RSAConference2020
San Francisco | February 24 – 28 | Moscone Center

HUMAN ELEMENT

SESSION ID: HTA-R07

Automotive/IoT Network Exploits: From Static Analysis to Reliable Exploits

Jonathan Brossard

Managing Director
Moabi
@endrazine

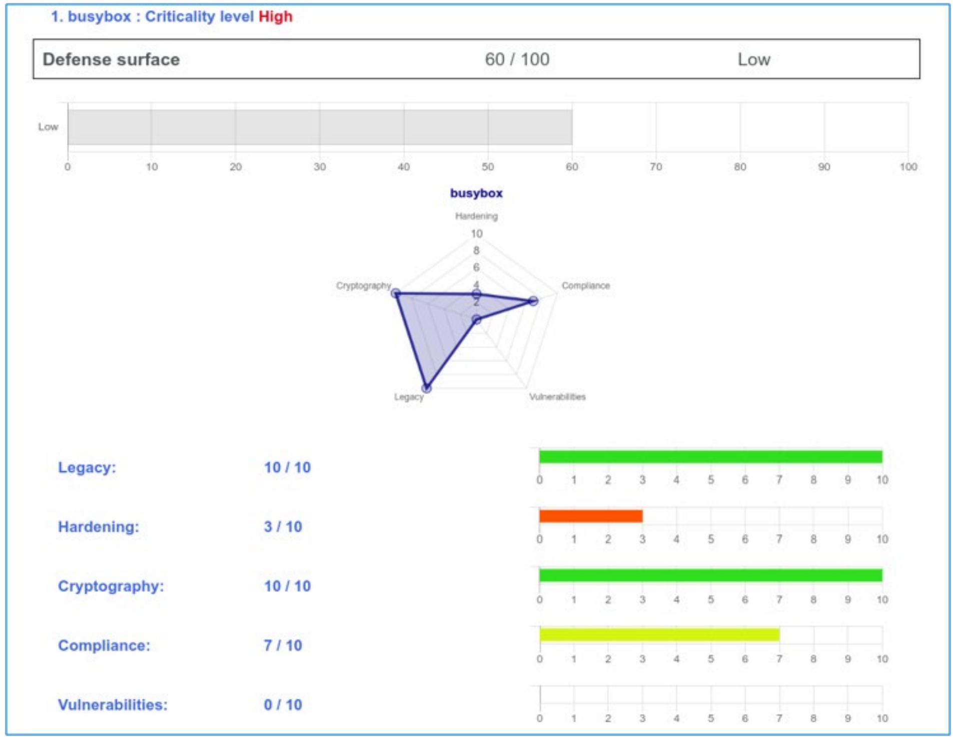
Who am I ?

- CEO at Moabi (San Francisco, Paris)
- Security Researcher : Bitlocker, MS IE/Edge, most BIOS Firmwares, SAP.
- "Inventor" of Hardware Backdooring (Rakshasa, 2011)
- Firmware Security Pioneer (INTEL-SA-00016)
- Previously Director of Offensive Security at Salesforce
- Speaker at Blackhat (x5), Defcon (x3)
- MS Engineering, MS C.S., PhD candidate

MOABI

Disrupting static analysis for IIoT

- 128b taint analysis and symbolic execution
- Built for IIoT / Industrial Processes
- Scales to 100k+ binaries / day / client
- Covers entire SSDLC
- Finds 0days automatically
- Enables secure supplier relationship management



1. busybox : Criticality level High

Defense surface	60 / 100	Low

Legacy:	10 / 10	
Hardening:	3 / 10	
Cryptography:	10 / 10	
Compliance:	7 / 10	
Vulnerabilities:	0 / 10	

Return on Experience

"Moabi is the perfect fit to address the new challenges posed by the greatest changes in the automotive industry, in over a century : electric cars, connected and self-driving vehicles ".

engineering department – Renault Nissan Mitsubishi Alliance

RSAConference2020

IIOT static analysis : case study

Auditing GenIVI

Use case : audit of GenIVI

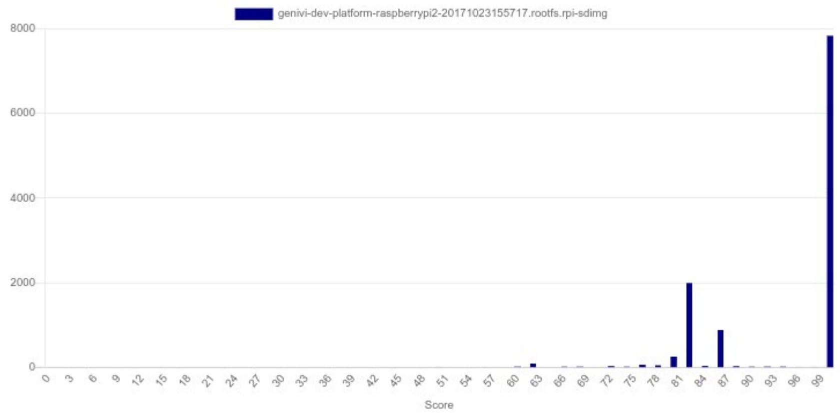
GENIVI

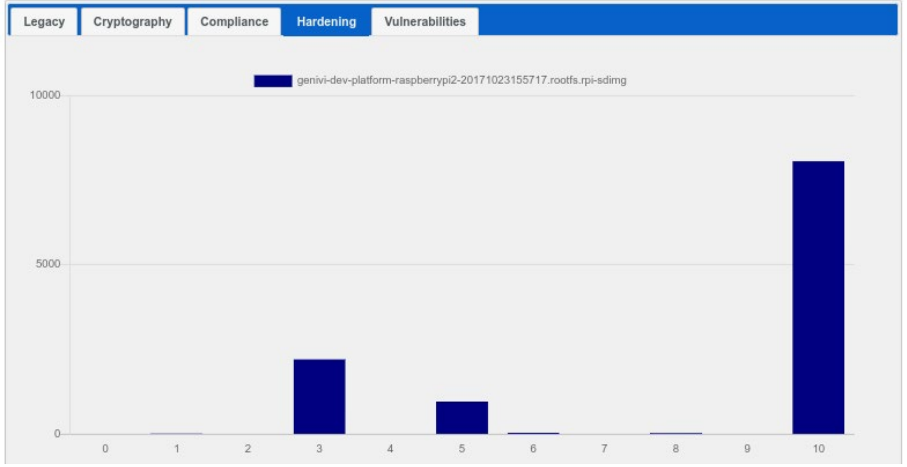
Integrating the central and connected vehicle cockpit.

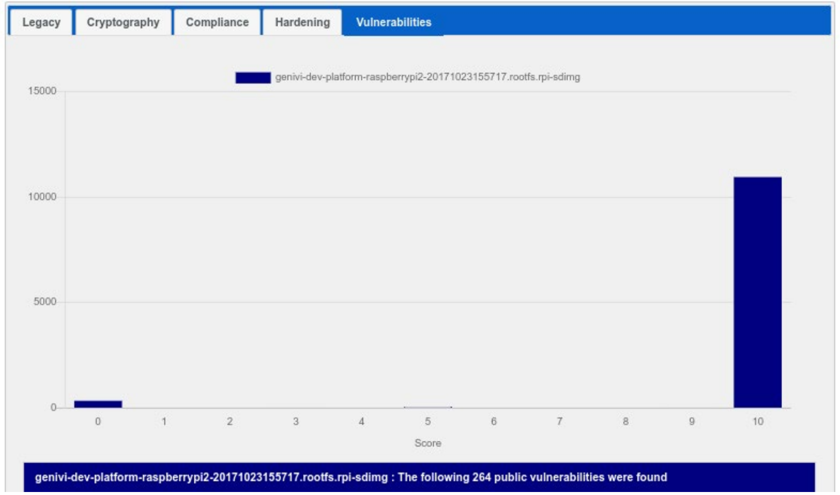
> Multi-OS Integration in the Centralized Vehicle Cockpit

> Future Vehicle Architectures Driving GENIVI Work

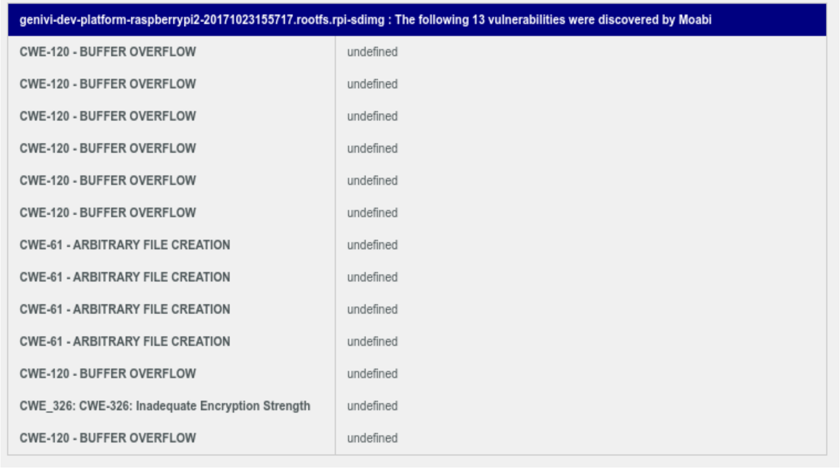
Use case : audit of GenIVI







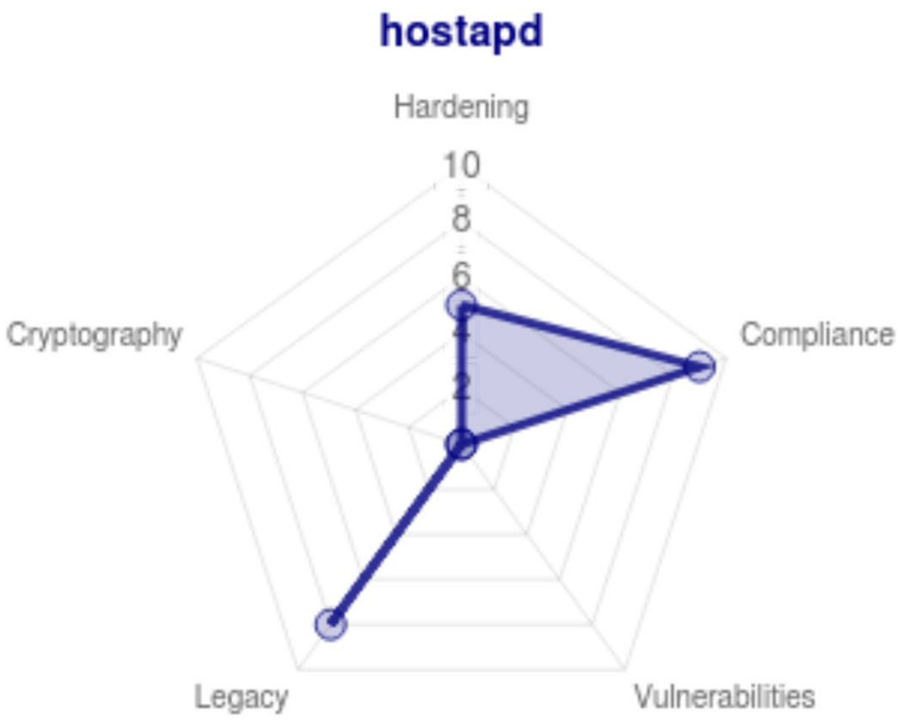
genivi-dev-platform-raspberrypi2-20171023155717.rootfs.rpi-sdimg : The following 264 public vulnerabilities were found



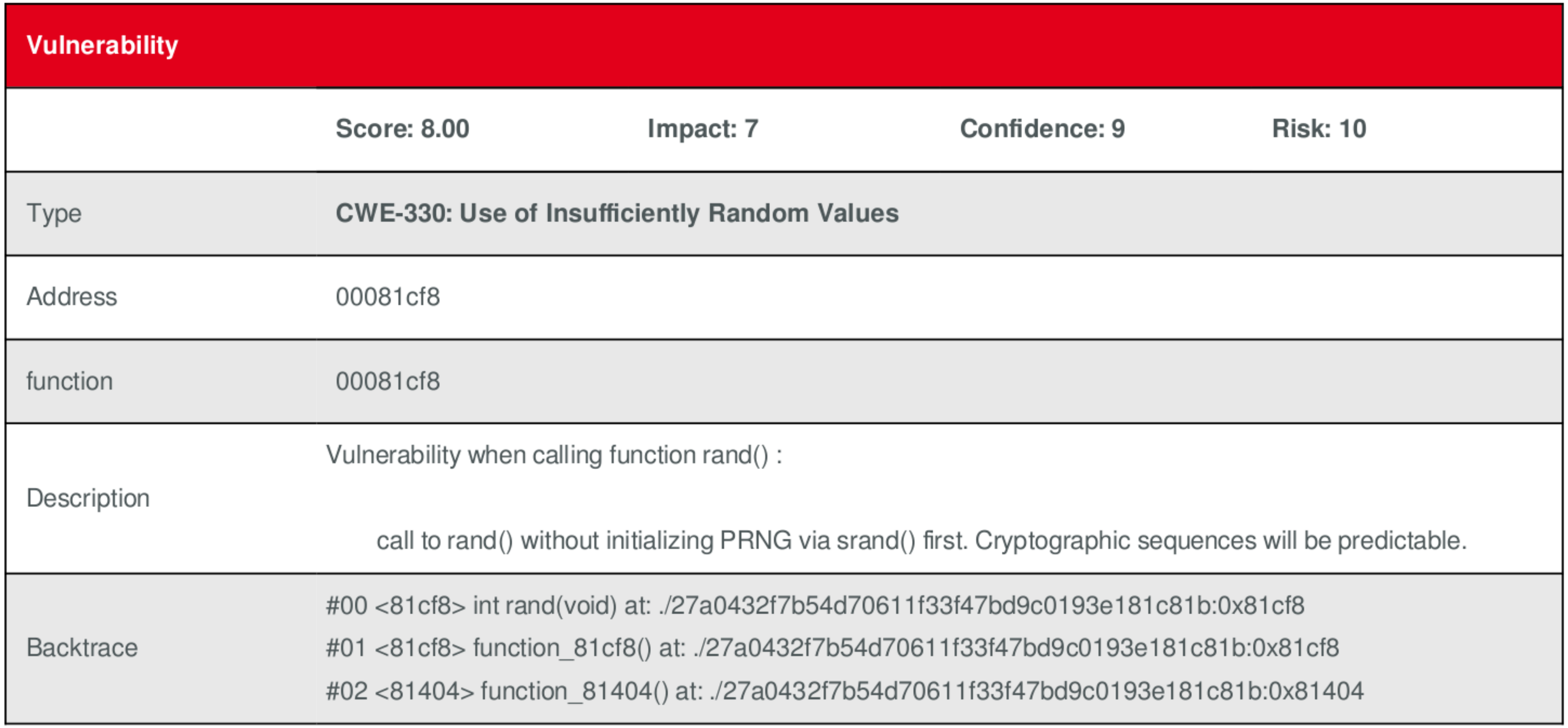
genivi-dev-platform-raspberrypi2-20171023155717.rootfs.rpi-sdimg : The following 13 vulnerabilities were discovered by Moabi

CWE-120 - BUFFER OVERFLOW	undefined
CWE-120 - BUFFER OVERFLOW	undefined
CWE-120 - BUFFER OVERFLOW	undefined
CWE-120 - BUFFER OVERFLOW	undefined
CWE-120 - BUFFER OVERFLOW	undefined
CWE-120 - BUFFER OVERFLOW	undefined
CWE-61 - ARBITRARY FILE CREATION	undefined
CWE-61 - ARBITRARY FILE CREATION	undefined
CWE-61 - ARBITRARY FILE CREATION	undefined
CWE-61 - ARBITRARY FILE CREATION	undefined
CWE-120 - BUFFER OVERFLOW	undefined
CWE_326: CWE-326: Inadequate Encryption Strength	undefined
CWE-120 - BUFFER OVERFLOW	undefined

Hostapd : overview



Hostapd : Pseudo Random Number Generator



Vulnerability				
	Score: 8.00	**Impact: 7**	**Confidence: 9**	**Risk: 10**
Type	**CWE-330: Use of Insufficiently Random Values**			
Address	00081cf8			
function	00081cf8			
Description	Vulnerability when calling function rand() : call to rand() without initializing PRNG via srand() first. Cryptographic sequences will be predictable.			
Backtrace	#00 <81cf8> int rand(void) at: ./27a0432f7b54d70611f33f47bd9c0193e181c81b:0x81cf8 #01 <81cf8> function_81cf8() at: ./27a0432f7b54d70611f33f47bd9c0193e181c81b:0x81cf8 #02 <81404> function_81404() at: ./27a0432f7b54d70611f33f47bd9c0193e181c81b:0x81404			

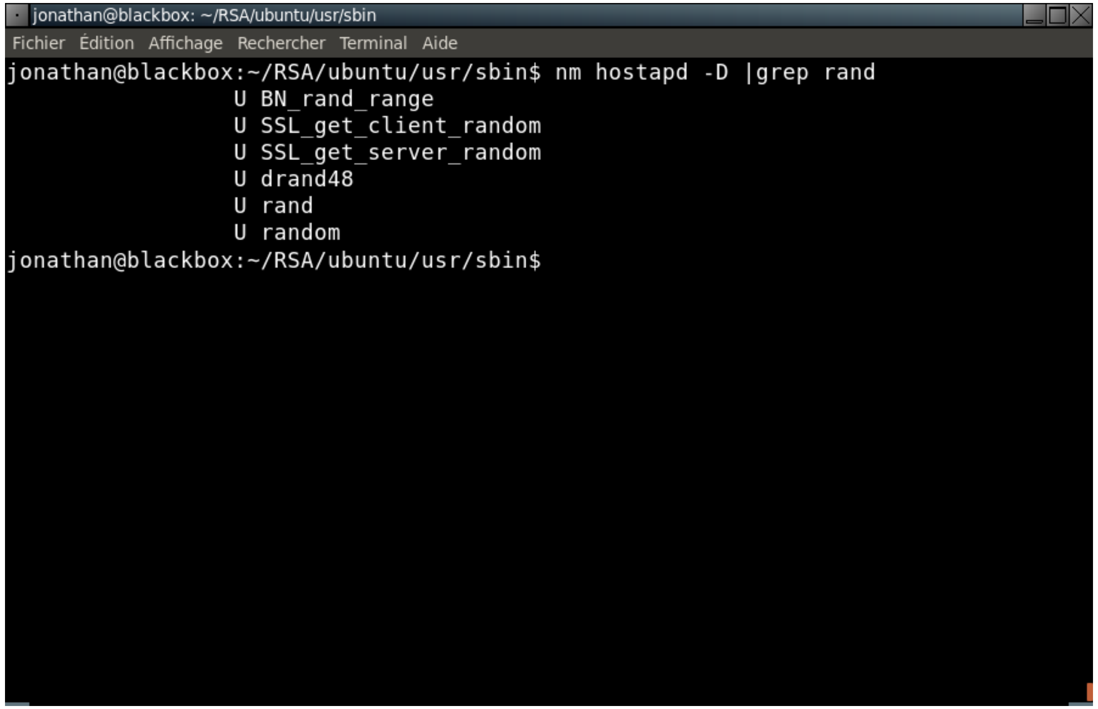
MOABI

RSA Conference2020

Hostapd : CVE-2016-17043

In depth analysis

Hostapd : Pseudo Random Number Generator


jonathan@blackbox: ~/RSA/ubuntu/usr/sbin
Fichier Édition Affichage Rechercher Terminal Aide

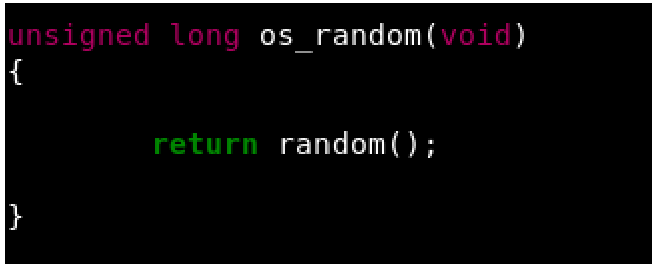
```
jonathan@blackbox:~/RSA/ubuntu/usr/sbin$ nm hostapd -D |grep rand
                 U BN_rand_range
                 U SSL_get_client_random
                 U SSL_get_server_random
                 U drand48
                 U rand
                 U random
jonathan@blackbox:~/RSA/ubuntu/usr/sbin$
```

Use of rand() and random()

without seeding PRNGS

- Reference: https://cwe.mitre.org/data/definitions/330.html
- Silently fixed in version 2.6 (2016)
- Reported to CERT in 03/2019

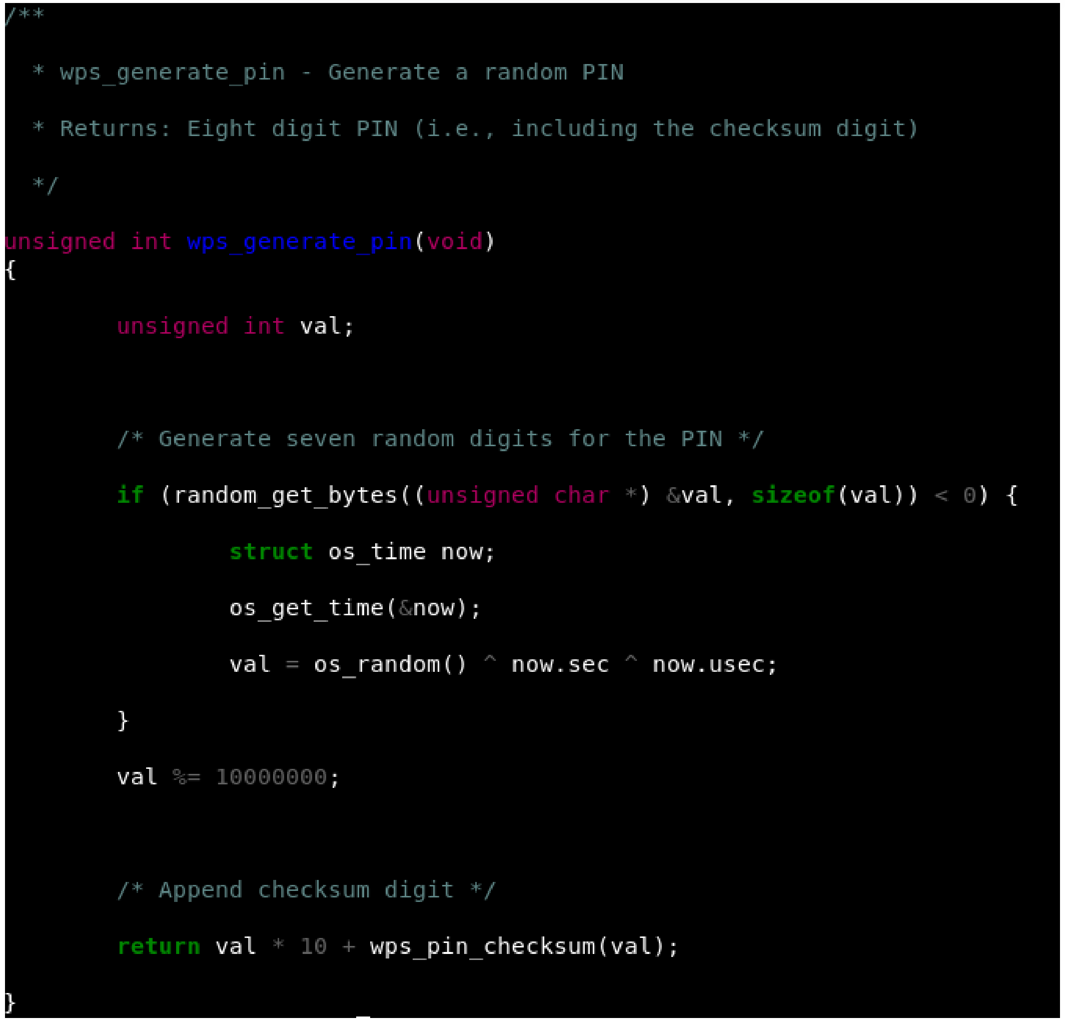
Hostapd : CVE-2016-10743



```
unsigned long os_random(void)
{

        return random();

}
```



```
/**
 * wps_generate_pin - Generate a random PIN
 * Returns: Eight digit PIN (i.e., including the checksum digit)
 */
unsigned int wps_generate_pin(void)
{
        unsigned int val;

        /* Generate seven random digits for the PIN */
        if (random_get_bytes((unsigned char *) &val, sizeof(val)) < 0) {
                struct os_time now;
                os_get_time(&now);
                val = os_random() ^ now.sec ^ now.usec;
        }
        val %= 10000000;

        /* Append checksum digit */
        return val * 10 + wps_pin_checksum(val);
}
```

MOABI

RSAConference2020

Exploit prototyping

Bridging the gap between static analysis and exploitation

From static analysis to reliable exploit : Witchcraft (WCC)

- The binary is ARM
- How do we verify if the vulnerability exists ?
- Let's rely on the Witchcraft Compiler Collection (WCC)

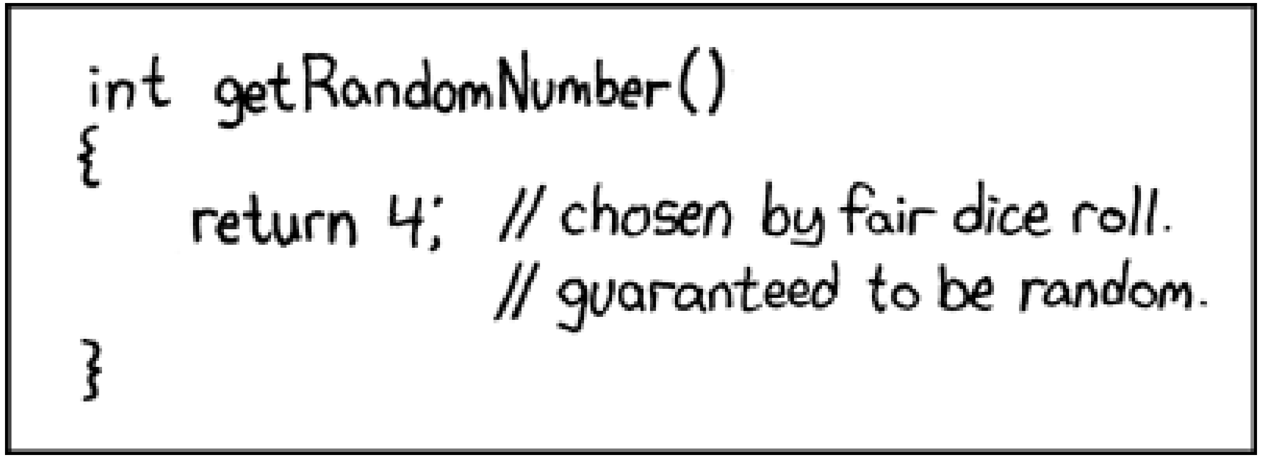
The Witchcraft Compiler Collection (WCC)

- Open Source Software
- Introduced at DEFCON and BLACKHAT 2016
- https://github.com/endrazine/wcc
- Rapid Cross platform analysis
- JIT binary translation from ARM to x86_64 thanks to qemu

DEMO

Witchcraft Compiler Collection

Pseudo Random Number Generator Failure



```
int getRandomNumber()
{
    return 4;  // chosen by fair dice roll.
               // guaranteed to be random.
}
```

Source: https://imgs.xkcd.com/comics/random_number.png

RSA Conference2020

Conclusion

Take away

Apply What You Have Learned Today

- Next week you should:
  - Update hostapd to latest version
- Within six months you should:
  - Identify gaps in your SSDLC to include modern technologies such as static analysis
  - Measure your organization's exposure to firmware vulnerabilities
  - Kickstart a secure supplier relationship management process

RSA Conference2020

Thanks for your attention

Get in touch : info@moabi.com