

2019 SAMSUNG SECURITY TECH FORUM

TRUST
With SAMSUNG



Samsung
Security Tech
Forum

SAMSUNG
Research

TRUST
With SAMSUNG

보안 컨설팅으로부터 얻은 교훈과 모범사례

박세준

CEO
Theori



\$ whoami



Brian Pak

박세준

CEO & Researcher



Carnegie Mellon University (CMU)

- B.S. in Computer Science ('11)
- M.S. in Computer Science ('12)



PPP, Founder



Kaprica Security, Co-founder

- 2011~2015

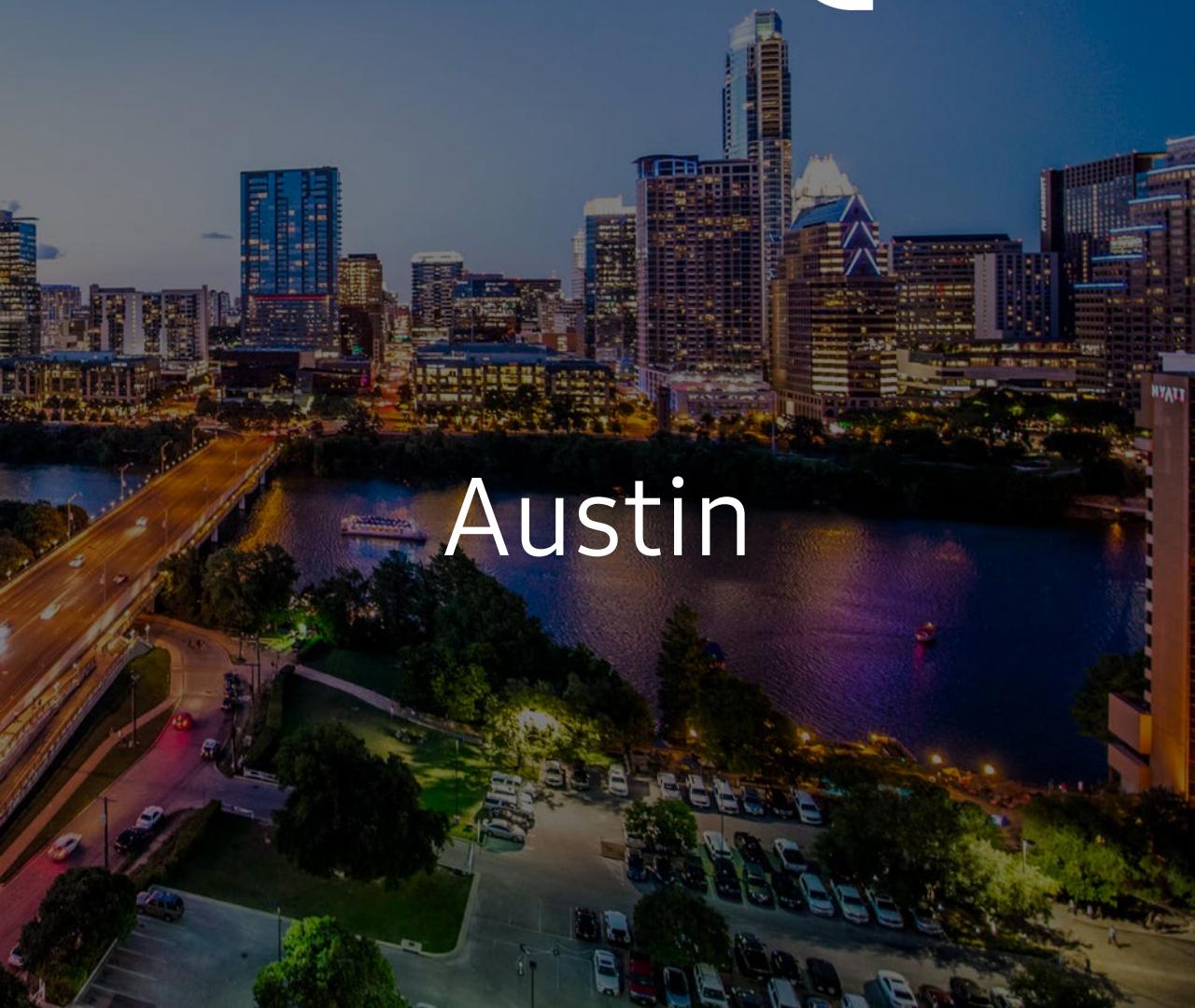


Theori, Co-founder

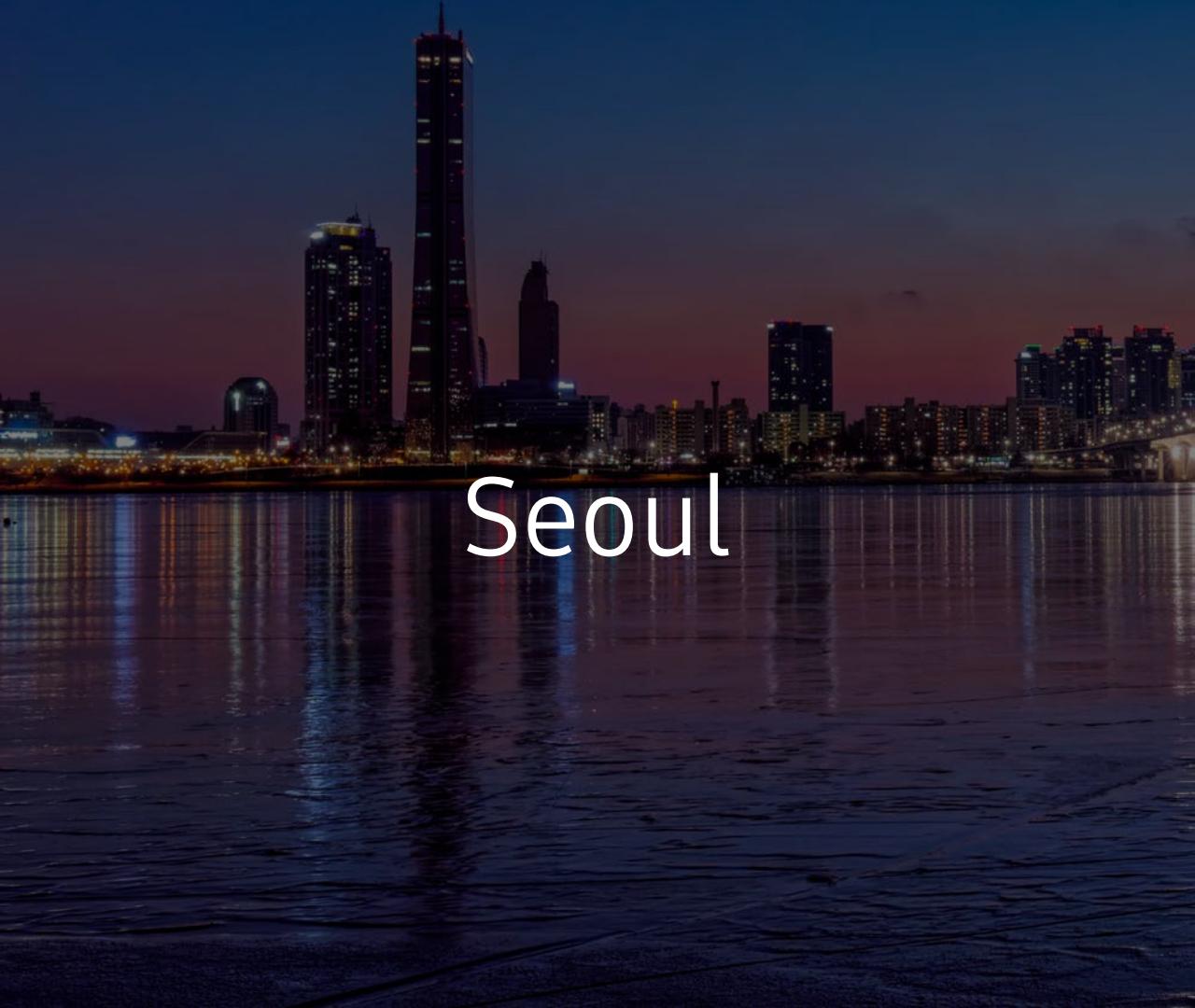
- 2016~



THEORI



Austin

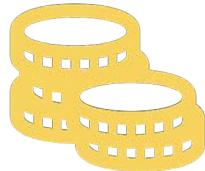


Seoul

Customers



Global
Start-up / Corporate



Cryptocurrency
Exchange



Financial
Institute



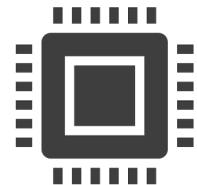
Game
Publisher



Blockchain as a
Service (BaaS)



Operating System
and Browsers



Electronics
Manufacturer

Offensive Security Research



Offensive Security Research



Proactive

vs. Reactive



사후약방문 대신 사전에 대비

침투 가능 경로 조사 및 차단

취약점 점검 및 패치



미리 더 안전한 시스템 설계

Offensive Security Research



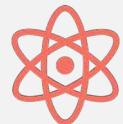
Adversarial

vs. Friendly, Internal, Defensive



공격자 관점에서 바라보기

보편적으로 보안이란 방어 관점에서 생각
실제 공격 시나리오 대비 필요



최신 공격 기법 및 우회 기법 등 적용

Offensive Security Research



Creative

vs. Rule-based, Checklist



각 타겟에 따른 맞춤형 공격

기존 사용하던 툴이나 룰 기반 점검의 한계
각기 다른 환경에 빠르고 유연한 적응 필요



새로운 공격 및 방어 기법 연구

Offensive Security Research



Assistive

+ = Risk Assessment, Risk Management



공격은 최선의 방어

공격 시도를 통해 알게된 내용 전달

효과적인 방어를 위한 현실적인 피드백



적절한 위험 평가 후, 위험 관리에 직접적인 도움

Offensive Security Research

장점

- 공격자의 관점에서 사전 보안성 검증 가능
- 최신 기술 및 기법을 연계하여 점검 가능

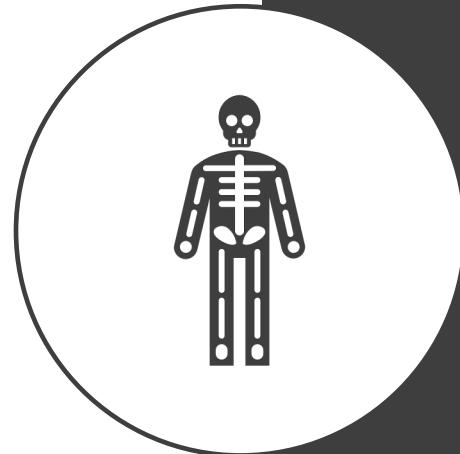
한계

- 결국 보안은 성공적으로 방어하는 것이 목표
- 취약점을 아는 것만으로는 방어할 수 없음 => 견고한 설계와 대응 필요

Threat Model

Threat Model

- 가장 원초적인 질문
 - 나를 공격할 사람 혹은 집단이 **누구**인가?
 - 개인, 단체, 국가, ...
- **불특정 다수**를 노리는 공격의 가능성
 - 예) WannaCry, GandCrab 등 랜섬웨어
 - 특정 타겟팅한 것은 아니지만, 누구나 영향을 받을 수 있음



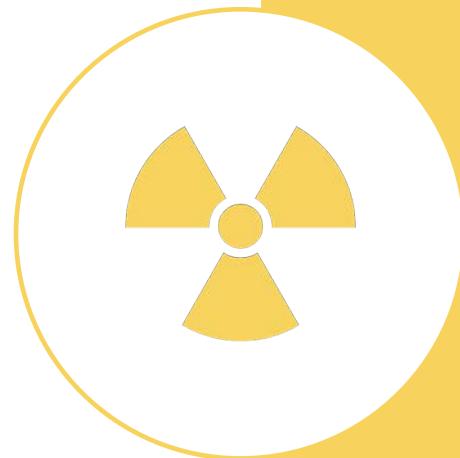
Threat Model

- 공격을 통해 얻으려고 하는 것은 무엇인가?
 - 금전, 명예 악명, 민감한 정보, ...
- 가장 중요하게 생각하고 지켜야 할 자산에 대한 정리
- 내가 공격자라면 무엇을 탈취할 것인가?



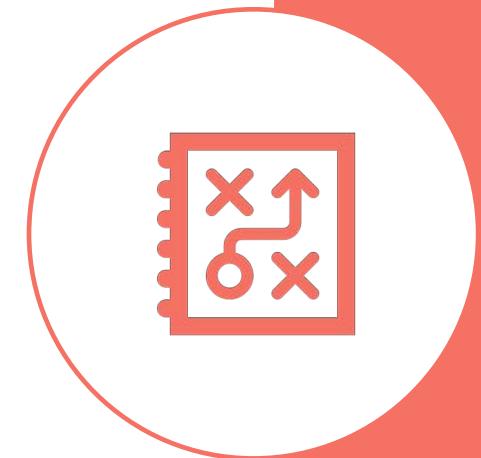
Threat Model

- 무엇이 노출 (exposed) 되어 있는가?
- 나의 약점이 무엇인가?
- 약점을 효율적으로 파악 및 관리하고 있는가?
 - 왜 여전히 약점이 존재하는지?



Threat Model

- 내가 **공격자**라면 어떻게 침투할 것인가?
- 침투에 **방어**를 위한 **설계**는 되어 있는가?



The background of the slide is a dark, moody photograph of a mountainous landscape. In the foreground, there are large, rugged rock formations with intricate textures and shadows. The middle ground shows more mountain peaks, some with snow or light-colored rock. The sky is filled with heavy, dark clouds, creating a somber and contemplative atmosphere.

Risk Assessment + Risk Management

Risk Assessment

- 노출된 위협에 대한 수치화
 - 예) 데이터 손실 및 유출, 금전적 피해 등



Equifax Settlement
\$500,000,000+
2017



FedEx
\$400,000,000+
2017



Capital One
\$100,000,000+
2019

Everything is a TRADE-OFF

Be HONEST to yourself!

Risk Management

- 그렇다면, 이러한 위협을 어떻게 관리해야 할까?
- 현실적인 위협 모델 작성
- 각 위협 경로마다 현재 관리하는 방법을 나열
 - 주기적으로 관찰 및 조정



Risk Management

- 보안에 적절한 **투자** 필요
 - Risk Assessment 결과에 기준하여 책정
- 보안 제품 및 서비스가 아니어도 **보안성 향상** 가능
 - 예) 기존 Windows 7 기기들을 최신 Windows 10 운영체제로 업그레이드
- 단, 보안 제품을 추가한다고 해서 꼭 보안성이 향상되는 것은 아님
 - 예) 보안 장비 및 소프트웨어 (망연계, DRM 관리 등) => **새로운 공격 벡터**





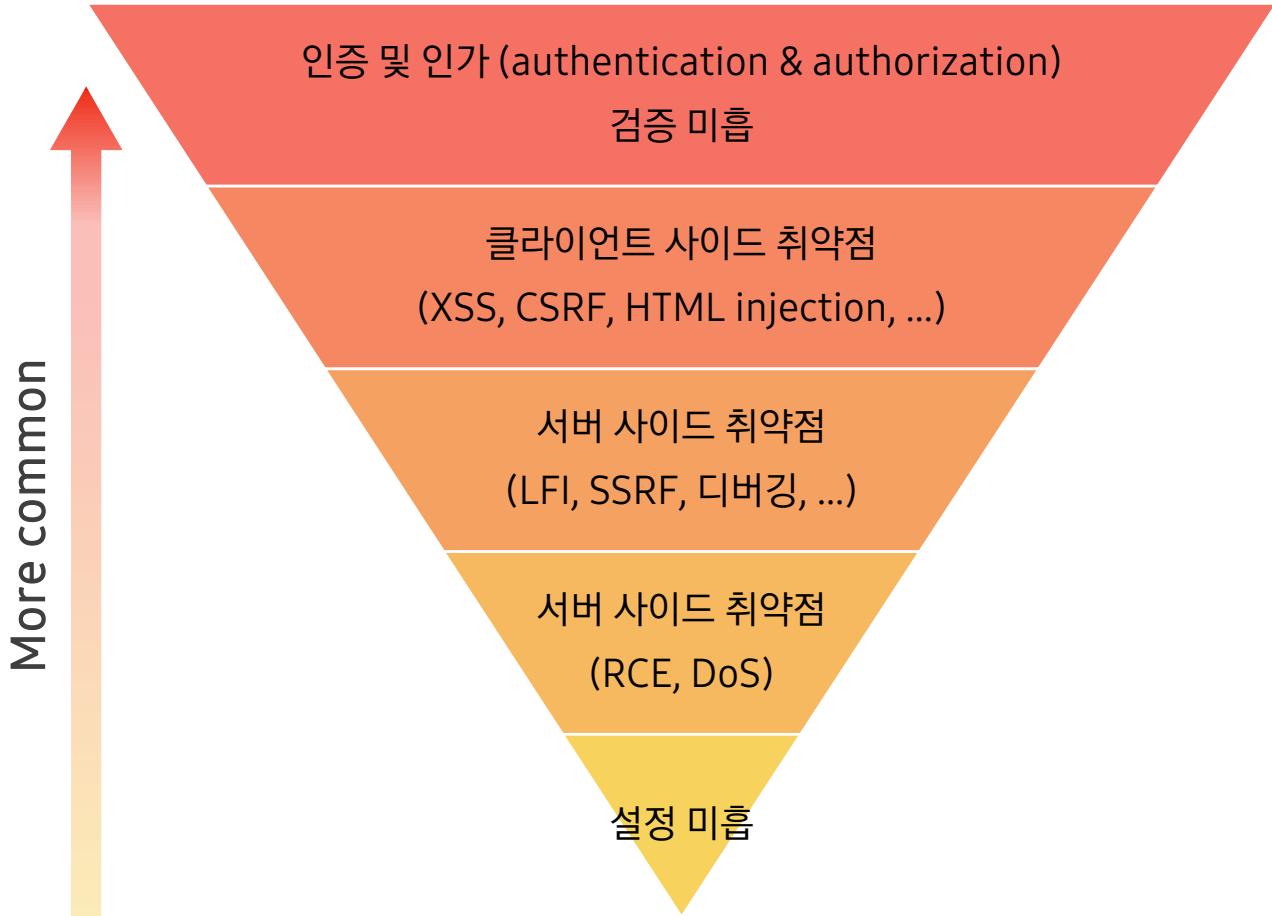
Case Study

Case Study

웹 서비스

- 주요 공격 기법
 - 웹 클라이언트/서버 취약점 (XSS, CSRF, SQLi, SSTI, SSRF 등)
- 자주 발견되는 취약점
 - 사용자 인증 미흡 (IDOR), 관리자/DevOps 인터페이스 노출
- 위협 받는 것 – 고객 데이터, 회사 자산, 사내 보안 등

Case Study – 웹 서비스



- API 서버
 - 사용자 인증 미흡
 - 사용자 권한 검증 및 분리 미흡
 - CORS 설정 미흡
- 클라이언트 (JavaScript)
 - XSS 필터링 부재
 - .html 렌더링
- 앱 서버 (Django, RoR, nodejs, ASP.NET)
 - 관리자/디버깅 인터페이스 노출
 - S3 버킷 권한 설정 미흡
 - Host 체크 미흡

Case Study – 웹 서비스

예제 1: SSO 로그인 Host 검증 미흡으로 인한 관리자 권한 획득

C#

```
private void HandleAdminRequest(GoogleTokenModel model) {  
    var username = model.email.Split('@')[0];  
    var account = _g_accounts.FindAdminAccount(username);  
    if (account == null || string.IsNullOrEmpty(account.UserName)) {  
        account = new APP.Models.createAdminAccount()  
        // ...  
    }  
}
```

패치권고: Google OAuth 인증 이후, 허용된 도메인만 허용

Case Study – 웹 서비스

예제 2: .html (innerHTML) 렌더링으로 인한 XSS 발생

```
Vue.js           jQuery

<p>
  Your notes:
  <span v-html="notes"></span>
</p>

// ...
if (notificationNum == 0) {
  notificationHtml += '<li class="none"></li>';
}
$('.notification_list').html(notificationHtml);
```

패치권고: innerText 사용 혹은 사용자 제어값 필터링/검증

Case Study – 웹 서비스

예제 3: whereRaw SQL Injection

Nodejs

```
const result = await knex
  .select(knex.raw(selectClause))
  .select(selectColumns)
  .from(table)
  .where(whereClause)
  .whereRaw(`DATE(uploaded_time) >= '${fromTime}'`) // !!!
  .whereRaw(`DATE(uploaded_time) <= '${toTime}'`) // !!!
// ...
```

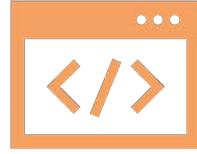
패치권고: Prepared statement 활용 및 파라미터 검증



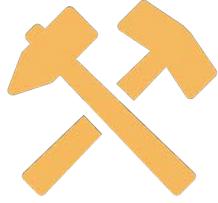
인증과 인가의
올바른 이해와 적용



사용자가 변경 가능한
데이터 검증 및 필터링

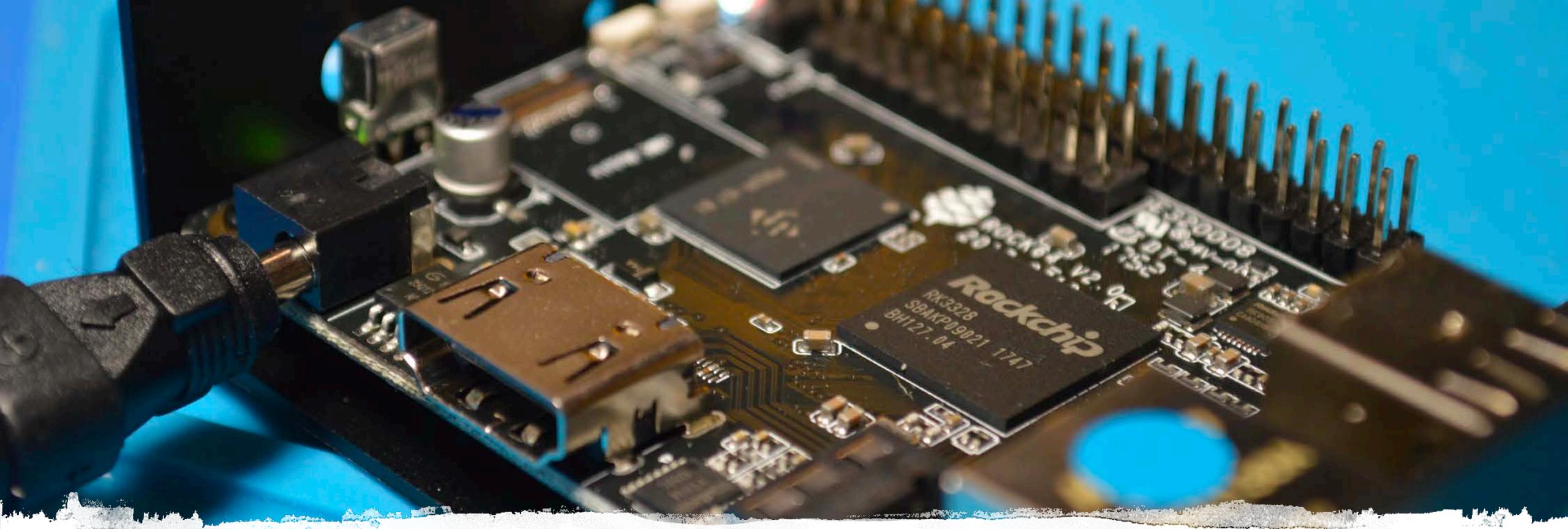


웹 프레임워크
적극 활용



각종 설정 재검토
및 지속적인 관리

Lessons Learned – 웹 서비스

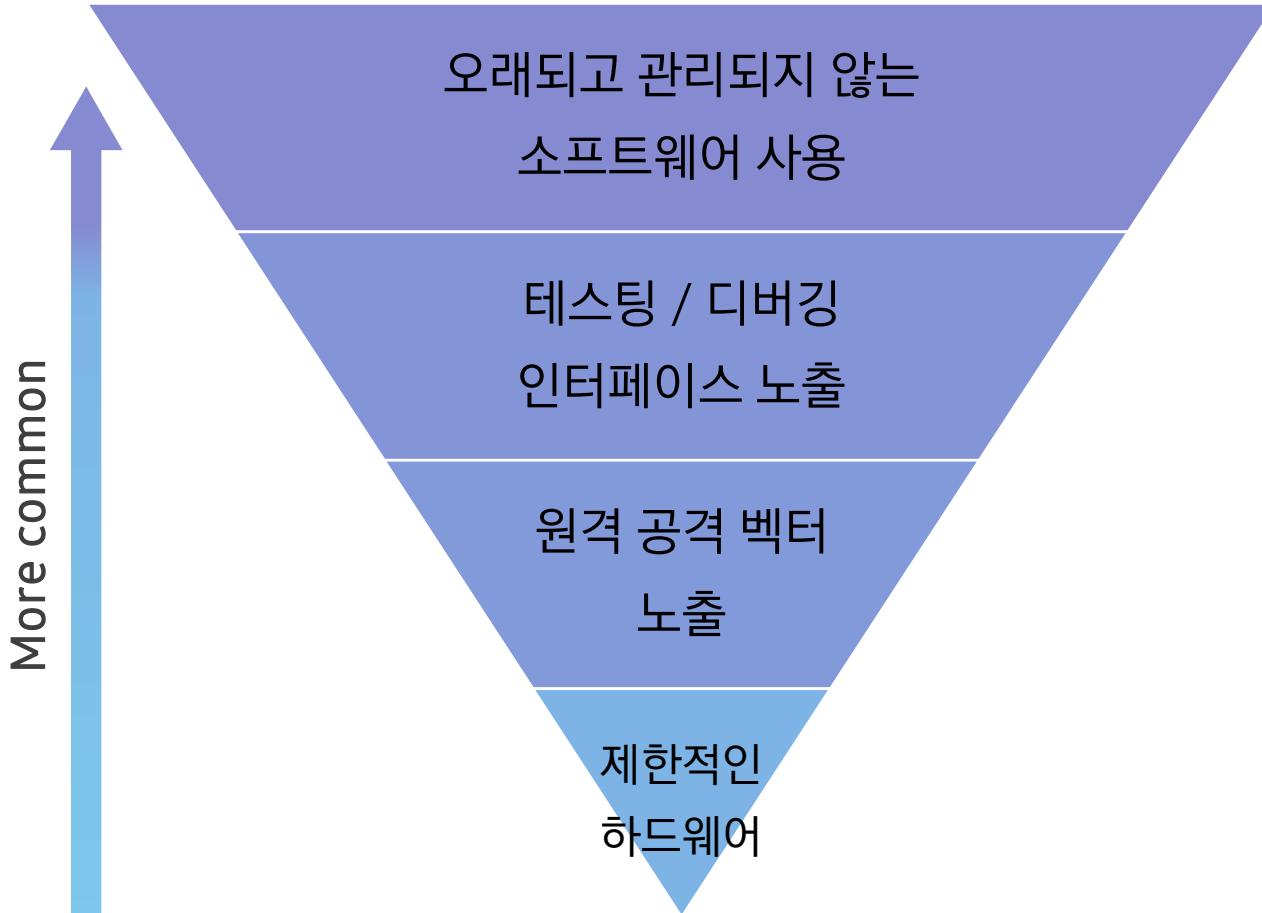


Case Study

임베디드 시스템

- 주요 공격 기법
 - 소프트웨어 취약점, 하드웨어 디버깅
- 자주 발견되는 취약점
 - Deprecated 운영체제 및 소프트웨어, 90년대 취약점(!), 디버깅 핀 노출
- 위협 받는 것 – 기기 탈옥, 개인정보 유출, 웜 바이러스 등

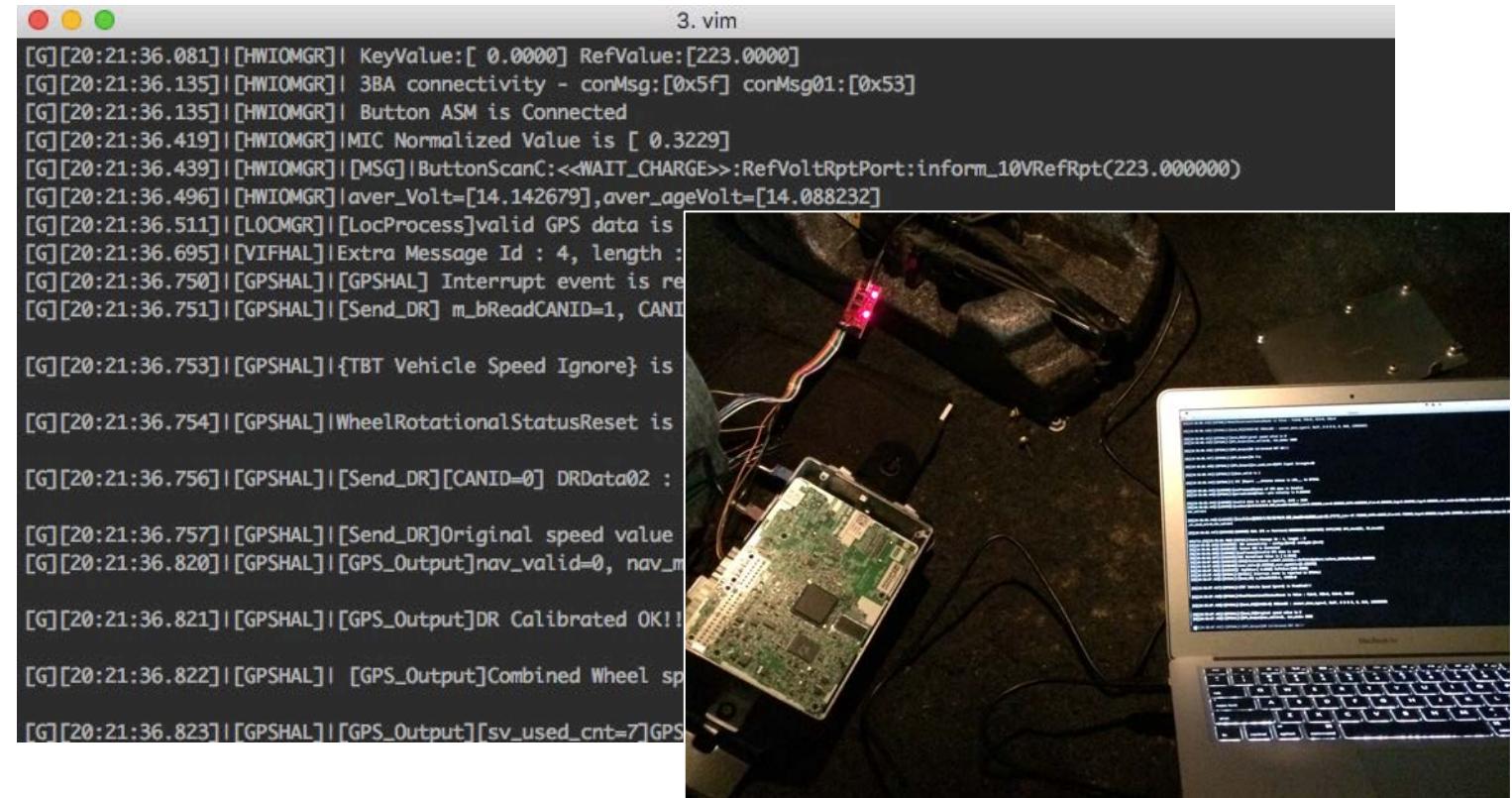
Case Study – 임베디드 시스템



- **소프트웨어**
 - Deprecated 버전 사용
 - 주기적 업데이트 및 관리 부재
 - 권한 분리 부재
- **하드웨어**
 - 테스팅 및 디버깅 포트 노출
 - 단가 절감을 위한 제한적인 하드웨어
- **원격 접근 벡터**
 - Wi-Fi, Bluetooth, HD Radio 등
 - Over-the-Air, Browser, ...

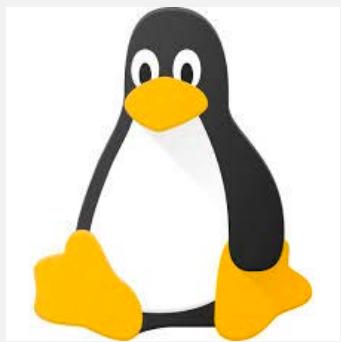
Case Study – 임베디드 시스템

예제 1: Serial 통신을 통한 쉘 액세스 및 펌웨어 덤프



Case Study – 임베디드 시스템

예제 2: 오래된 소프트웨어 => Stagefright + DirtyCOW Full-chain



Linux Kernel 3.4.35



패치권고: 최신 보안 업데이트 적용 및 지속적인 유지 보수

Case Study – 임베디드 시스템

예제 3: Remote Code Execution (RCE) via Bluetooth



```
(gdb) shell cat /proc/2210/maps | grep libc.so > exploit.py
40072000-400ba000 r-xp 00000000 b3:02 39069      /system/lib/libc.so
400ba000-400bc000 dr--p 00047000 b3:02 39069 p4 ... /system/lib/libc.so
400bc000-400bf000 rw-p 00049000 b3:02 39069      /system/lib/libc.so
(gdb) p &system
$1 = (<text variable, no debug/info>) 0x400975c0 <system>
```

```
'current_len': 9}]
{'tid': 30036, 'plen': 43, 'pdu_id': 'SDP_SVC',
 'len': 2, 'current_len': 9}]
{'tid': 54135, 'plen': 43, 'pdu_id': 'SDP_SVC',
 'current_len': 9}]
{'tid': 40462, 'plen': 43, 'pdu_id': 'SDP_SVC',
 'len': 601882348, 'total_len': 2, 'current_len': 9}]
{'tid': 21791, 'plen': 43, 'pdu_id': 'SDP_SVC',
 'len': 48, 'total_len': 2, 'current_len': 9}]
[*] libc_base: 0x400672000
[*] system: 0x400975c1
```

Case Study – 임베디드 시스템

예제 4: Home Routers RCE via Command Injection

Diagnostic Tools

Diagnostic Parameters

Diagnostic Tool: Ping Traceroute

IP address/Domain name:

Ping Count: ping(1 - 50)

Ping Packet Size: (0 - 65500 Bytes)

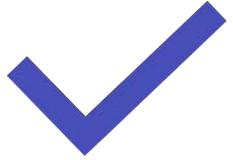
Ping Timeout: (1 - 60 Seconds)

Traceroute Max TTL: (1 - 30)

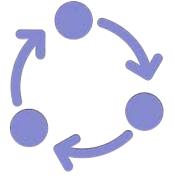
C (MIPS decompiled)

```
{  
    // ...  
    // acStack1044 contains user-controlled string  
    sVar2 = strlen(acStack1044);  
    sprintf(acStack1044 + sVar2, 0x400 - sVar2, "%");  
    util_execSystem("oal_startPing", acStack1044);  
    return 0;  
}
```

패치권고: 사용자 인풋 필터링 및 검증



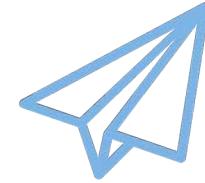
개발 중 지속적인
보안 업데이트 관리



SSDLC (Secure SDLC)
적극 검토 및 적용



양산 시, 테스팅/디버깅
인터페이스 비활성화



배포 후 OTA 업데이트
프로세스 수립

Lessons Learned – 임베디드 시스템

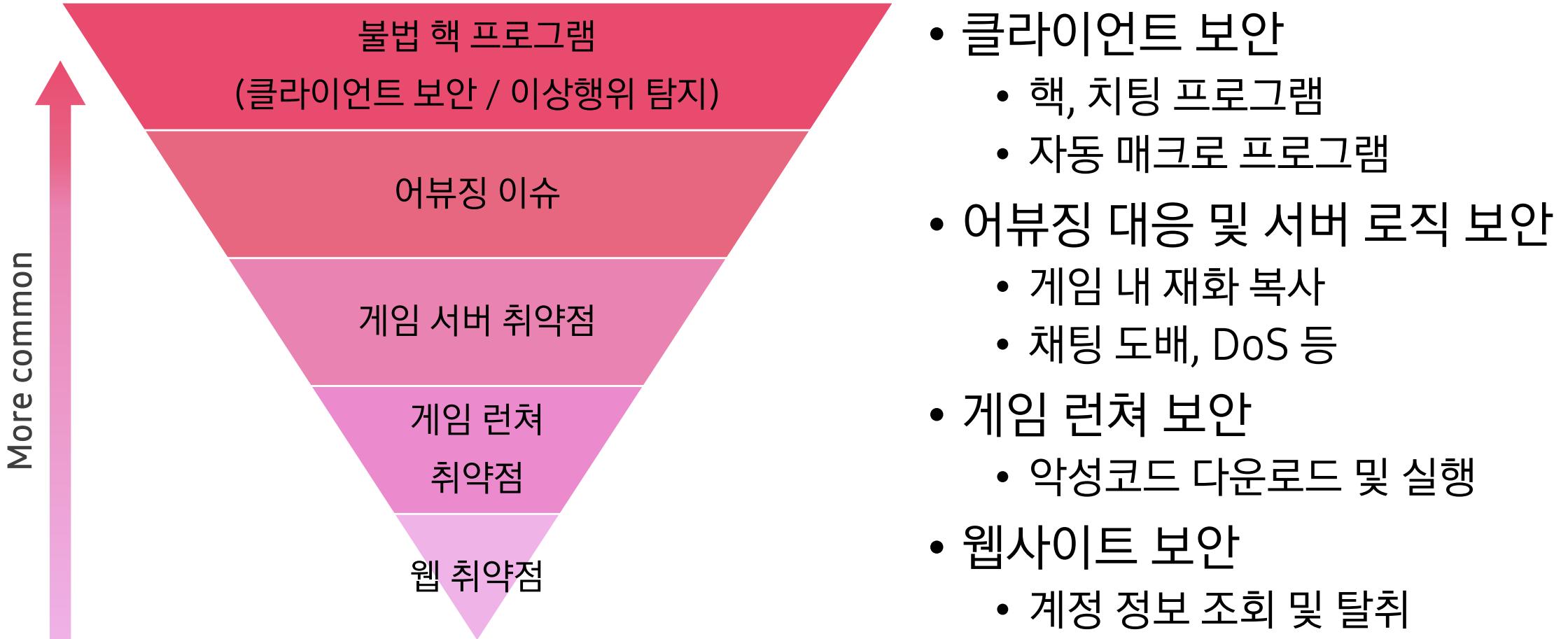


Case Study

게임 서비스

- 주요 공격 기법
 - 메모리 분석 및 변조, 바이너리 난독화/패커 분석
- 자주 발견되는 취약점
 - 불법 핵/치팅 프로그램, 클라이언트 DoS, 게임 런쳐 RCE 등
- 위협 받는 것 – 게임 이코노미, 신뢰성 + 공정성, 개인정보 등

Case Study – 게임 서비스



Case Study – 게임 서비스

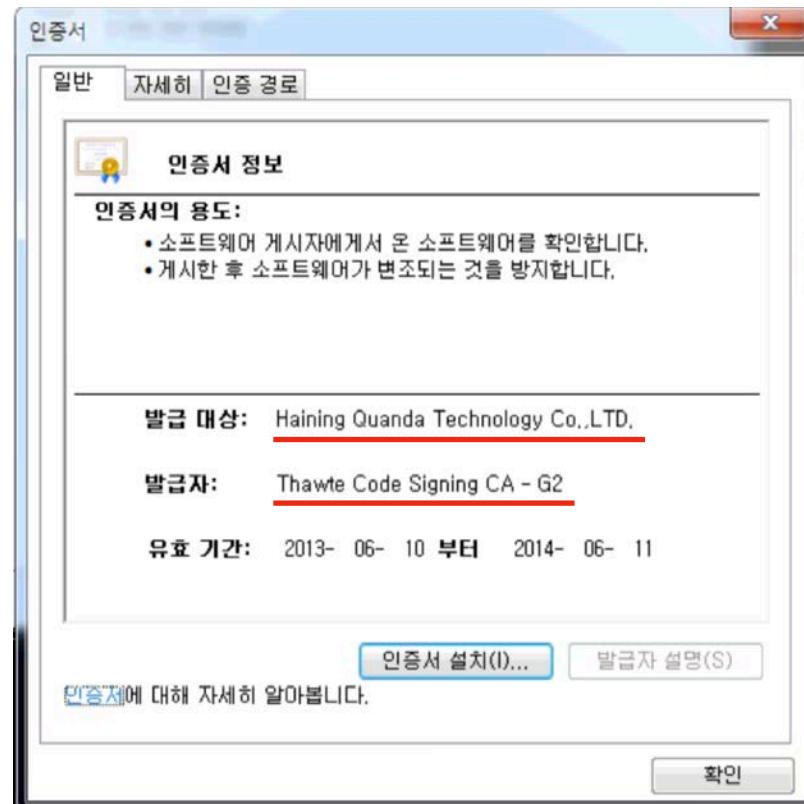
예제 1: 핵, 치팅 프로그램



ESP/월핵, 자동플레이,
Aim-bot, Trigger-bot

Case Study – 게임 서비스

예제 1: 핵, 치팅 프로그램



```
sub_2DEE09:    ; FUNCTION CHUNK AT 00000000022FB57 SIZE 0000001E BYTES
    . . .
    var_8= qword ptr -8
    ; END OF FUNCTION CHUNK FOR sub_2DEE09
```

- Themida, VMProtect 난독화
 - Code virtualization

- 유저모드 + 커널모드

- 커널 드라이버

- 유효한 인증서로 서명

```
loc_22FB57:    ; FUNCTION CHUNK AT 00000000022FB57 SIZE 0000001E BYTES
    . . .
    lea    rdi, byte_19FFD1
    mov    rdi, [rdi+110341h]
    lea    rdi, [rdi+3AF2AEBh]
    xchg   rdi, [rsp+8+var_8]
    jmp    nullsub_160
    ; END OF FUNCTION CHUNK FOR sub_2DEE09
```

- 악성코드 배포

Case Study – 게임 서비스

예제 2: P2P 기반 게임 치팅 및 DoS



- 게임이 시작되면 P2P로 전환 => 각 클라이언트가 보낸 데이터 **신뢰**
- 악성 패킷 전달 => 데이터 파싱 **오류** => 게임 강제 종료 (crash) 유발

Case Study – 게임 서비스

예제 3: 게임 런쳐 취약점을 통한 RCE

```
$ curl -si http://localhost:1120/agent
{
  "pid" : 3140.000000,
  "user_id" : "S-1-5-21-1613814707-140385463-2225822625-1000",
  "user_name" : "S-1-5-21-1613814707-140385463-2225822625-1000",
  "state" : 1004.000000,
  "version" : "2.13.4.5955",
  "region" : "us",
  "type" : "retail",
  "opt_in_feedback" : true,
  "session" : "15409717072196133548",
  "authorization" : "11A87920224BD1FB22AF5F868CA0E789"
}
```

- 로컬호스트에 서버 실행
 - JSON RPC 프로토콜
- 모든 리퀘스트는 유효한 Authorization 필요
 - 로컬에서만 열람할 수 있다는 전제
 - DNS Rebinding 통해 우회 가능
- RPC 통해서 프로그램 설치 및 실행 가능



클라이언트 보안
(안티 치트)



서버사이드 모니터링
통해 치팅 및 어뷰징 감지

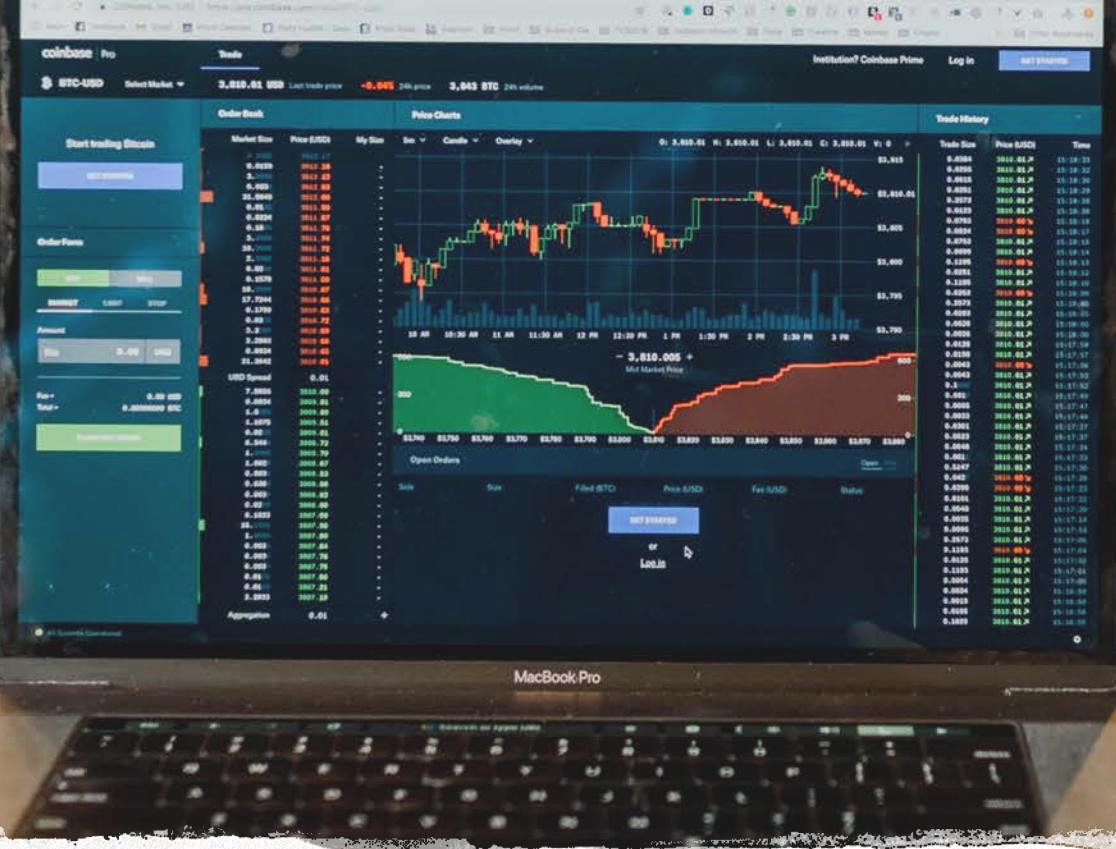


높은 권한을 가진
게임 런쳐 보안



웹사이트 및 in-game
웹앱 보안

Lessons Learned – 게임 서비스

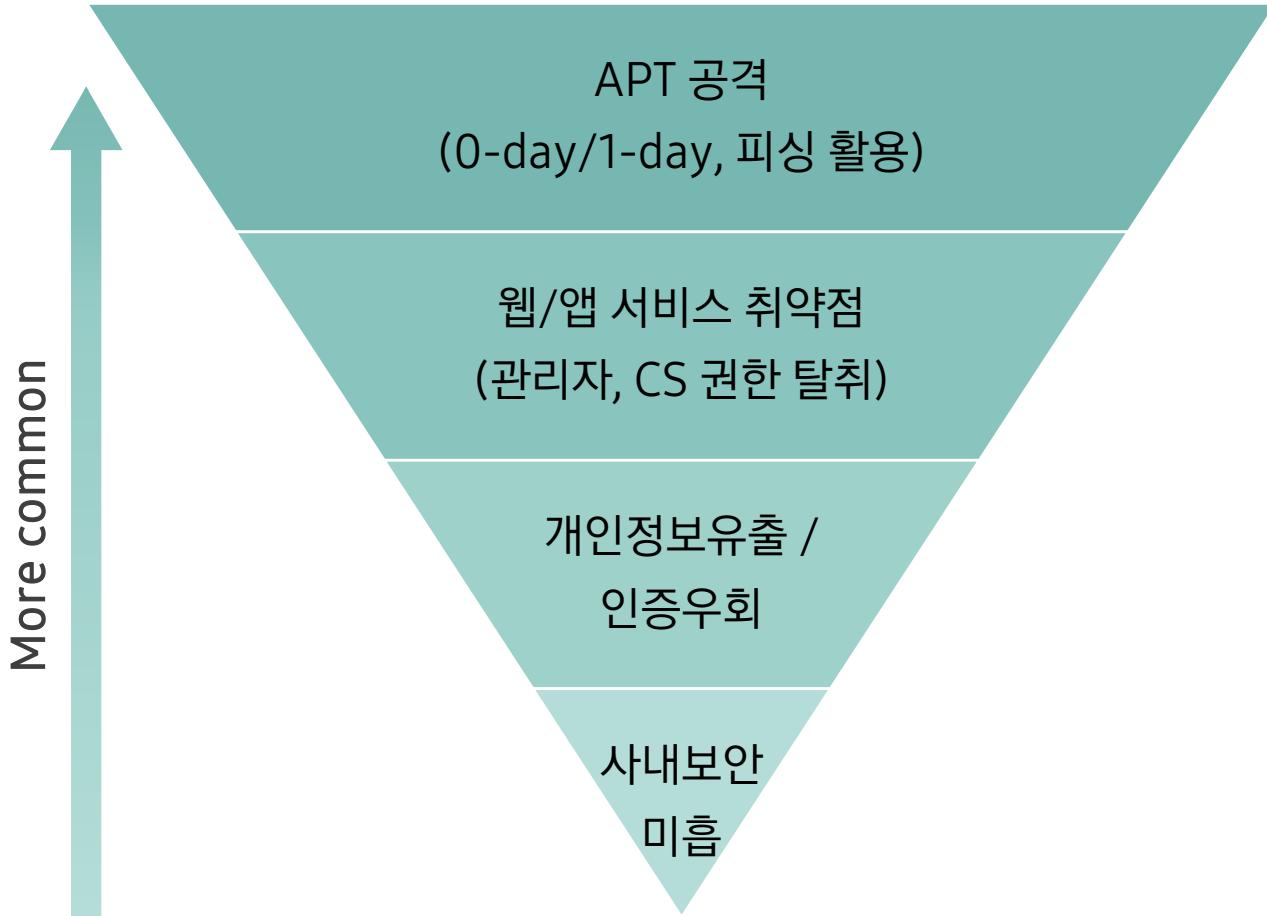


Case Study

금융권 (은행 / 가상화폐거래소)

- 주요 공격 기법
 - 계정 탈취, 0-day 사용 APT 공격, 보안 솔루션 우회, 피싱, 자동거래 등
- 자주 발견되는 취약점
 - 웹/모바일 앱 취약점, 관리자/개발자 노출, 내부 시스템 권한 분리 미흡 등
- 위협 받는 것 – 자산, 개인정보, 서비스 안정성, 사내 보안 등

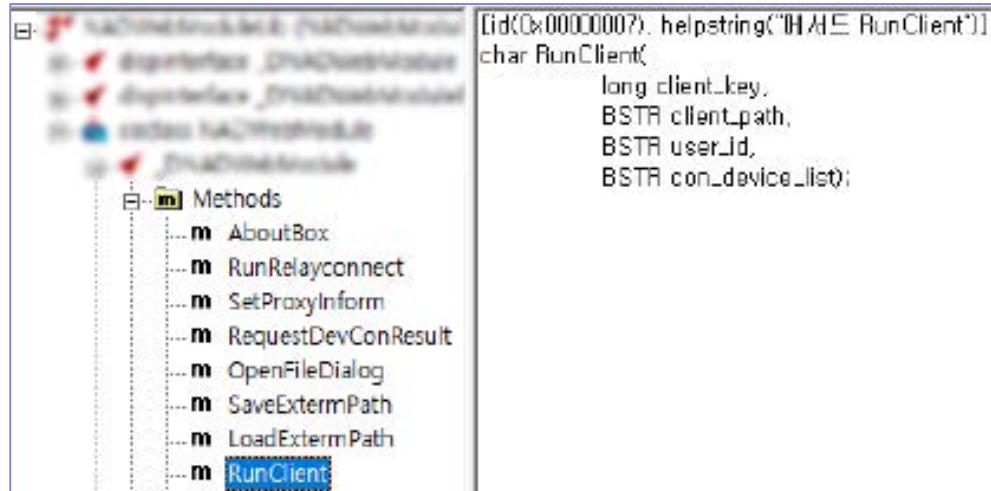
Case Study – 금융권



- APT 스타일 공격
 - 개발자 대상 공격
 - 입사 지원서 악성코드 => 사내 침투
- 부정거래 및 이상징후
 - 로깅, 모니터링
 - 사고대응
- 사내보안
 - 물리적 보안 (입출입, Wi-Fi, 망분리)
 - 비밀번호, 키 관리
- 퍼포먼스

Case Study – 금융권

예제 1: 접근통제 솔루션 취약점을 통한 내부망 침투



```
<html>
<script>
window.onload = function() {
    target.RunClient(1, "c:\\\\...\\\\powershell.exe ...",
        "", "");
};

</script>
<object id="target" classid="clsid:XXXX...XXX">
</html>
```

권고사항: 호스트 검증 및 사용자 입력 인자값 필터링

Case Study – 금융권

예제 2: 내부 git repository 액세스 (접근제어/권한분리 미흡)



GitLab



Bitbucket

권고사항: ssh key만 허용하는 인증 또는 2FA 활용, 권한분리

Case Study – 금융권

예제 3: 앱 루팅 탐지 우회

0

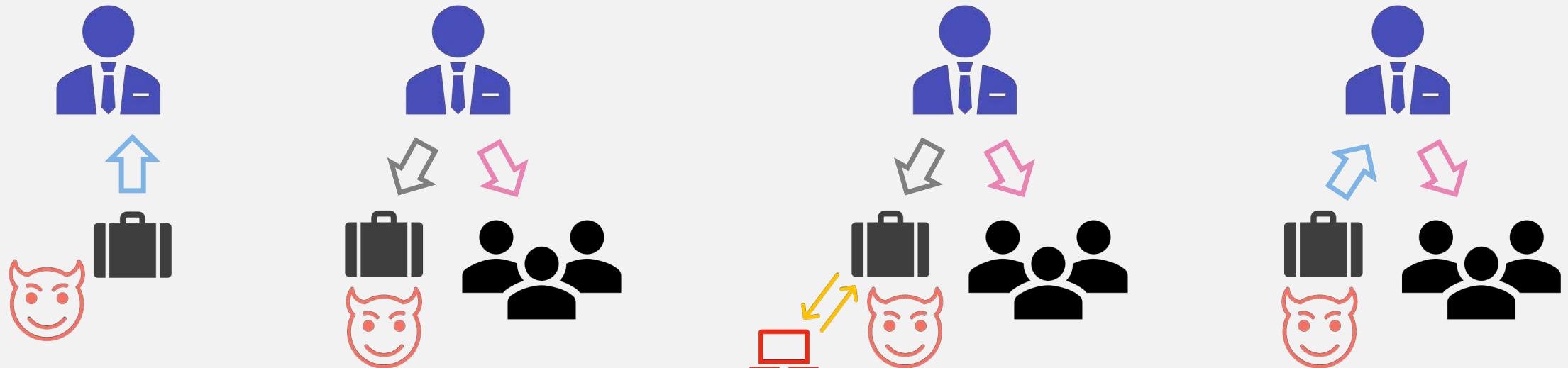
현재까지 업무를 진행하면서 앱 위변조 방지, 루팅 탐지, 난독화 등 다양한 모바일 보안 솔루션이 성공적으로 우회되지 않은 횟수

고려사항: 보안 솔루션을 적용해도 취약점이 사라지는 것은 아님

Case Study – 금융권

예제 4: 사내 작업용 노트북 탈취 시나리오 (물리 보안)

- 하청업체 상주직원에게 내부 지급하는 작업용 노트북은 외부 반출이 안됨



권고사항: 2인 이상의 보안 직원 상주, 보관물품 CCTV 모니터링 강화

Case Study – 금융권

예제 5: 핸드폰 번호 조회를 통한 사용자 실명 유출

```
# ...
params do
  # ...
  requires :real_name
  requires :phone_number
end
post 'send_request' do
  phone_number = params[:phone_number]
  request = SendRequest.create_send_request!((
    phone_number: phone_number,
    real_name: params[:real_name],
  ))
  render(request, with: SendRequest)
# ...
```

Ruby on Rails (grape)

패치권고: 일부 정보 마스킹, 불필요한 API 제거 (사용성 vs 보안성)

Case Study – 금융권

예제 6: 0-day/1-day 취약점 및 백오피스 툴 공격



Firefox zero-day was used in attack against Coinbase employees, not ...
ZDNet - Jun 20, 2019
"On Monday, Coinbase detected & blocked an attempt by an attacker to leverage the reported **0-day**, along with a separate **0-day Firefox** ..."

Mozilla Patches Firefox Critical Flaw Under Active Attack
Threatpost - Jun 19, 2019

가상화폐거래소 APT 공격

백오피스 툴 공격

사용자 검색	
ID	닉네임
322	=SUM(A1:A2)

A1	B	C	D	E	F
1	id	status	nickname		322
2	322	active			

사용자 검색	
ID	닉네임
322	=HYPERLINK("c:\windows\system32\calc.exe")

CVE-2017-8759:
GC 이용한 악성 WSDL 다운로드 및 파싱 (RCE) 가능



고객 자산 및
계정 보안의 중요성



최신 공격기술과
다양한 피싱 공격 대비



침해사고 시, 역추적할
충분한 로그 필요



놓치기 쉬운 사내보안

Lessons Learned – 금융권

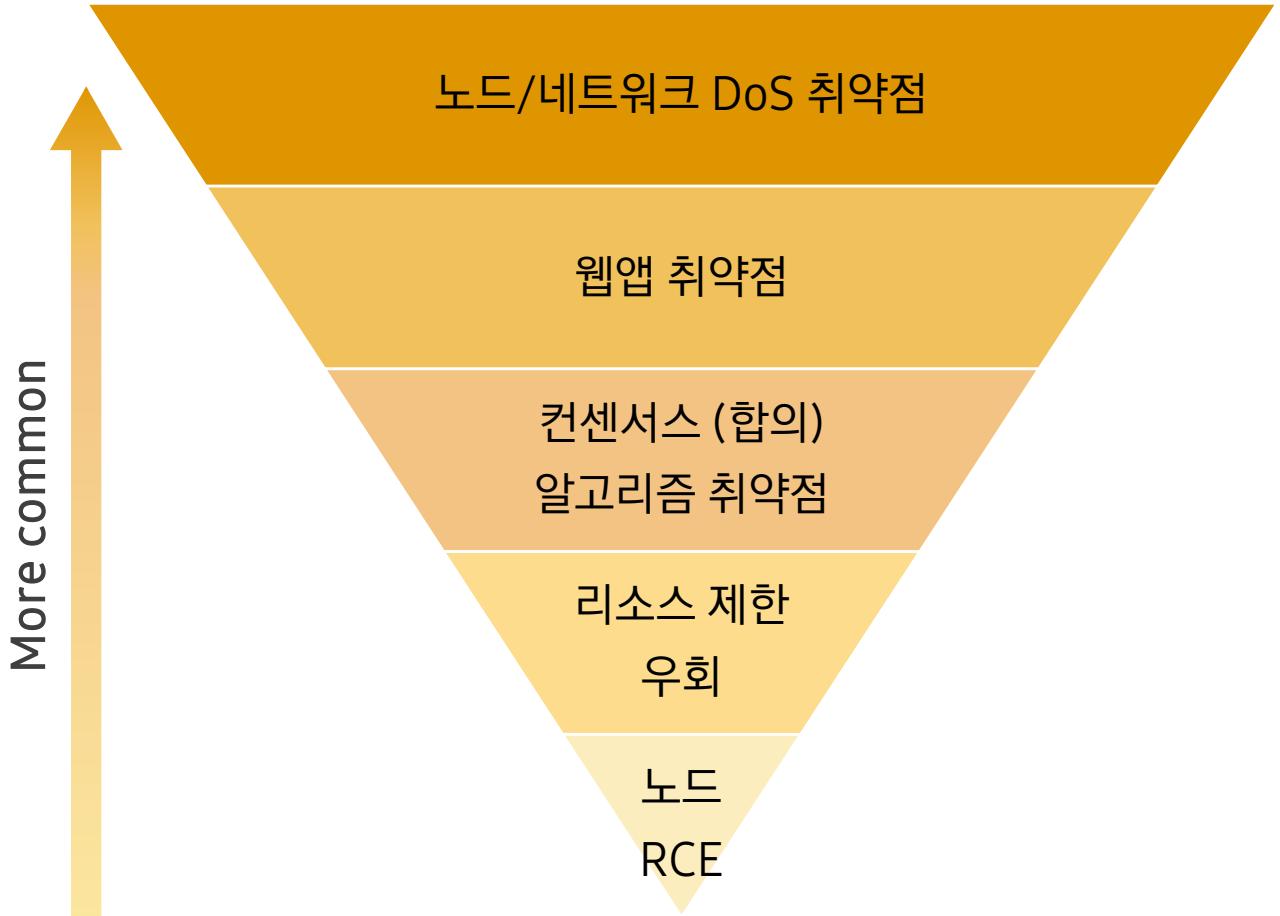


Case Study

블록체인 플랫폼

- 주요 공격 기법
 - 악성 스마트 컨트랙트 배포, 합의 알고리즘 악용 등
- 자주 발견되는 취약점
 - 노드/네트워크 DoS, 새로운 타입/필드 검증 미흡, gas 제한 우회 등
- 위협 받는 것 – 체인 안정성, 코인 이코노미 등

Case Study – 블록체인 플랫폼



- 블록체인의 핵심: 탈중앙화 (분산화)
 - 하지만, 모든 참여 노드가 죽어버린다면?
 - DoS도 엄청난 파급력의 취약점 클래스
- 사용성 편의를 위해 제공되는 웹 UI
 - 중앙 서버 로직
 - 계정 및 자산 탈취 등
- 합의 과정의 오염 또는 이코노미를 좌우하는 리소스 제한 등의 우회
 - 블록체인 공정성과 안정성 위협
- 대다수가 같은 프로그램 실행
 - RCE 취약점이 발견된다면?

Case Study – 블록체인 플랫폼

예제 1: API/RPC 인자값 검증 미흡으로 인한 DoS 취약점

```
func parseArgs(rawArgs json.RawMessage, types []reflect.Type) ([]reflect.Value, Error) {  
    // ...  
    for i := len(args); i < len(types); i++ {  
        if types[i].Kind() != reflect.Ptr {  
            return nil, &invalidArgsError{fmt.Sprintf("missing value!")}  
        }  
        args = append(args, reflect.Zero(types[i]))  
    }  
    return args, nil  
}  
  
func (api *API) GetBlockByRange(start *rpc.BNumber, end *rpc.BNumber) (map[string]interface{}, error) {  
    // ...  
    s := start.Int64()  
    e := end.Int64()  
    // ...  
}
```

Case Study – 블록체인 플랫폼

예제 1: API/RPC 인자값 검증 미흡으로 인한 DoS 취약점

```
INFO[01/02,11:24:14 +00] [29] FastWebSocket endpoint opened  
url=ws://0.0.0.0:8552
```

```
panic: runtime error: invalid memory address or nil pointer dereference  
[signal SIGSEGV: segmentation violation code=0x1 addr=0x0 pc=0x490dc00]
```

패치: nil 값을 명시적으로 확인

```
func (api *API) GetBlockByRange(start *rpc.BNumber, end *rpc.BNumber) (map[string]interface{}, error) {  
    // ...  
+    if start == nil || end == nil {  
+        return nil, errRangeNil  
+    }  
    s := start.Int64()  
    e := end.Int64()  
    // ...  
}
```

Case Study – 블록체인 플랫폼

예제 2: 웹 API 취약점을 통한 contract 주소 임의 변경

```
module.exports = {
  path: '/api/deploy/:contractId',
  post: async (req) => {
    const body = req.body;
    const cId = req.params('cId', true);
    const cAddr = body.data.cAddr;
    // ...
    if (body.result && cAddr) {
      await chain.contract
        .updateContractInfo(cId, cAddr);
    }
    // ...
  },
};
```

Nodejs

패치권고: 정보 변경 권한 여부 검증; 내부에서만 호출 가능하도록 변경

Case Study – 블록체인 플랫폼

예제 3: 리소스 제약 우회 및 리소스 소진 공격

```
func (s *API) EstimateGas(ctx context.Context, args CallArgs) (hexutil.Uint64, error) {  
    // ...  
    if uint64(args.Gas) >= params.TxGas {  
        hi = uint64(args.Gas)  
    } else {  
        hi = params.UpperGasLimit  
    }  
    cap = hi  
  
    // ...  
    _, _, _, failed, err := s.doCall(ctx, args, rpc.PendingBlockNumber, vm.Config{}}, localTxExecutionTime)  
    // ...  
}
```

estimateGas API를 사용하면 자원 제한없이 EVM 실행 가능

Case Study – 블록체인 플랫폼

예제 3: 리소스 제약 우회 및 리소스 소진 공격

```
contract PoC {  
    mapping(uint256 => string) data;  
    function sendMe() public view returns(uint256) {  
        uint256 a;  
        for(uint256 i=0; i<0xffffffffffffffff; i++) {  
            a++;  
            data[i] = "AAAAAAAAAAAAAAA...AAA";  
        }  
        return a;  
    }  
}
```

Solidity

statedb에 값을 계속 작성하여 DoS를 유발하는 컨트랙트 배포

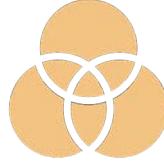
패치권고: VM Config에 UseOpcodeCntLimit & 실행 TimeLimit 설정



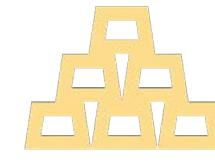
탈중앙화 + 중앙시스템
합쳐진 모델 다수 존재



서비스 거부 (DoS)
공격이 미치는 파급력



오픈소스 기반 프로젝트
코드 관리의 어려움



메인넷 운영 시작 후
변경이 쉽지 않음

Lessons Learned – 블록체인 플랫폼

Case Study

연구 및 개발

- 주요 접근 방법
 - 최신 논문 리뷰, 창의적인 생각(!), 선행 연구
- 주요 관심 주제
 - 새로운 보호 기법 (및 공격 기법), 취약점 자동 분석 등
- 궁극적 목표 – 기존 기술 개선, 새로운 위협 발견 및 제거



Case Study – 연구 및 개발



기업과제

- 새로운 보호기법 설계 및 검증
- 공격기법 및 보호기법 우회 방법 연구
- 기간: 5~8주

정부과제

- 취약점 분석 기법 자동화 연구
- 새로운 대상 및 패러다임 가능성 연구
- 기간: 12~48+주

Case Study – 연구 및 개발

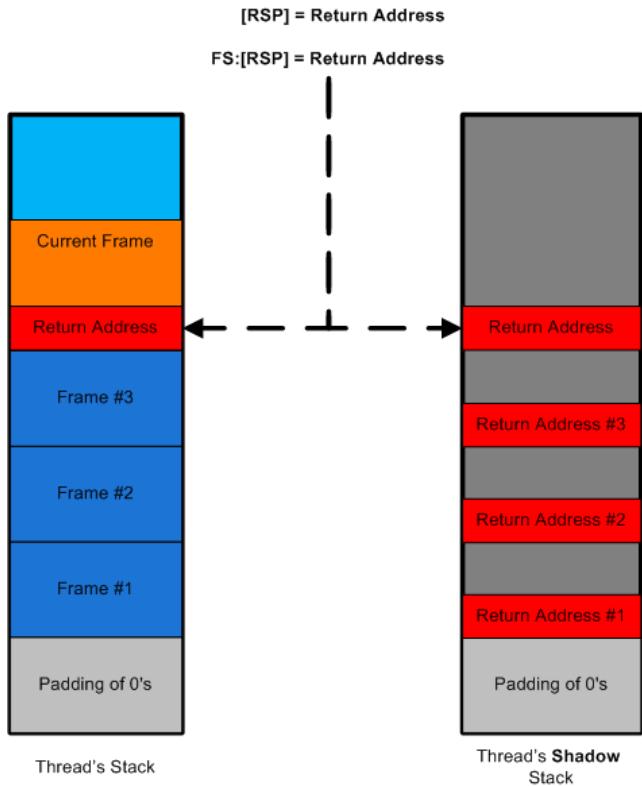
예제 1: DARPA Cyber Grand Challenge



- 2013년부터 인프라스트럭쳐, 문제 설계 및 제작 등 시작
 - 2014년: 대회 공개
 - 2015년: 예선전 진행
 - 2016년: 본선 진행 (DEFCON)
- 우승 상금: 200만 달러
 - 2등 100만 달러, 3등 75만 달러
- 취약점 분석 및 패치 자동화

Case Study – 연구 및 개발

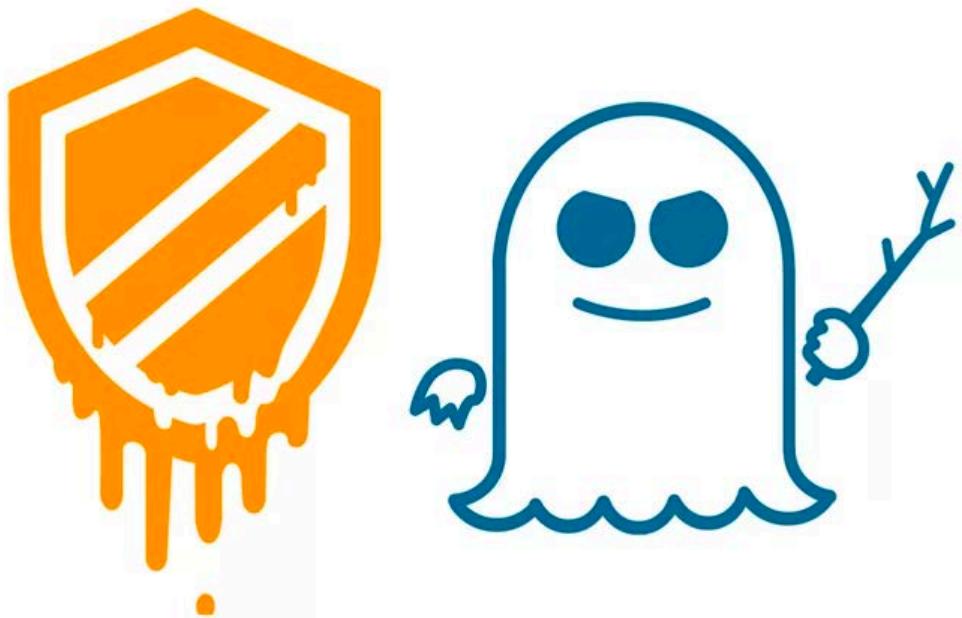
예제 2: 새로운 보호기법 (RFG) 보안성 검증



- Control Flow Guard (CFG)를 보완하는 기술
 - 스택에 있는 리턴 주소가 오염되지 않았는지 안전하게 체크할 수 있는 방법
- Control Flow Integrity (CFI)
- 호환성, 퍼포먼스 비용 문제 등
- **배포 전, 보안성 검증 필요**

Case Study – 연구 및 개발

예제 3: Spectre in Browser 가능성 연구



- 브라우저에서 Spectre CPU 취약점을 공격할 수 있을까?
- 그렇다면, 어떤 방어 기법을 설계해야 할까?
 - 퍼포먼스 고려
- 새로운 Spectre Variants 증명
- 기존 위협 및 새로운 위협에 대한 연구 및 미래방향 고민

Case Study – 연구 및 개발

예제 4: Thunderbolt DMA 보호기능 검증

THUNDER[⚡]CLAP



- 썬더볼트 포트를 이용한 PCIe 통한 DMA 공격
 - 운영체제에서 IOMMU 이용한 방어
- DMA 보호 기능이 충분히 안전한가?
 - 각종 드라이버 취약점 분석 (우회)
- 새 스펙의 출현 => 새 공격 벡터
=> 새 보호기능 연구개발 및 검증



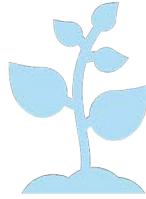
이론과 현실의 경계



최신 트렌드 팔로우업
+ 트렌드 리더



연구의 가치를 아는
대기업 및 정부기관



지속적인 사내 연구개발
지원과 독려

Lessons Learned – 연구 및 개발

Conclusion



Conclusion

Offensive Security Research

Conclusion

Active Threat Modelling

Continuous Risk Assessment
+ Risk Management

Conclusion

Invest in Security, and Thank Me Later

Conclusion

Do NOT try to do all at once

Share information, Ask for help

Thank you.

TRUST
With SAMSUNG

Questions?



Samsung
Security Tech
Forum

SAMSUNG
Research