모든 것이 컴퓨터다

그리고 그 컴퓨터들은
소프트웨어부터 칩까지
뚫립니다.

김근호 · ROOT-Kim
CYAI Lab · SSG 회장

SSG Internal Seminar

질문 링크

질문은 여기로 남겨주세요.

발표 중 궁금한 점을 QR로 남기면 마지막 Q&A 시간에 같이 보겠습니다.

Embedded security intro · Q&A link

QUESTION

이 건물에 컴퓨터가 몇 대일까요?

스마트폰과 노트북을 빼도

~~수십 대~~

수백 대

공유기 · IP카메라 · 월패드 · 엘리베이터 제어기

자판기 · 프린터 · 에어컨 · 출입 시스템

대부분은 보안 업데이트를 거의 받지 않습니다.

DEFINITION

임베디드가 뭔데?

DEVICE

특정 기능만 하는 컴퓨터

공유기, IP카메라, 드론, 자동차 ECU. 인터넷이 붙으면 IoT가 되고, 원격 공격면이 생깁니다.

ATTACK

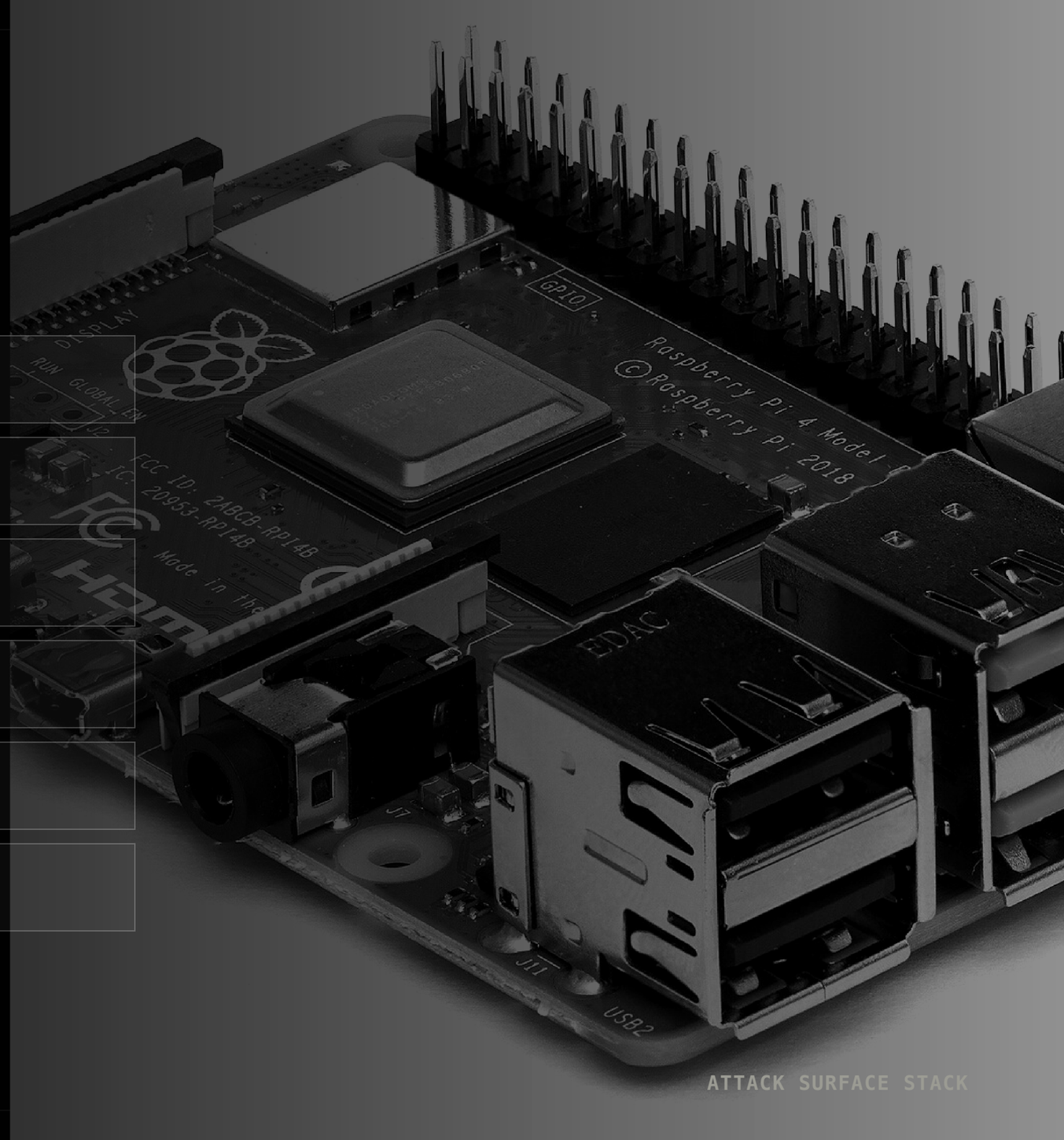
소프트웨어부터 칩까지

펌웨어 분석, 웹 인터페이스, 네트워크, JTAG/UART, 전력/EM/글리칭까지 이어집니다.

ONE ROUTER

공유기 하나 안에 모든 공격면이 있다

- 관리자 웹페이지 ← 웹해킹
- 스마트폰 연동 앱 ← 앱 분석
- 내부 프로그램 ← 시스템해킹
- 통신/패킷 ← 네트워크
- 리눅스 OS ← 커널
- 기판/칩/포트 ← 하드웨어



뭘 공부해야 하나요?

1

Linux

터미널에서 파일 찾고 프로세스 보기

2

Network

IP, 포트, HTTP, 패킷 흐름

3

C / RE

Ghidra에서 함수 구조 따라가기

웹 · 시스템 · 네트워크 · 하드웨어가 **한 기기 안에서 합쳐집니다.**

ROADMAP

어떻게 공부하나요?

01
Ubuntu VM
터미널

02
binwalk
펌웨어 풀기

03
Burp
웹 요청 보기

04
Ghidra
바이너리 읽기

05
Hardware
공유기 분해

1~4까지는 노트북 하나면 됩니다.

저도 이 순서로 스마트 도어벨을 분석했고, 3개월 뒤 취약점 18개와 CVE를 받았습니다.

TOOLS

도구와 장비

소프트웨어: 0원

binwalk · Ghidra · FirmAE · Wireshark · Burp Suite

기본 하드웨어: ~7만원

중고 공유기 · 어댑터 · 인두 · 멀티미터

무선/RF: 25~40만원

Flipper Zero · SDR

칩 레벨: 30만원+

JTAG debugger · ChipWhisperer

처음은 노트북만으로 됩니다. 하지만 실제 기기를 만지는 순간부터는 어쩔 수 없이 돈이 조금씩 듭니다.

REALITY CHECK

임베디드는 느리지만 깊다

첫 접근이 어렵다

기기마다 구조가 다르고, 어디서부터 봐야 할지 감이 잘 안 잡힙니다.

재현 어려움

기기와 펌웨어 버전이 모두 변수

그래서 강함

한 번 뚫으면 실제 세계와 연결됨

깊게 파는 사람에게 기회가 남아있는 분야.

PREP

지금 단계에서 하면 좋은 준비

1

Linux 터미널에서 파일 찾기, 프로세스 보기

2

네트워크 기초: IP, port, HTTP 요청/응답

3

C 기초: 포인터, 문자열, 함수 호출 흐름

아직 해킹을 몰라도 괜찮습니다. CS 기초가 임베디드 분석의 재료가 됩니다.

하드웨어 공격은 한 가지가 아닙니다

Debug Interface

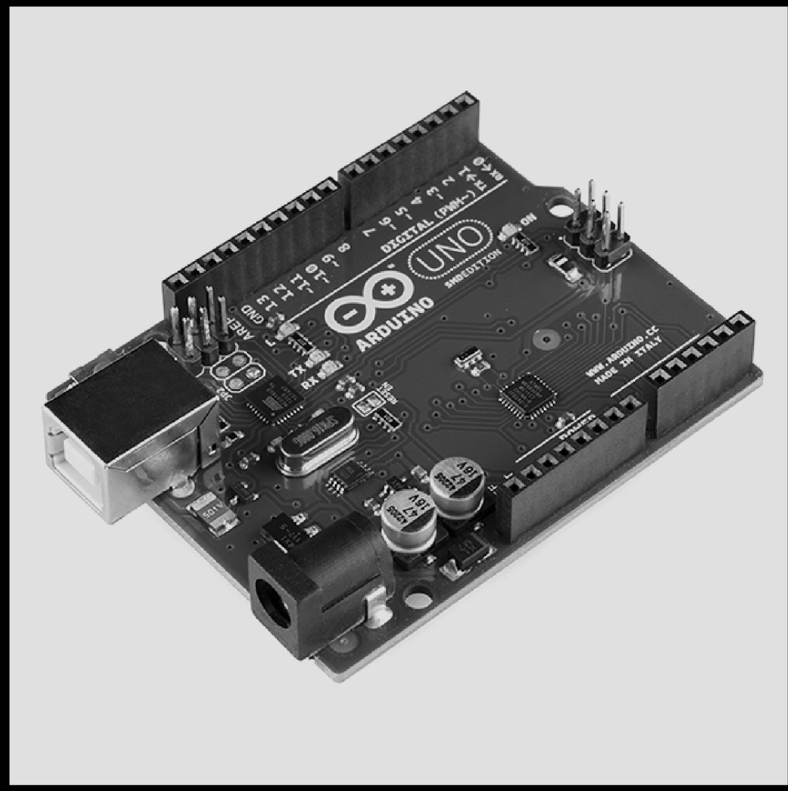
UART, JTAG, SPI flash, 인두, 로직 애널라이저.

Side-channel

Power, EM, timing leakage로 secret/key를 추정합니다.

Fault Injection

Voltage/clock glitching으로 secure boot나 check를 우회합니다.



펌웨어를 풀면 실제 리눅스가 나옵니다

```
$ binwalk -e firmware.bin
$ cd squashfs-root/

/etc/passwd
/www/cgi-bin/
/bin/busybox
/usr/sbin/dropbear
/lib/*.so
```

파일시스템

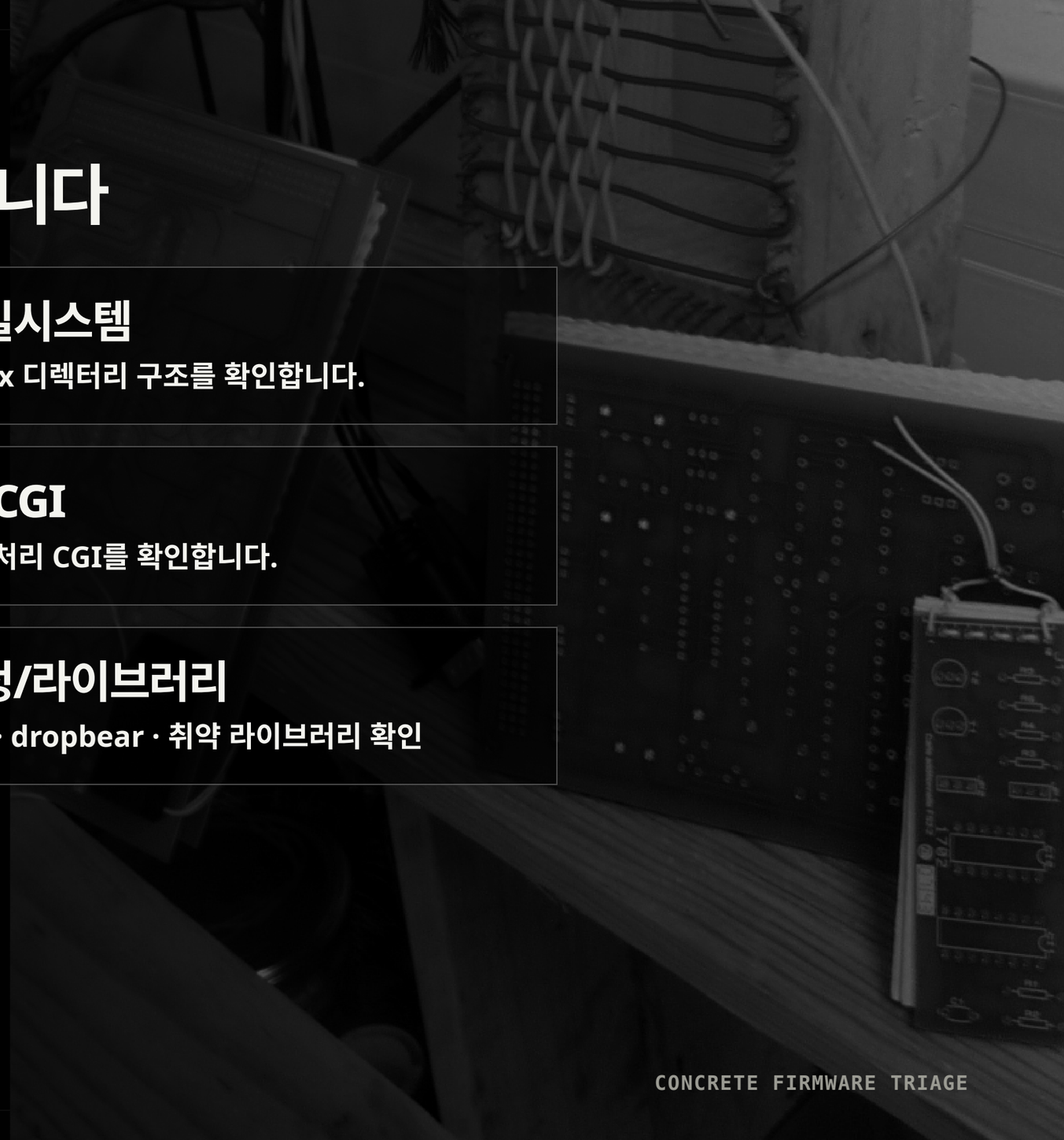
Linux 디렉터리 구조를 확인합니다.

웹/CGI

API 처리 CGI를 확인합니다.

계정/라이브러리

계정 · dropbear · 취약 라이브러리 확인



실제 예시: IoT 스마트 도어벨 분석

1. 기기/보드 확인

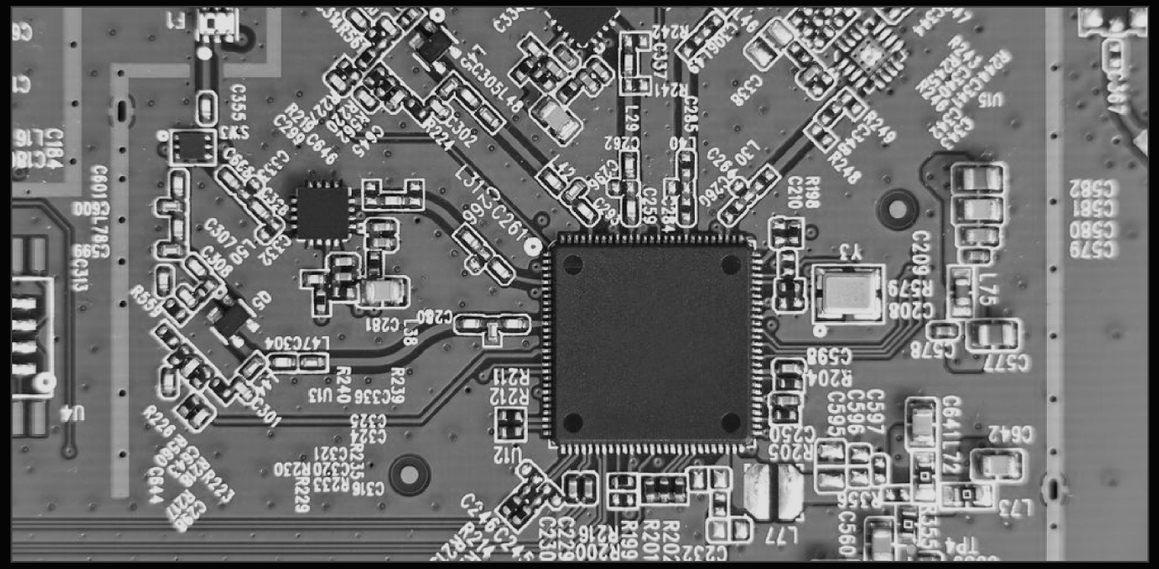
스마트 도어벨 PCB, UART 핀, 칩 정보를 먼저 확인합니다.

2. 펌웨어 열기

binwalk로 파일시스템을 풀고 설정·웹·API 흔적을 확인합니다.

3. 취약점으로 연결

앱·트래픽·펌웨어 결과를 합쳐 재현합니다.



분석 가이드 QR

전체 과정과 캡처 자료 보기

관리자 페이지도 웹해킹입니다

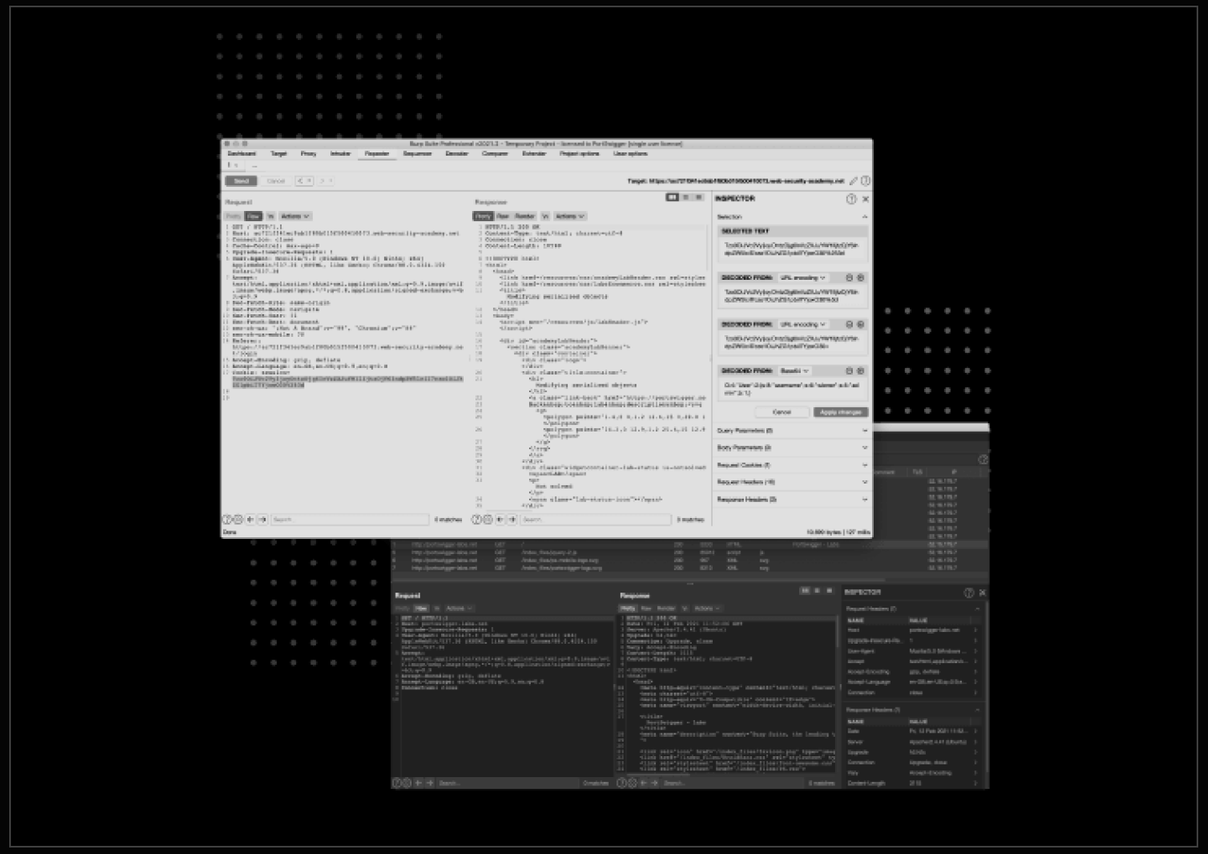
Burp로 보는 것

로그인 요청, 세션 쿠키, 설정 변경 API, 파일 업로드 엔드포인트를 봅니다.

자주 나오는 취약점

인증 우회, command injection, path traversal, CSRF, default password.

웹해킹을 공부한 사람은 이 지점에서
임베디드 분석으로 자연스럽게 이어집니다.



Ghidra에서는 뭘 보나

문자열

/bin/sh, admin, password, curl, wget, system

위험 함수

strcpy, sprintf, system, popen, memcpy

입력 흐름

HTTP parameter → parser → command execution

처음부터 어셈블리를 다 읽는 게 아니라, 문자열과 함수 이름으로 위험한 흐름을 먼저 좁힙니다.

기판에 선을 꽂으면 내부가 보입니다

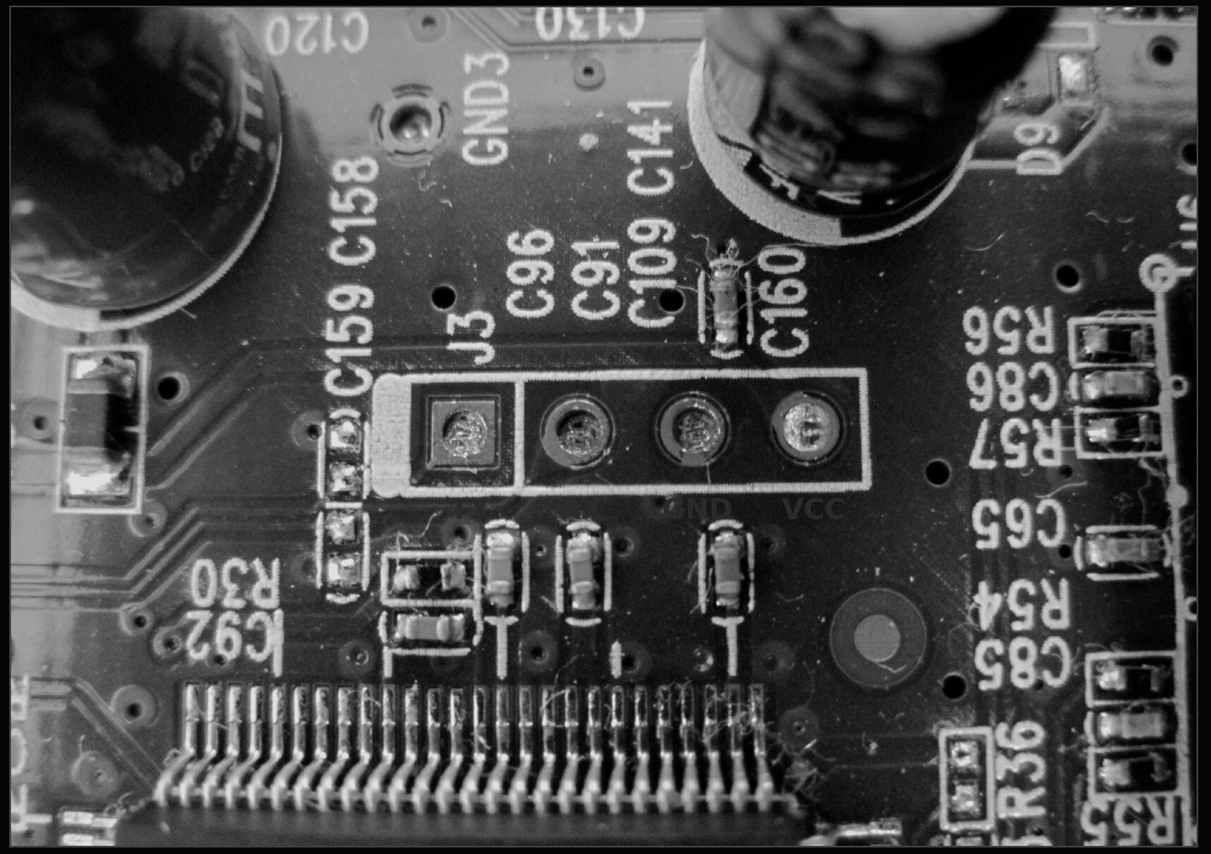
UART

TX/RX/GND를 찾아 USB-TTL로 연결하면 부팅 로그와 shell을 볼 수 있습니다.

JTAG

칩 디버깅 인터페이스입니다. 메모리 읽기, breakpoint, firmware dump에 사용됩니다.

전압, 접지, 핀맵을 확인하지 않으면 기기를 손상시킬 수 있습니다.



이걸 어디서 써먹나요?

CTF

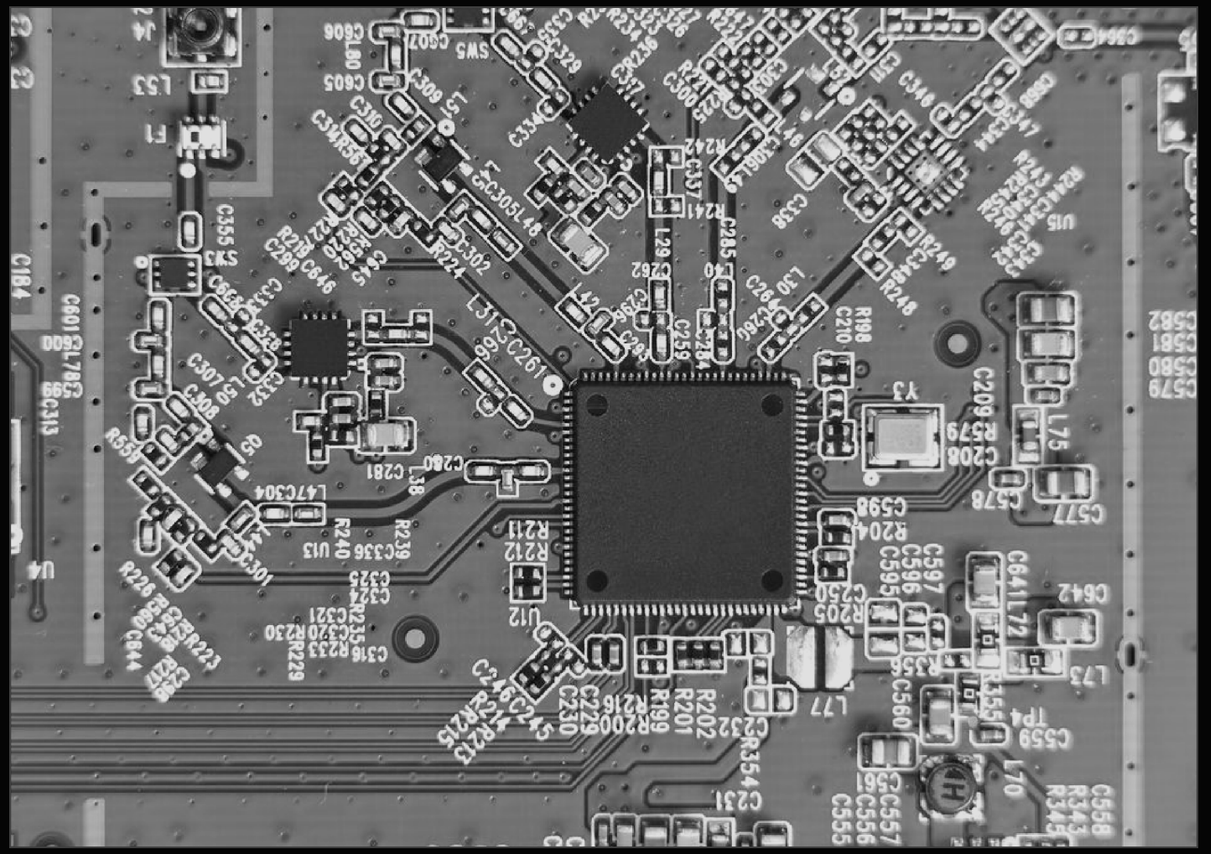
펌웨어, 리버싱, pwn, crypto가 한 문제 안에서 섞입니다.

Pwn2Own

공유기, NAS, 프린터, 스마트홈, 차량 대상 실전 공격이 나옵니다.

Bug Bounty

IoT vendor, router vendor, automotive supplier disclosure로 이어질 수 있습니다.



직무로 가면 이렇게 나뉩니다

제품 보안

출시 전 펌웨어/앱/API 점검, threat modeling, secure boot, OTA 보안.

취약점 연구

공개 제품 분석, 0-day 연구, vendor disclosure, exploitability 평가.

차량/OT

ECU, IVI, CAN, 진단 프로토콜, 공장/에너지 설비 보안.

도구 개발

펌웨어 에뮬레이션, 정적 분석, fuzzing, 자동화 파이프라인.

전자제품 하나를 뜯으면
웹, 시스템, 네트워크, 하드웨어가
한 번에 연결됩니다.

임베디드의 세계에 들어오면
그냥 물건이 아니라 분석 대상처럼 보일 겁니다.