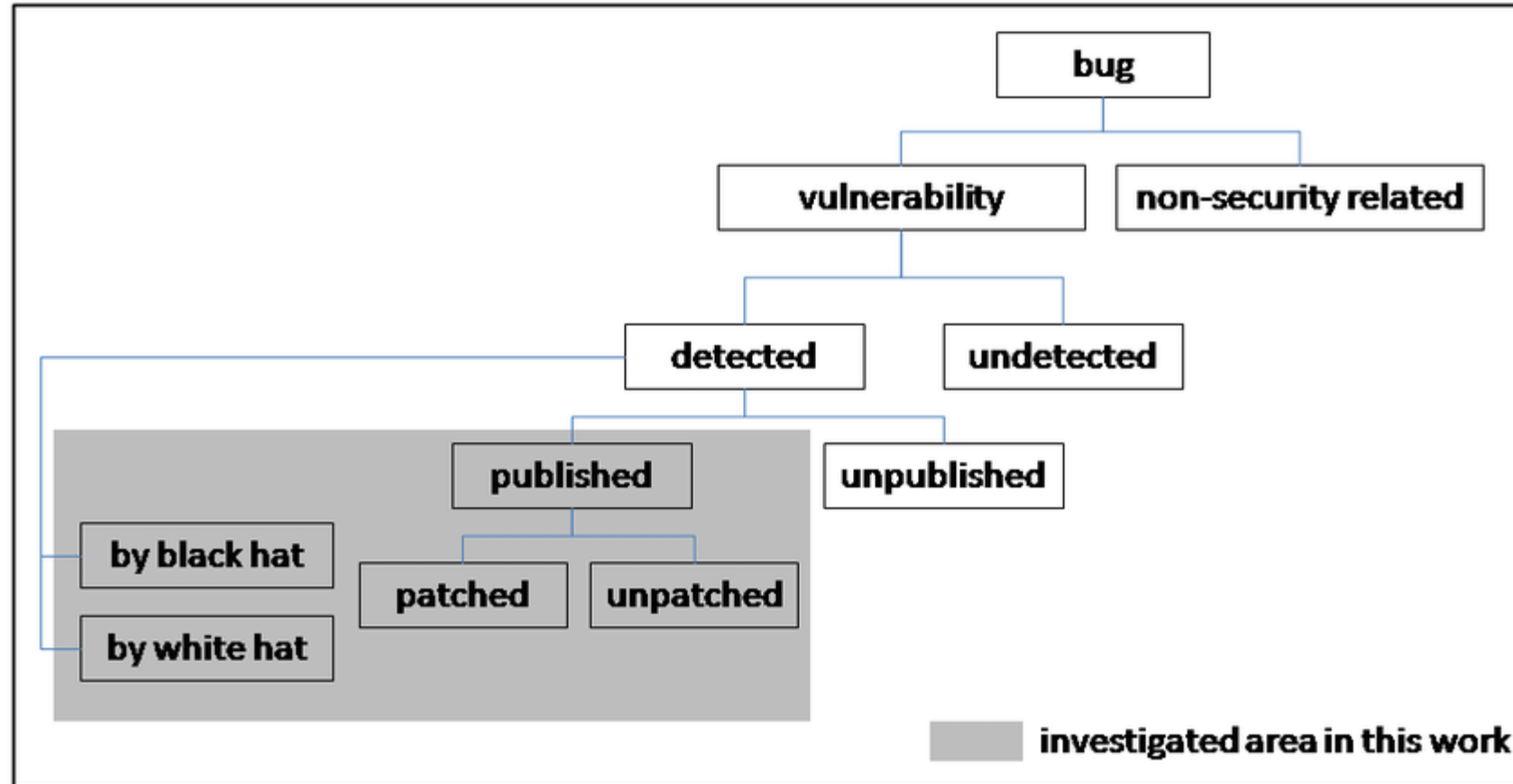
---

- **SW bug**
  - a coding error that needs to be fixed.
- **SW Weakness**
  - A weakness is caused by a bug or ill-formed data.
  - A weakness type is also a meaningful notion, as different vulnerabilities may have the same type of underlying weaknesses.
- **SW Vulnerability**
  - an instance of a weakness type that leads to a security failure.
  - It may have more than one underlying weaknesses linked by causality.
  - **A security flaw, glitch, or weakness found in software code that could be exploited by an attacker** (threat source).
- **Security Failure**
  - A violation of a system security requirement.
- **System security requirement:** Confidentiality, Integrity, Availability, Authentication, Authorization, Non-repudiation, ...

Source: The Bugs Framework (BF) - Software developers' and testers' "Best Friend", <https://samate.nist.gov/BF/Home/Terminology.html>

# SW Bug and Vulnerability

- Classification of software bugs and vulnerabilities



**Source:** Security of Open Source and Closed Source Software: An Empirical Comparison of Published Vulnerabilities, G. Schryen, AMCIS, 2009,

# Bug Bounty Program

- SW/Firmware/HW 또는 서비스 취약점을 찾아 신고하면 포상금을 지급하는 프로그램
- 자사 서비스와 제품의 신규 취약점을 신고 받고 이를 평가해 포상금을 지급하는 제도

# Bug Bounty Program (BBP)

---

- **a deal** offered by many websites, organizations and software developers by which individuals can receive recognition and compensation for reporting bugs, especially those pertaining to security exploits and vulnerabilities.
- **a sponsored, organized effort** that compensates security researchers for surfacing and reporting otherwise unknown network and software security vulnerabilities, thereby enabling the digitally connected business to manage and reduce their cybersecurity risks.
  - It rewards independent researchers and ethical hackers when they find a bug or a security vulnerability in a service/website.
- **“Organizations may also employ the use of financial incentives (also known as “bug bounties”) to further encourage external security researchers to report discovered vulnerabilities. Bug bounty programs can be tailored to the organization’s needs. ...”**
  - NIST SP 800-53 R5 ADDS VULNERABILITY DISCLOSURE PROGRAMS TO FEDERAL SECURITY AND PRIVACY CONTROLS, by Casey Ellis, Oct 7, 2020
  - a public bug bounty program is a subset of a vulnerability disclosure program

# Bug Bounty Programs

BBP / Organization	Description
Mozilla	Mozilla's bug bounty program offers a USD 3,000 reward for security critical and high severity bugs in its Firefox, Thunderbird, or related Mozilla services (Mozilla, n.d.).
GitHub	GitHub is a distributed revision control system, mostly for software code. The bounty program covers the GitHub API, GitHub Gist, and its main website github.com. Rewards range from USD 100 up to USD 5,000 (GitHub, n.d.).
Google	Google's reward program covers Google.com, Youtube.com, Blogger.com, and Orkut.com, among others, as well as browser apps and extensions developed by the search giant. Rewards start from USD 100 and go up to USD 20,000 (Google, n.d.). Further, Google runs an experimental program, in which it seeks security patches for selected open source projects. Here awards between USD 500 to USD 10,000 are offered (Google, 2013).
Samsung	Samsung's Smart TV Security Bug Bounty Program seeks submissions for its 2012-2014 Smart TV and Blu-Ray products. The bounty reward is USD 1,000 or more (Samsung, n.d.).

**Source:** Analyzing Bug Bounty Programs: An Institutional Perspective on the Economics of Software Vulnerabilities, A. Keuhn & M. Mueller, 2014

# 26 Bug Bounty Programs by the World's Biggest Tech Companies

**Source:** by Ankush Das on February 17, 2022

<https://geekflare.com/tech-companies-bug-bounty-programs/>

- A bug bounty program helps websites, services, and organizations find issues (bugs and vulnerabilities) in their offerings.
- A bug bounty program is a perfect place for security researchers or hackers to put their skills to the test.
- It gives the feel of a public competition and a run for the money with your skills.

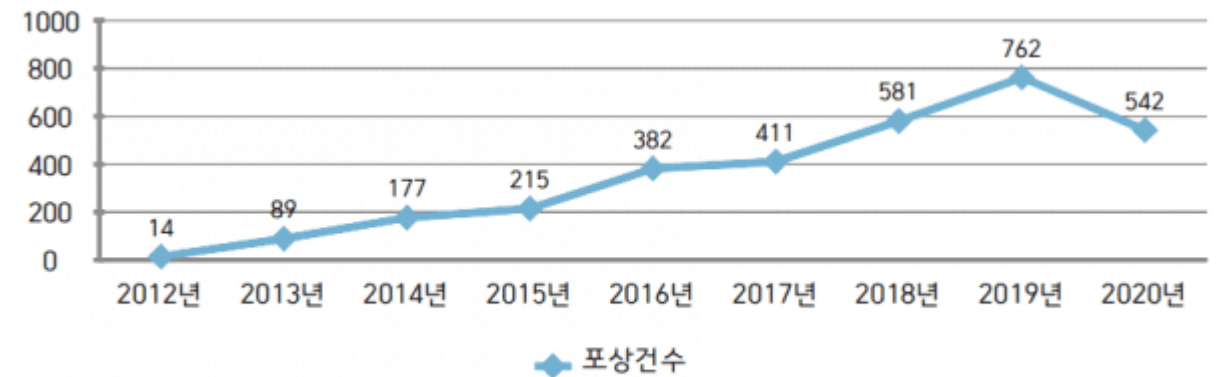
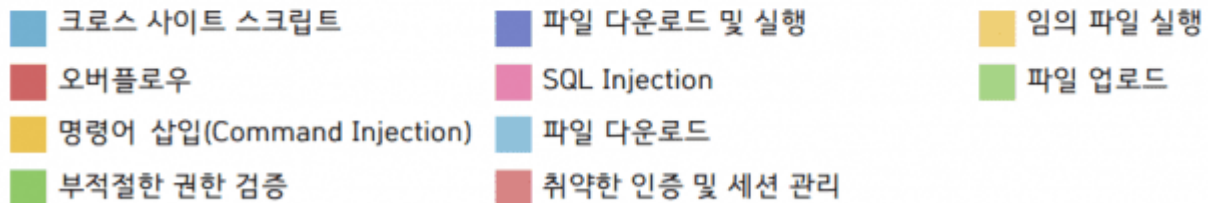
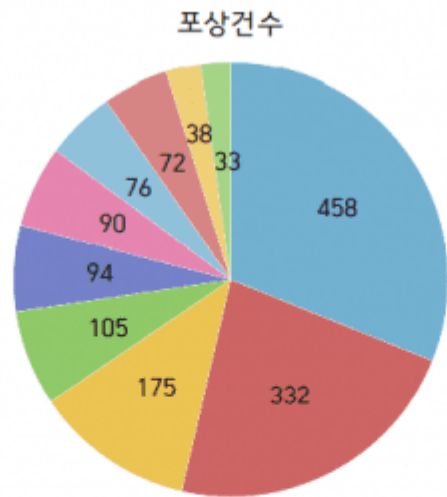
1. Apple Security Bounty
2. Meta Bug Bounty
3. Bug Hunters by Google
4. Microsoft Bug Bounty
5. Mozilla Security Bug Bounty
6. Twitter – utilizes the HackerOne platform
7. Uber – utilizes the HackerOne platform
8. Tesla's program can be found on Bugcrowd
9. Intel Bug Bounty (SW, Firmware, HW issues)
10. Tencent Security Response Center
11. Samsung Rewards Program (Mobile Security)
12. Cisco Meraki
13. Netflix Bug Bounty
14. PayPal – utilizes the HackerOne platform
15. Intuit Bug Bounty
16. ...



# SW 개발보안

## ● 국내 SW, 잘 알려진 보안 취약점도 대응 못해 – (ZDNet Korea, 2021.02.19)

- XSS·SQL 등 익숙한 유형이 다수..."SW 개발보안 인식 부족 방증"
- 국내 소프트웨어(SW)에서 발견되는 취약점 중 대부분이 이미 흔히 알려진 유형의 취약점
- KISA는 지난 2012년부터 보안 취약점 신고 포상제를 운영하고 있다. 2019년부터는 웹사이트 취약점 발굴을 위한 모의 해킹 대회 '해 더 챌린지'도 개최하고 있다.
- 프로그램의 소스코드를 작성하는 과정에서부터 취약점을 제거하는 **SW 개발보안(시큐어코딩)** 활용이 미진



[그림 1] 년도별 보안 취약점 포상건수

[그림 3] 공격유형별 그래프

# Bug Bounty Program (보안 취약점 포상제)

- 해킹 치밀해지는데...'보안 취약점 포상제' 활성화 언제? – [\(ZDNet Korea 2020.12.31\)](#)
- 보안업계 "참여 기업·포상금 지급 범주 확대 필요"
- 해킹 피해를 사전에 예방하기 위해, 보안 취약점에 포상금을 걸고 제보를 받는 '버그 바운티' 제도가 활성화돼야
- 해외에서는 구글, 애플, 마이크로소프트(MS), 페이스북 등 글로벌 IT 기업 등이 버그 바운티 제도를 운영
  - 기업과 해커를 연결해주고, 취약점에 따른 포상금을 받을 수 있게 하는 플랫폼 '해커원'도 활발히 운영.
  - 미국 국방부와 인텔, 페이팔, 퀄컴, 트위터 등 주요 글로벌 기업을 포함한 고객사 약 2천여곳과 약 100만명의 해커가 참여.
- 우리나라는 주로 한국인터넷진흥원(KISA), 금융보안원 등 공공기관에서 버그 바운티 제도를 운영
  - "외부인이 기업용 SW에 접근하기 어렵기 때문에 모의 해킹을 통한 취약점 발견이 잘 이뤄지지 않고 있고, 기업 입장에서는 이런 취약점 분석 활동이 서비스 장애를 일으킬까 우려를 한다"며 "화이트해커들이 이런 SW에 대해서도 취약점을 발견할 수 있게 하는 플랫폼을 내년 구축해 2022년부터 실제 서비스될 예정"

# Bug Bounty Program (보안 취약점 포상제)

- "웹 공격이 가능한 취약점들은 현행 취약점 신고포상제에서 포상을 받을 수 없고, 특정 SW를 실행할 때 유효한 취약점에 대해서만 제도를 이용할 수 있다"며 "기업에서 개인정보 유출 등을 유발하는 취약점은 주로 웹서비스 기반 취약점이라 버그 바운티 제도에 이런 부분이 반영되는 것이 바람직
- **웹서비스 취약점이 버그 바운티 제도에 반영되지 않는 이유는 법적 한계 때문이다.** 취약점을 찾기 위한 모의 해킹을 수행하더라도, 서비스 중인 웹사이트나 시스템에 접근 권한 없이 침입하는 것은 **정보통신망법 상 불법으로 규정돼** 있기 때문

## < 개방형 보안취약점 분석플랫폼 >



- 정부는 우수 신고사례에 대한 포상금 지급 등을 추진하는 **정보통신망법 개정을 예고**
- 정부가 디지털 뉴딜 추진 과제 중 하나로 밝힌 '개방형 보안 취약점 분석 플랫폼'





# Bug Bounty Program

- "보안취약점 찾으면 보상"...삼성SDS, 버그 바운티 판 키운다 (서울경제, 2020-12-22)
  - 삼성SDS가 화이트 해커가 보안 취약점을 신고하면 포상하는 '버그 바운티' 프로그램을 도입한다. 삼성SDS는 버그 바운티 도입을 통해 자사 제품 뿐만 아니라 다른 기업의 취약점도 발견할 수 있는 플랫폼으로 발전시켜 판을 넓히겠다는 계획이다.
  - 해킹존은 기존의 버그 바운티와 달리 여러 고객이 참여 가능한 플랫폼이라는 차별점을 가지고 있다. 빠르게 발전하는 해킹기술에 대응하기 위해 다수 사용자들의 집단지성을 통해 취약점을 찾아낼 수 있는 플랫폼으로 신고자는 포상을 받고, 기업은 버그를 발견해 보안성을 높일 수 있는 상생 방안으로 꼽힌다.
  - 신고대상은 삼성SDS의 제품 및 서비스를 포함한 8개 서비스이며 현재까지 이용자 300여명을 확보했다.
  - 유효 취약점 한 건당 최대 1,000만원의 포상금을 받을 수 있다. 이미 지난 11월부터 302건의 취약점이 보고되는 등 시범운영 기간에도 상당한 효과를 나타내고 있다.

# Bug Bounty Program

## [PASCON 2021] 삼성SDS 해킹존 “버그 바운티로 해킹공격의 집단면역 체계 구축” 강조

- “전통적 침투테스트로는 한계...글로벌 기업 대부분 버그 바운티 활용해 보안수준 강화”
- **전통적인 침투테스트**는 높은 비용으로 소수의 인력을 고용해 일정 기간동안 점검하는 시간기반 취약점 점검 방식이다. 대부분의 대한민국의 기업들은 이 침투테스트를 연 1~2회를 받고 있다.
  - 앱 하나당 평균 1.5명의 인원이 필요하며, 연간 1~2회 정도 실행하는 것이 대부분이다. 하지만 최근 애플리케이션은 상시적으로 업데이트 및 패치가 진행되기 때문에 이러한 인원과 횟수로는 취약점 관리가 어려우며, 비용 역시 많이 든다.
  - 그럼에도, 매번 점검할 때 마다 취약점이 새로 도출되거나 보안사고가 일어나는 일이 빈번하게 발생한다.
  - 그 이유는 악의적인 해커의 수는 너무나 많고 그들은 시간의 제약조차 없기 때문에 발견하기 더 어렵고 복잡한 취약점을 찾을 수 있기 때문이다.
- 그래서 최근 글로벌 기업들은 **이 한계를 뛰어넘는 방법으로 버그바운티**를 차용하고 있다.
- 삼성SDS 버그 바운티 플랫폼 ‘해킹존’
  - <https://hackingzone.net/>

“버그바운티를 통한 집단면역 체계 구축”			
			
대규모 보안전문가	높은 점검 효과	지속적인 점검	합리적 점검비용
1,000명 ↑	무제한 점검	10분 / 365일	50% ↓
다양한 보안전문가 가입 App당 수백명 점검 참여	인력,공수,범위 제한없이 다방면 대량 취약점 제보	최초 제보 평균 10분 연중 상시점검 가능	유효 취약점 제보 실적 비례 지급

출처 : 데일리시큐(<https://www.dailysecu.com>), 2021.11.13

# Bug Bounty Platform vs. Traditional Penetration services

Bug Bounty Platform	Traditional Penetration services
Access to thousands of security researchers with diverse skill set	Limited group of security researchers
Incentivised for <b>quality and severity</b> for Bugs. <b>Pay for Result model.</b>	Incentivised for <b>quantity</b> of bugs found <b>Pay of effort employed model.</b>
Time to market reduced by <b>more than 50%</b> . Follows Agile model development	Time to market higher. Services employed at the end of development lifecycle.
Offers testing for more sophisticated vulnerability scenarios	Offers testing for limited vulnerability scenarios because of limited group of security researchers
Very competitive environment. The one who reports a bug first gets the rewards	Not exposed to a competitive environment, that can affect quality of work
Pricing is based <b>Pay Per Bug (PPB)</b> Model	Pricing is based on report basis. Focus on quantity, low and medium priority bugs
Creates a culture of Openness and adoption towards information security practices	Creates a culture of fear and meeting compliance requirement.



# KISA의 보안 취약점 신고포상제

- 보안 취약점을 악용한 침해사고를 사전에 예방하고, 전문가들의 신규 취약점 발굴을 장려하기 위하여 2012년 10월부터 보안 취약점 신고포상제를 운영하고 있습니다.
- **신고대상 취약점** : '소프트웨어'에 대한 보안 취약점으로 최신버전의 소프트웨어 영향을 줄 수 있는 보안 취약점(제로데이 취약점)
- **평가 및 포상 일정** : 분기별 평가를 실시하여 포상금을 지급(3, 6, 9, 12월에 평가 및 포상 실시)
- **주의사항** : 실제 서비스 중인 웹사이트나 시스템(서버, 네트워크, 보안장비 등)에 정보통신서비스 제공자의 동의를 받지 않고 정당한 접근권한 없이 또는 허용된 접근 권한을 넘어 취약점을 발굴하는 행위는 **정보통신망 침입행위로 간주될 수 있으므로 평가 및 포상대상에서 제외됨은 물론, 법에 의해 처벌받을 수 있습니다.**
  - («정보통신망 이용촉진 및 정보보호 등에 관한 법률» 제48조제1항, 제71조제1항 제9호 및 제2항 참고)

출처: <https://knvd.krcert.or.kr/rewardExplain.do>

KISA 한국인터넷진흥원



상세검색

취약점 소개

신고포상제

취약점 신고/조회

기업 및 기관

취약점 정보 공유

알림마당

신고포상제

보안 취약점 신고 포상제

공동 운영 제도

Home > 신고포상제 > 보안 취약점 신고포상제

○ 보안 취약점 신고포상제

보안 취약점 신고포상제

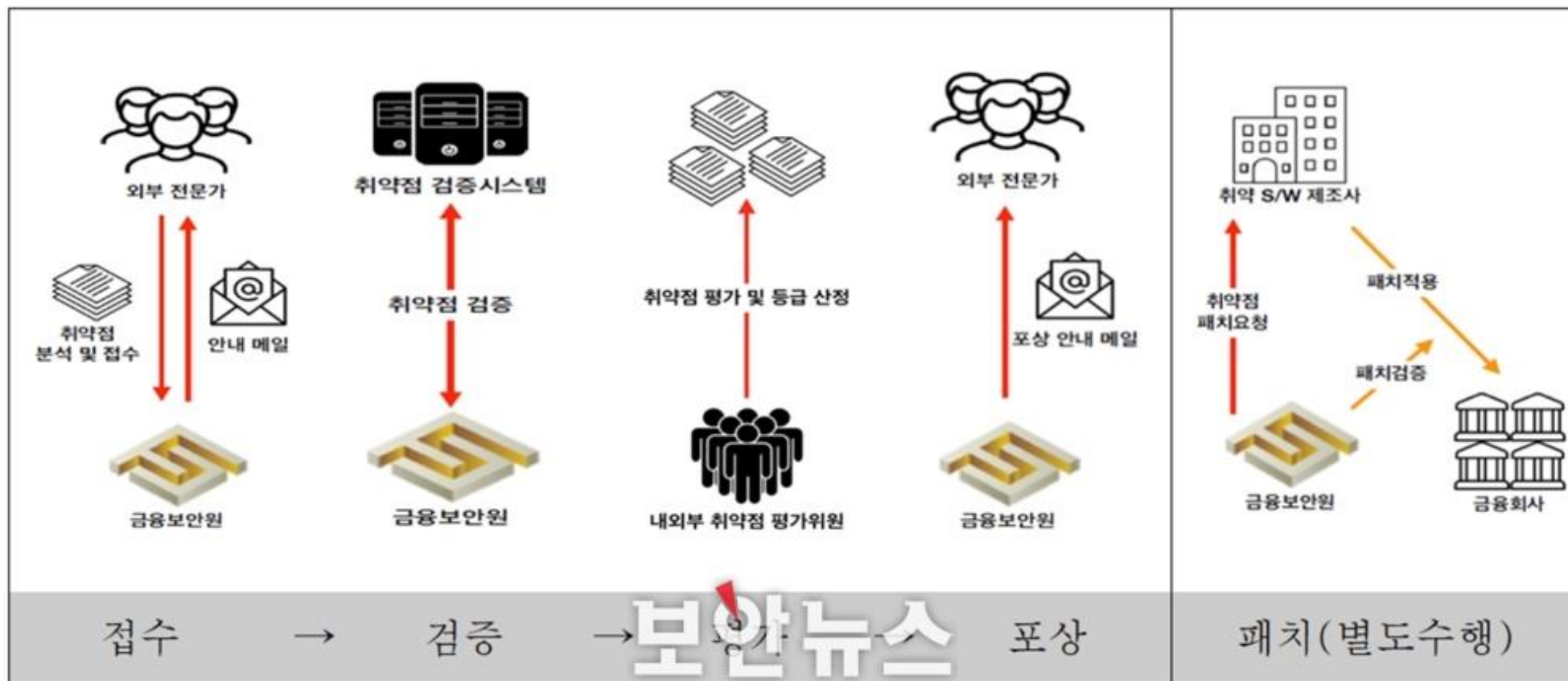
보안 취약점 신고포상제란?



# 금융보안원: Bug Bounty Program

## 금융보안원, 2021년 금융권 버그바운티 실시

### [2021년 금융권 버그바운티 운영 절차]



신고 대상은 국내 금융회사가 전자금융 소비자에게 제공하는 **Non-ActiveX 소프트웨어의 신규 보안 취약점**이다.  
단, 정보통신망법 등 관계 법령을 위반해 신고한 취약점은 신고 대상에서 제외된다.

“보안에서 100% 완벽한 것은 없다.  
특히 금융권의 IT 개발·운영 아웃소싱이 증가하고  
비대면·온라인 거래가 일상화되는 상황에서  
금융회사와 금융소비자가 이용하는  
**소프트웨어에 대한 보안은 필수**”



# Bug Bounty Program

- **네이버, 보안취약점 제보 포상제 독립 운영** – (ZDNet Korea, 2019.09.03)  
KISA 공동운영 종료...“독자적 보안 관리 역량 갖춰”
- **네이버, ‘2021 네이버 버그바운티 어워드’ 시상식 진행** -- (IT Daily, 2022.02.21)
  - 지난 한 해 166명 참여, 총 1억 1,600만 원 포상금 지급
- **구글 보안전문가, 페이스북 메신저 도청 취약점 찾아 6천달러 지급받아** – (데일리시큐, 2020.11.23)
  - 페이스북 메신저 버그로 인해 통화 전 도청 가능해
  - 최근 페이스북이 안드로이드용 메신저 앱에 존재하는 버그를 패치
  - 보안연구원 실바노비치는 해당 문제를 보고하고 6천달러의 바운티를 받았으며, 이는 페이스북에서 현재까지 가장 높은 세가지 버그바운티 중 하나다. 구글 연구원은 그녀가 해당 보상금을 GiveWell이라는 비영리 단체에 기부한다고 말했다
- **확대되는 민간 '버그바운티' 시장...주도권 경쟁 '활활'** -- (아이뉴스 24, 2021.06.13)
  - 취약점 진단 수요 높아 버그바운티도 활발...삼성SDS·엔키·티오리·파스텔플래닛 등

# Naver Bug Bounty Program

네이버 버그 바운티 프로그램은 네이버 서비스의 취약점을 조기에 찾아 사용자들에게 안전한 서비스를 제공하기 위한 프로그램입니다.

전 세계의 보안 전문가들의 도움으로 네이버 서비스의 보안 취약점을 빠르게 찾아 고치고, 보안 전문가들의 노력에 적절한 포상을 지급함으로써 네이버 서비스를 더욱 안전하게 만드는 일을 장려합니다.

출처: <https://bugbounty.naver.com/ko/>

하단의 웹사이트에서 발생하는 취약점들을 대상으로 합니다. (대상 사이트 외의 제보는 포상 대상이 아닙니다.)

- 네이버페이 : pay.naver.com 및 \*.pay.naver.com
- 네이버 블로그 : blog.naver.com 및 \*.blog.naver.com
- 네이버 카페 : cafe.naver.com 및 \*.cafe.naver.com
- 네이버 영화 : movie.naver.com, 및 \*.movie.naver.com. 단, ticket.movie.naver.com은 제외
- 네이버 쇼핑 : shopping.naver.com 및 \*.shopping.naver.com
- 네이버 바이브 : vibe.naver.com 및 \*.vibe.naver.com
- 네이버 웹툰 : comic.naver.com 및 \*.comic.naver.com
- 네이버 시리즈 : series.naver.com 및 \*.series.naver.com
- 네이버 시리즈온 : serieson.naver.com 및 \*.serieson.naver.com
- 네이버 예약 : booking.naver.com 및 \*.booking.naver.com
- 네이버 파파고 : papago.naver.com 및 \*.papago.naver.com
- 네이버 회원 : nid.naver.com 및 \*.nid.naver.com
- 네이버 메일 : mail.naver.com 및 \*.mail.naver.com
- 네이버 주소록 : contact.naver.com 및 \*.contact.naver.com
- 네이버 메모 : memo.naver.com 및 \*.memo.naver.com
- 네이버 지도 : map.naver.com 및 \*.map.naver.com
- 네이버 플레이스 : m.place.naver.com, \*.place.naver.com, smartplace.naver.com 및 \*.smartplace.naver.com
- 그리고 위 서비스에서 사용하는 API Gateway(apis.naver.com)도 포함합니다.

하단의 최신버전 어플리케이션에서 발생하는 취약점들을 대상으로 합니다.

- MYBOX 탐색기

## IoT 디바이스

- NL-\*KR 모델명의 국내 출시 클로바 디바이스(Clock, Lamp, Friends, Friends Mini) 및 연동 어플리케이션
- ※ WAVE(NL-S500), U+ 공급모델(NL-\*KRL) 디바이스와 3rd Party 도메인은 제외

# Naver Bug Bounty Program

## 3. 포상

취약점	예시	포상금액
Account takeover	Authentication Bypass	USD ~\$20,000
Remote Code Execution	Ability to send packets containing arbitrary system call to server	USD ~\$5,000
Full access to filesystem or database	SQL Injection	USD ~\$3,000
Execute code on the client	XSS	USD ~\$500
Logical flaw Bugs	Sensitive actions, Purchase Bypass	USD ~\$500
Other security vulnerabilities	Information Leakage, CSRF, SSRF	USD ~\$300

- CSRF 취약점은 다음의 기능만 대상으로 합니다 : 네이버 회원 정보, 네이버 카페 관리, 네이버 블로그 관리
- 포상금은 취약점의 위험도와 보고서 등을 종합적으로 검토하여 주어집니다.
- 사용자의 개입이 많이 필요한 경우 포상 금액이 줄어들 수 있습니다.
- 새로운 방식의 공격 기법이나 버그타입에 대해서는 좀 더 많은 포상금이 주어질 수 있습니다.

# Whale Security Bug Bounty Program

- Naver Corporation launches the Whale Security Bug Bounty Program to encourage security researchers in helping us to find and fix security vulnerabilities on Whale and to reward their efforts spent to make our product secure.
- Naver focuses on bugs in the latest version of Whale browser.
  - The bugs must be reproducible on the latest version in the time of reporting
  - Bugs in third party libraries used by only Whale (not Chromium) are eligible
  - Bugs in synchronization are eligible
- <https://bugbounty.whale.naver.com/ko/>

Vulnerability	Examples	Maximum Rewards
Sandbox Escape	File system access, Arbitrary external program execution	USD \$7,500
Remote Code Execution	Arbitrary code execution using memory corruption bugs without sandbox escape	USD \$5,000
Same Origin Policy Violation	Universal XSS	USD \$4,000
Information Leak	Arbitrary memory read	USD \$2,000
Protection Bypass	Bypassing malicious program download protection, Bypassing malicious page blocks	USD \$750
Spoofing	Address bar spoofing, Referer spoofing	USD \$500
Built-in Extensions Vulnerabilities	XSS in built-in extensions	USD \$500

# Samsung Mobile Security Rewards Program



<https://security.samsungmobile.com/rewardsProgram.smsb>

**We appreciate your interest and intention to help improve the security of Samsung Mobile products.**

We take security and privacy issues very seriously; and as an appreciation for helping Samsung Mobile improve the security of our products and minimizing risk to our end-consumers, we are offering a rewards program for eligible security vulnerability reports.

Please check below for more information on guidelines and eligibility for Samsung Mobile Security Rewards Program.

## Conditions for rewards qualification:

1. Security vulnerability report ("Report") must be applicable to eligible Samsung Mobile devices (including smartphones, tablets and wearable devices listed below), services, applications developed and signed by Samsung Mobile, or eligible 3rd party applications developed for Samsung Mobile.

- Eligible Samsung Mobile Devices in their latest available Android version and firmware:

*Galaxy Fold, Galaxy Fold 5G, Galaxy Z Fold2, Galaxy Z Fold2 5G, Galaxy Z Fold3 5G, Galaxy Z Flip, Galaxy Z Flip 5G, Galaxy Z Flip3 5G, W20 5G, W21 5G*

*Galaxy S series (S8 Lite, S9, S9+, S10, S10+, S10e, S10 5G, S10 Lite, S20, S20 5G, S20+, S20+ 5G, S20 Ultra, S20 Ultra 5G, S20 FE, S20 FE 5G, S21 5G, S21+ 5G, S21 Ultra 5G, S21 FE 5G, S22, S22+, S22 Ultra)*

*Galaxy Note series (Note9, Note10, Note10 5G, Note10+, Note10+ 5G, Note10 Lite, Note20, Note20 5G, Note20 Ultra, Note20 Ultra 5G)*

*Galaxy A series (A6, A6+, A7 (2018), A8 Star, A8s, A9 (2018), A10, A10e, A10s, A20, A20e, A20s, A30, A30s, A40, A50, A50s, A60, A70, A70s, A80, A90 5G, A01, A01 Core, A11, A21 A21c A31 A41 A51 A51 5G A71 A71 5G A02 A02c A12 A22 A22 5G A22e 5G A32 A32 5G A42 5G A52 A52 5G A52c 5G A72 A82 5G A03 A03c A03 core A13 5G)*

# Samsung Mobile Security Rewards Program

## ● Security Reporting

If you have identified a potential security vulnerability in any Samsung Mobile product or software, please report it here [Create Report](#)

Please carefully read the reporting guidelines below and [Samsung's security risk classification criteria](#) prior to reporting.

The severity level is classified to 5 levels (Critical, High, Moderate, Low, and No (or Very Low) Security Impact) depending on the security risk and impact

## Required Information

- Firmware version of affected products
- Description of potential vulnerability as detail as possible
- Steps to reproduce the issue
- Proof of Concept (PoC) (including video, image, APK, sample code, etc.)
- Expected correct behavior or workaround
- Disclosure plans, if any

### Classification

### Description / Impact

#### Critical

- Remote arbitrary code execution in a privileged process, bootload
- Arbitrary code execution in the TEE or SE (Hardware-based security)
- Remote secure boot bypass
- Unauthorized access to hardware-protected key
- Remote permanent denial of service (device inoperability; completely permanent or requiring re-flashing the entire operating system or a factory reset)
- Remote bypass of user interaction requirements for any developer, security, or privacy settings
- Remote bypass of user interaction requirements on package installation or equivalent behavior
- Unauthorized access to data secured by the SE (Hardware-based security solution)



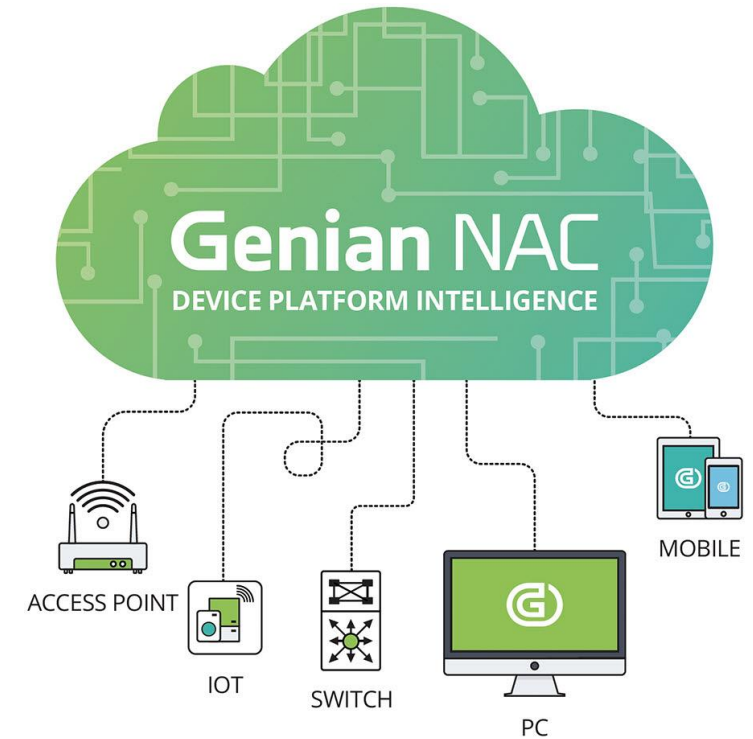
# Genians Bug Bounty Program

## Genians Bug Bounty Program

지니언스의 버그바운티 프로그램은 국내외의 보안전문가들의 도움으로 제품 및 서비스의 보안 취약점을 빠르게 찾아 고치고, 보안전문가들의 노력에 적절한 보상을 지급함으로써 제품을 더욱 안전하게 만드는 일을 장려합니다.

신고서 다운로드
취약점 접수

<https://genians.co.kr/resources/bug-bounty/>



주요 글로벌 기업은 자사 제품 및 서비스의 취약점 발굴 및 보안강화를 위해 버그바운티를 운영 중에 있으며, 국내 기업들도 독립적인 버그바운티를 자체적으로 운영중입니다. 또한 국내 NCSC, KISA 는 취약점을 악용한 침해사고 예방을 위해서 보안 취약점 신고포상제를 운영하고 있으며, ...




















## ● 지니언스의 허용 범위: Genian NAC 4.x, Genian NAC 5.x

- '지니안 NAC(Genian NAC)'는 내부 정보 보호 체계를 수립하여 내부 자산과 사용자를 보호하고 기업 자원을 안전하게 사용할 수 있도록 지원하는 유무선 네트워크 접근제어(NAC: Network Access Control) 솔루션

# Public Bug Bounty Program List

## Public Bug Bounty Program List (<https://www.bugcrowd.com/bug-bounty-list/> )

- The most comprehensive, up to date crowdsourced list of bug bounty and security vulnerability disclosure programs from across the web curated by the hacker community.
- This list is maintained as part of the [Disclose.io](https://disclose.io) Safe Harbor project.

Program Name	New	Bug Bounty	Swag	Hall of Fame	Submission URL	Safeharbor
0x Project						
1Password Game						
23andme						
24sessions						
ARM mBed						
AT&T						
Abn Amro						



# Summary

---

- **Software is ubiquitous**
- **There are so many buggy programs**
- **Bad bugs can prove deadly**
- **Examples Software Bugs/Flaws**
  - Improper initialization
  - Side effects
  - Scoping
  - Control flows
  - Integer security
  - Null pointer dereference
  - Operator precedence logic error
- **Bug Bounty Programs**