34C3: tuwat!

Public FPGA based DMA Attacking

UlfFrisk

Agenda

Background and Previous work

Transmit and Receive PCIe TLPs

DUMP memory

FPGA Design

Attack vulnerable vanilla Linux system

Attack vulnerable UEFI → Windows Virtualization Based Security

Future Hardware

About Me: Ulf Frisk

Employed in the financial sector – Stockholm, Sweden

Previously presented at SEC-T and DEF CON

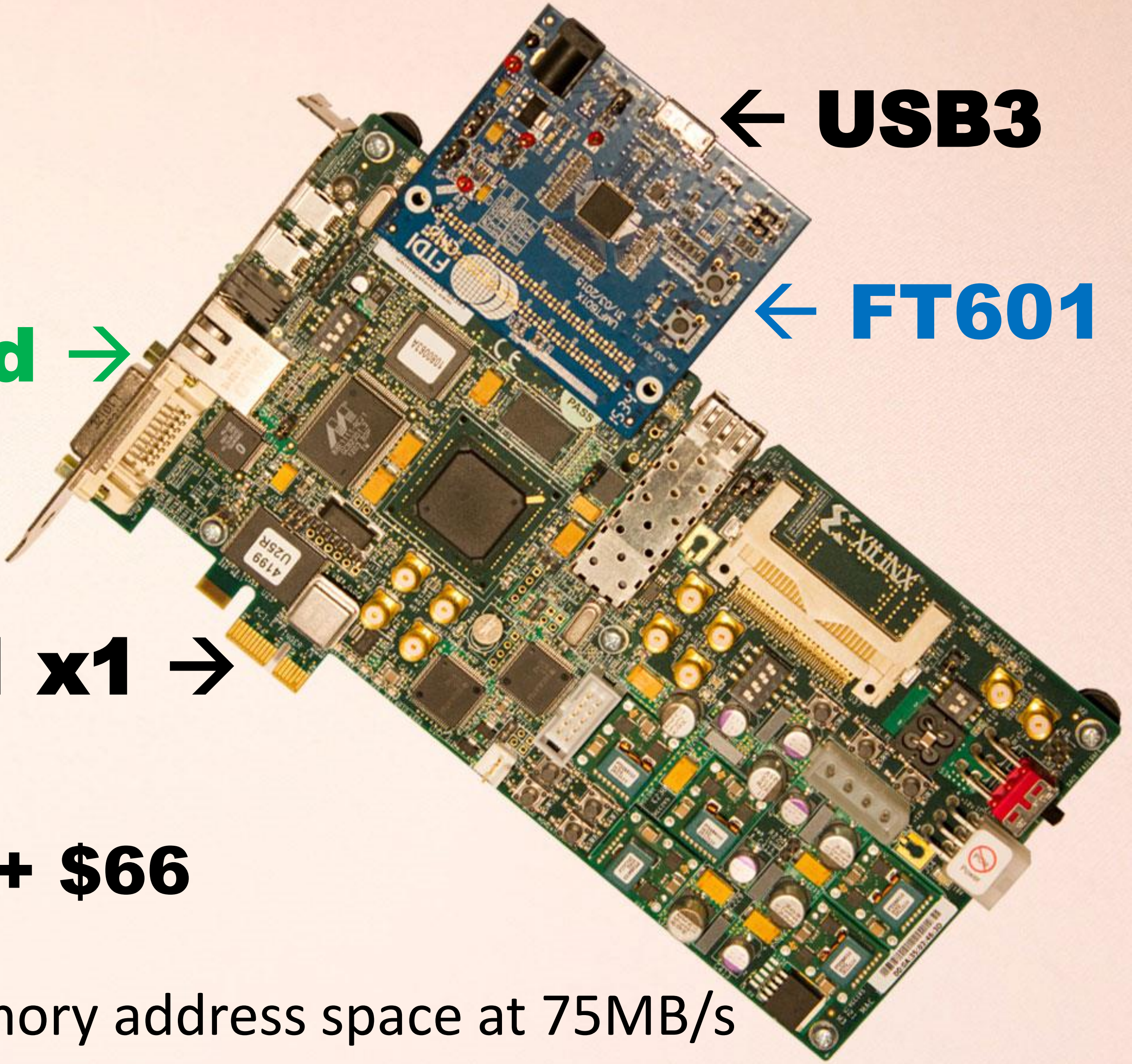
Author of the PCILeech Direct Memory Acccess Attack Toolkit

Hobby Project

Disclaimer

This talk is given by me as an individual
My employer is not involved in any way

PCILeech FPGA



← USB3

← FT601

Xilinx SP605 dev board →

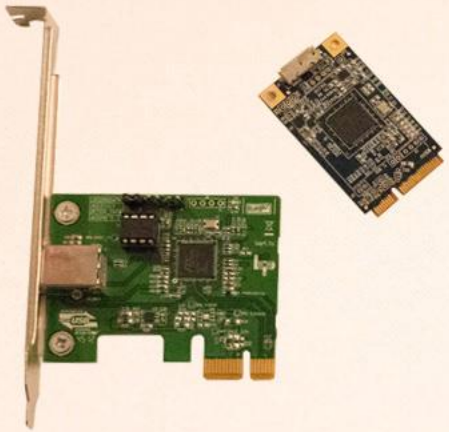
PCIe gen1 x1 →

$495 + $66

DMA to 32-bit and 64-bit memory address space at 75MB/s

Some blobs are vendor proprietary

USB3380 vs SP605



USB3380

Sold Out! (was $195)

Smaller

Faster PCIe gen2 x1 (150MB/s)

Unstable (lock-up on DMA fail)

32-bit DMA addressing only

SP605/FT601

$500-$600

Bulkier

Slower PCIe gen1 x1 (75MB/s)

Stable

64-bit DMA addressing

DMA Attacks

Inception – Firewire DMA attacking

```
v.0.2.0 (C) Carsten Maartmann-Moe 2012
```

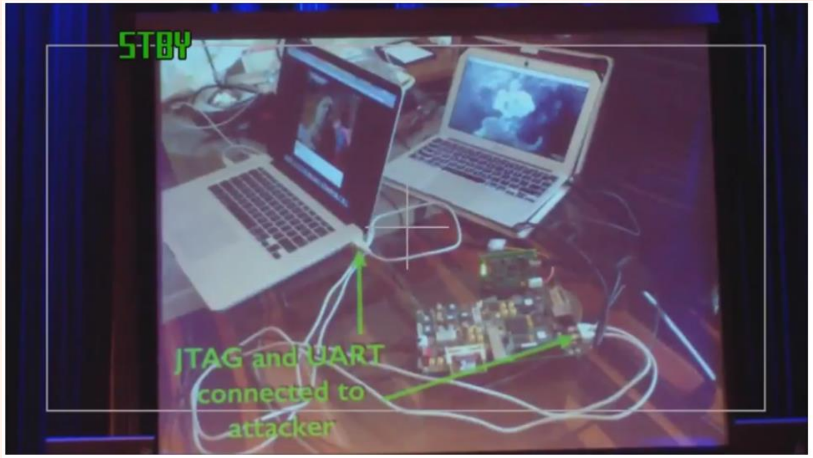
IOMMUs / VT-d introduced >2008

FPGA PCIe DMA academic research

"IronHide" by @_kamino_ in 2010-2012

Thunderbolt PCIe attacking

@snare & rzn used the SP605 in 2014



0x07: Snare - Thunderbolt and lightning, very very frightening

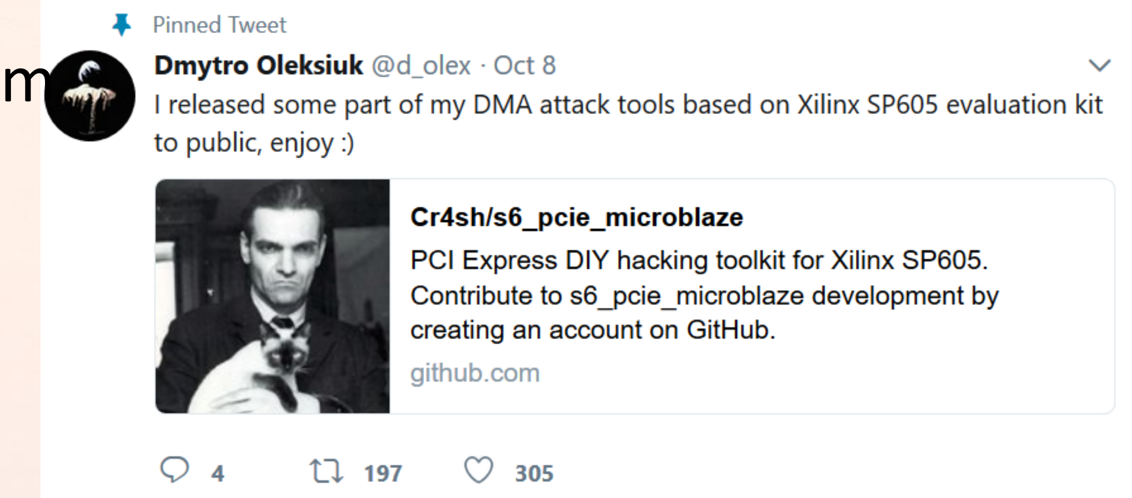
1st Public DMA attack focused FPGA bitstream

By Dmytro Oleksiuk @d_olex – 2017

"PCI Express DIY hacking toolkit"

Also supported by PCILeech

Huge thanks for pushing me to learn Verilog and letting me take early peek at source code!



Pinned Tweet

Dmytro Oleksiuk @d_olex · Oct 8

I released some part of my DMA attack tools based on Xilinx SP605 evaluation kit to public, enjoy :)

Cr4sh/s6_pcie_microblaze

PCI Express DIY hacking toolkit for Xilinx SP605. Contribute to s6_pcie_microblaze development by creating an account on GitHub.

github.com

4 197 305

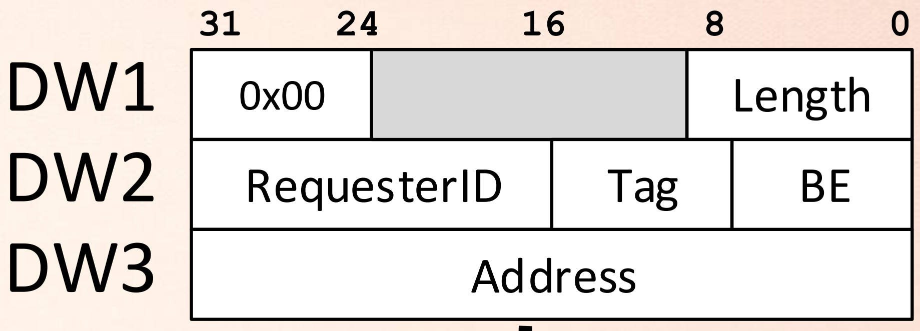
PCIe Transaction Layer Packets / TLPs

DWORD (32-bit) based

Header = 3-4 DWORDs long

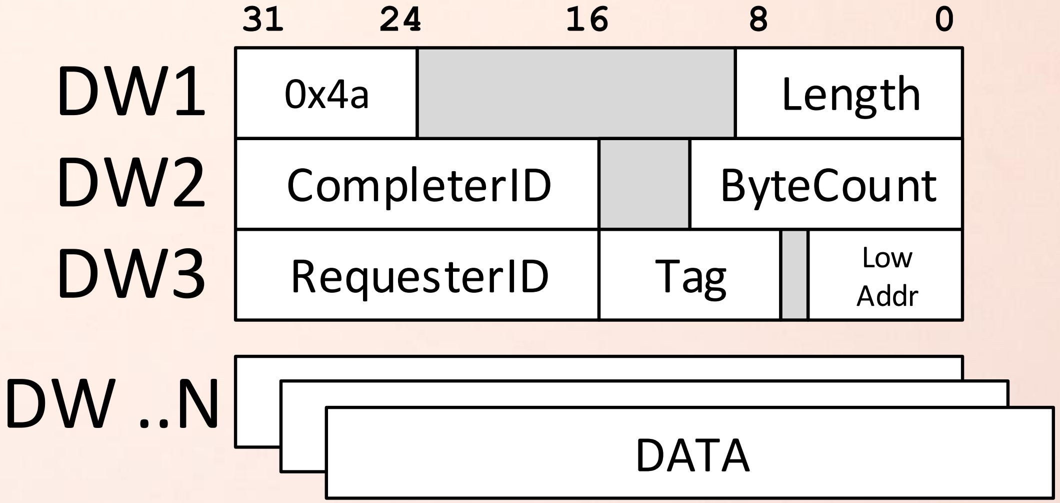
Types: MemRdWr, IO, Cfg, Msg, Cpl, …

32-bit Read TLP



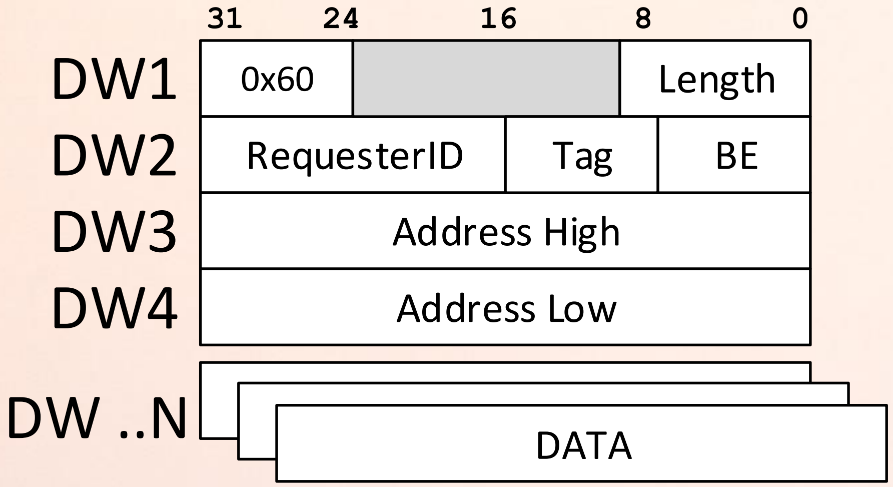
	31	24	16	8	0
DW1	0x00	▨		Length	
DW2	RequesterID		Tag	BE	
DW3	Address				

Completion TLP



	31	24	16	8	0
DW1	0x4a	▨		Length	
DW2	CompleterID		▨	ByteCount	
DW3	RequesterID		Tag	▨	Low Addr
DW ..N	DATA				

64-bit Write TLP

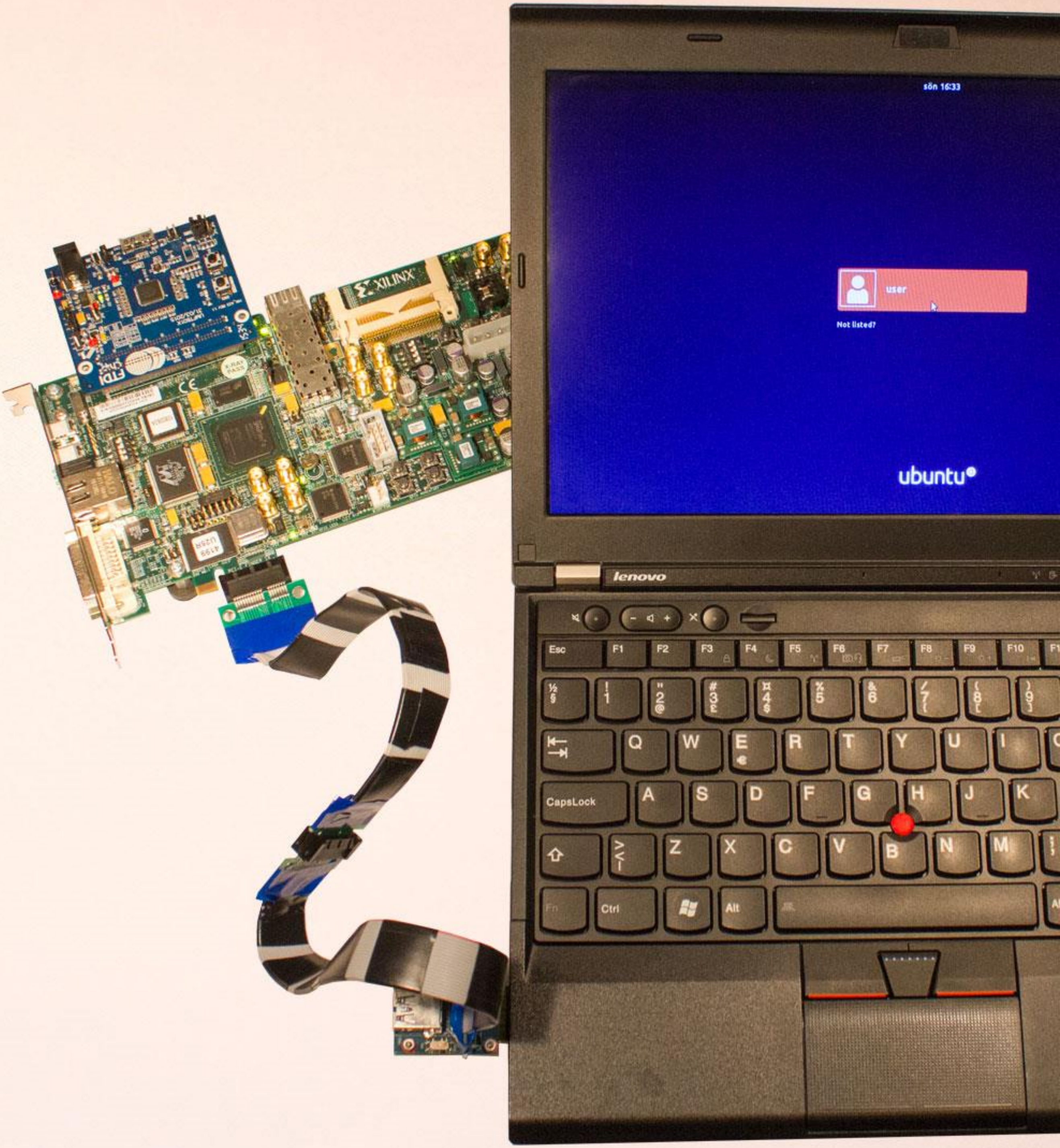


	31	24	16	8	0
DW1	0x60	▨		Length	
DW2	RequesterID		Tag	BE	
DW3	Address High				
DW4	Address Low				
DW ..N	DATA				

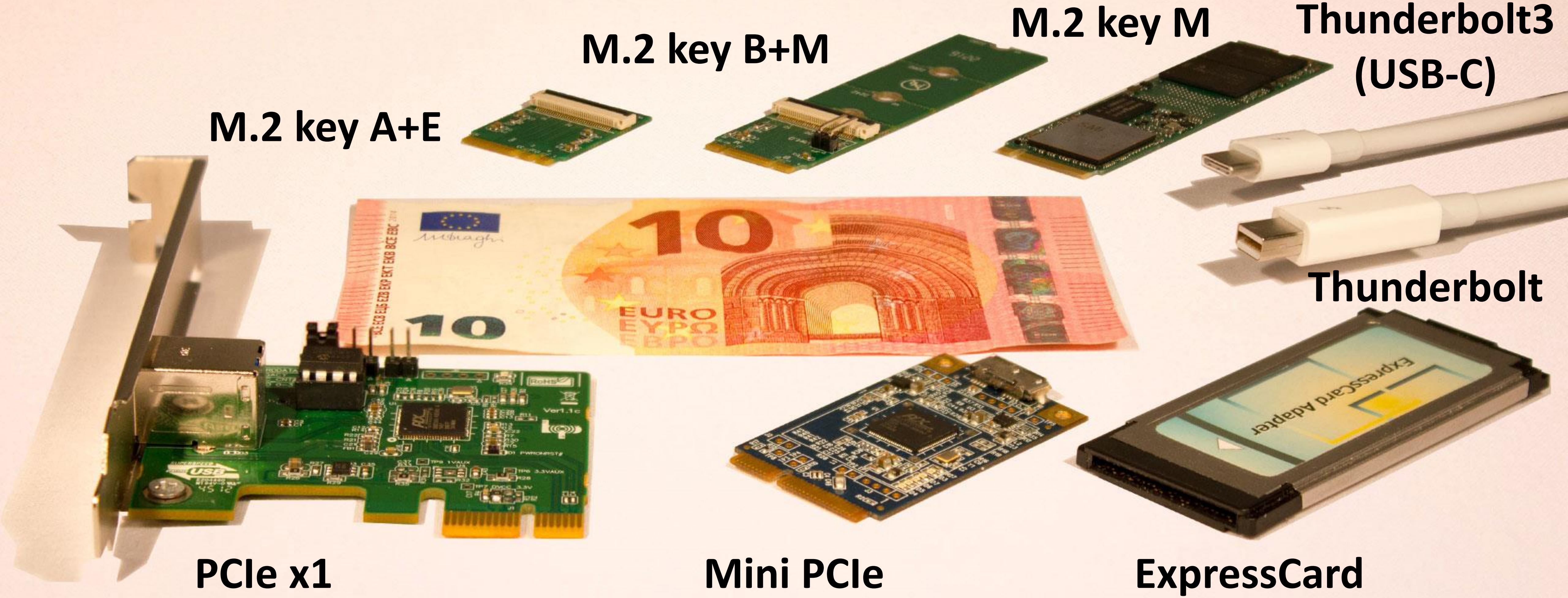
DEMO

Transmit and
Receive PCIe TLPs

Enumerate Memory
Dump Memory



PCI Express Form Factors



M.2 key A+E

M.2 key B+M

M.2 key M

Thunderbolt3 (USB-C)

Thunderbolt

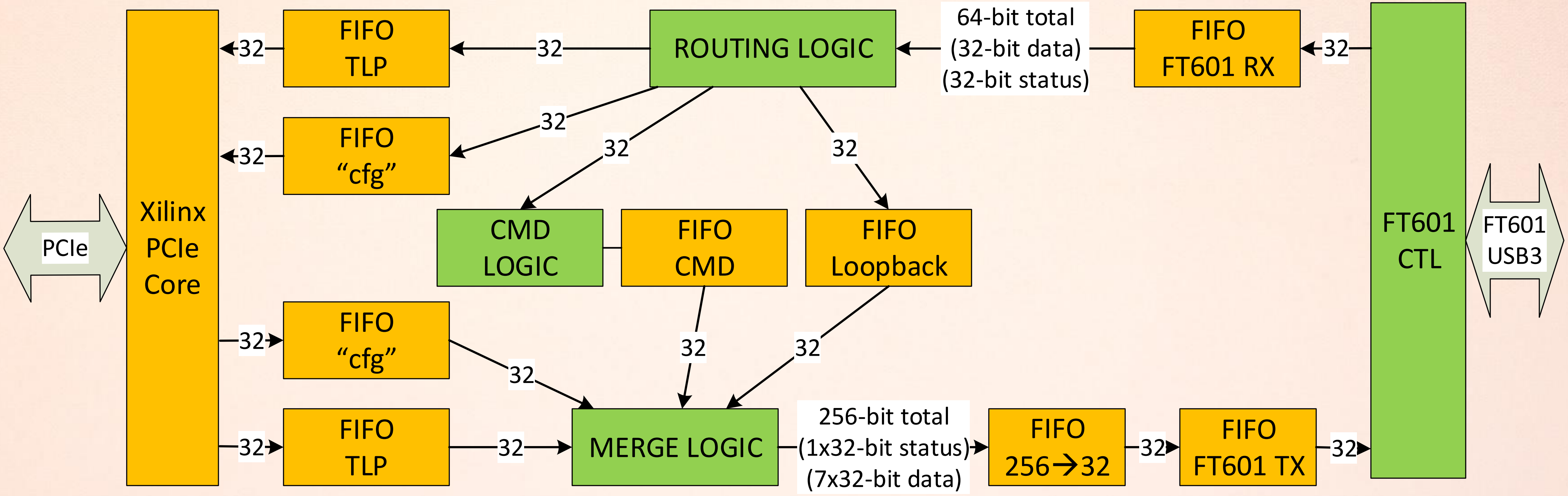
PCIe x1

Mini PCIe

ExpressCard

Everything here is PCI Express in different form factors and variations.

FPGA Design



PCIe

Xilinx PCIe Core

FIFO TLP

FIFO “cfg”

ROUTING LOGIC

64-bit total (32-bit data) (32-bit status)

FIFO FT601 RX

FT601 CTL

FT601 USB3

CMD LOGIC

FIFO CMD

FIFO Loopback

FIFO “cfg”

FIFO TLP

MERGE LOGIC

256-bit total (1x32-bit status) (7x32-bit data)

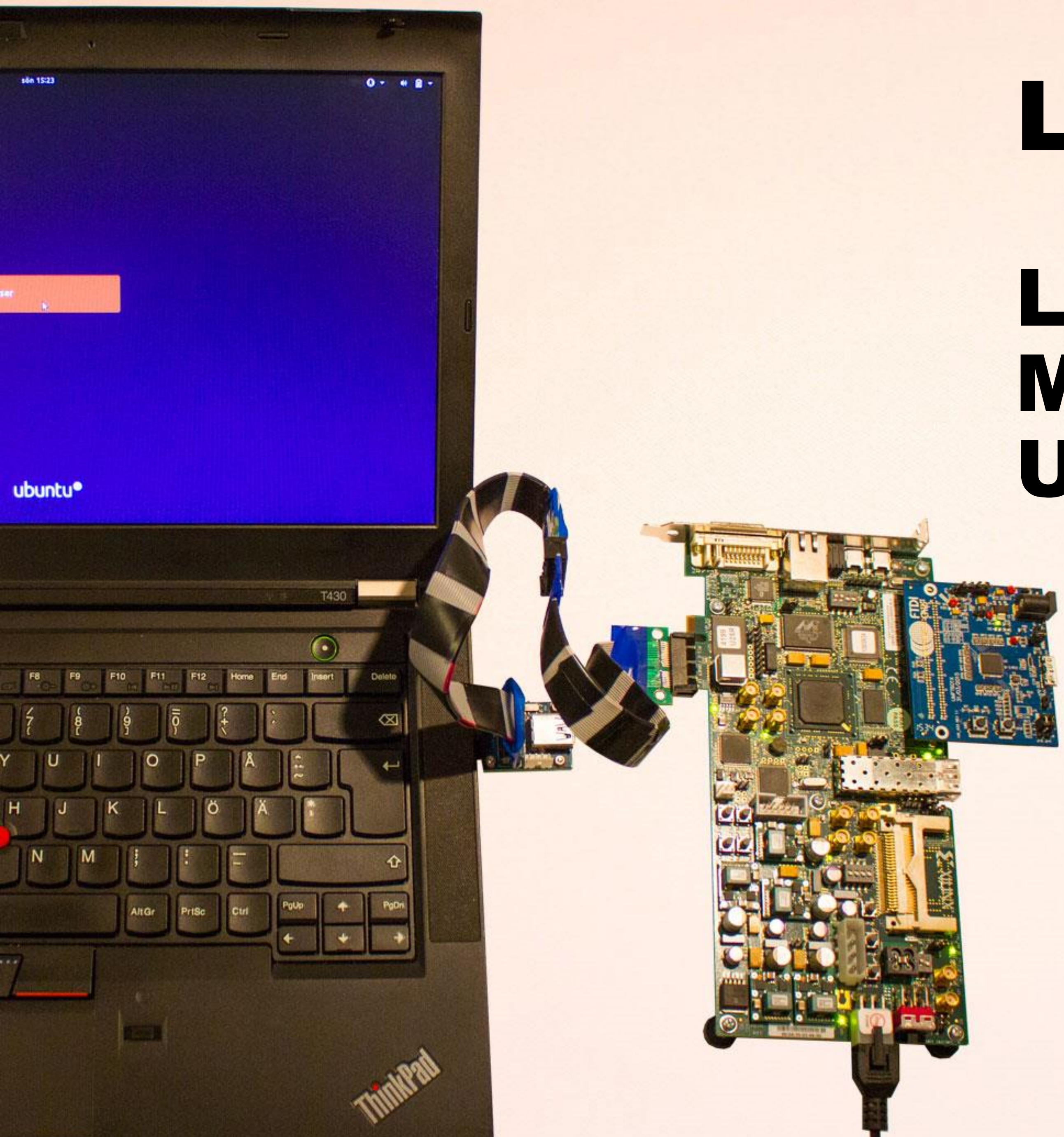
FIFO 256→32

FIFO FT601 TX

32

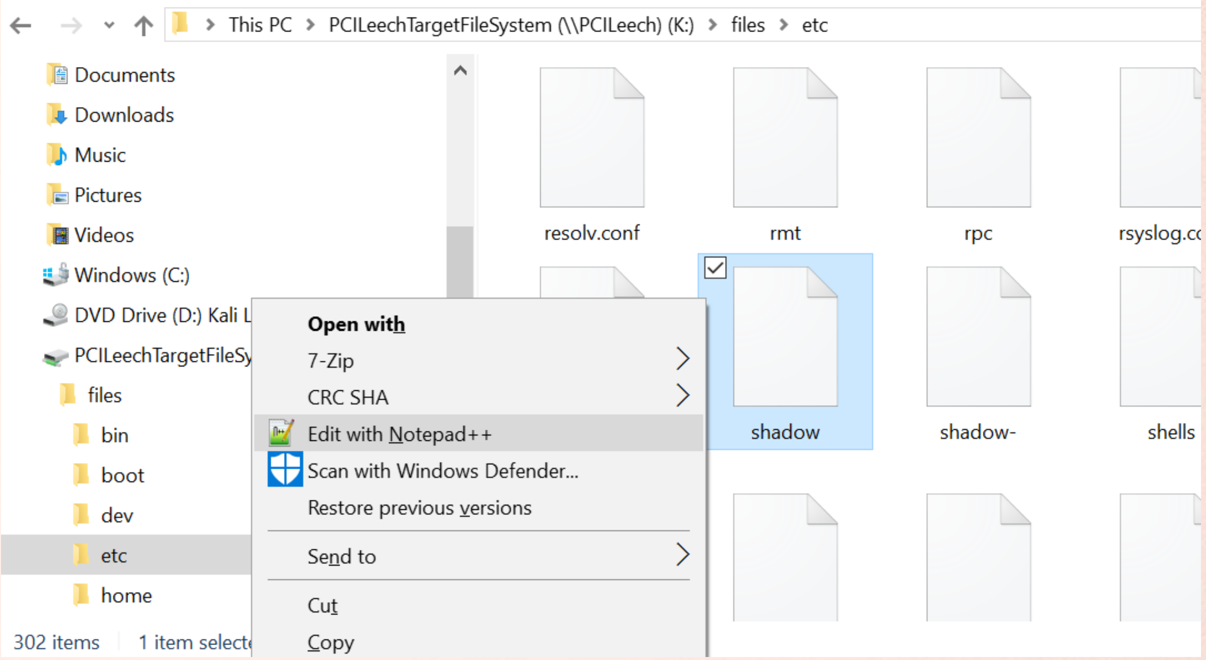
🟧 = **Xilinx IP-blocks**

🟩 = **Open PCILeech modules/logic**



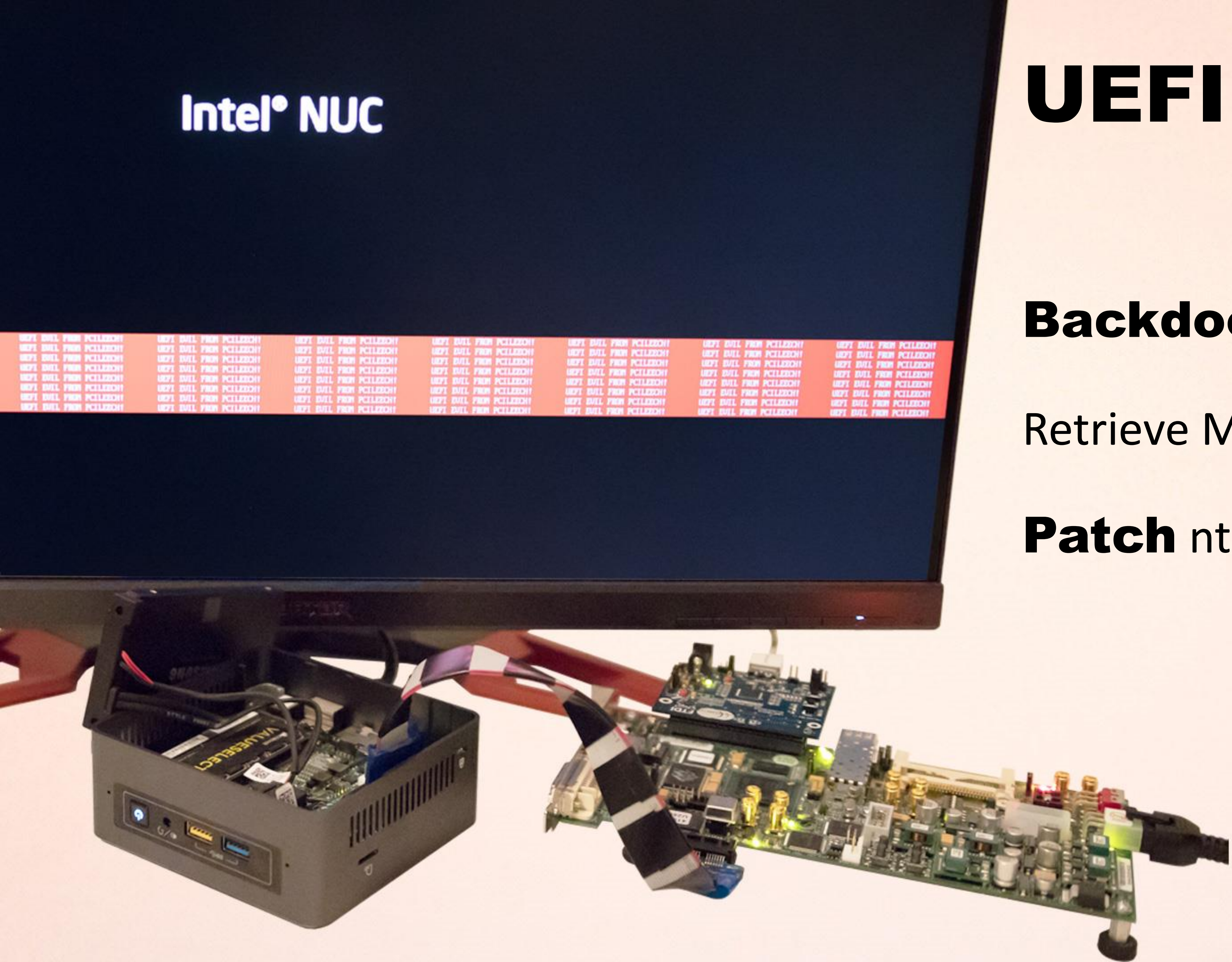
LINUX DEMO

Locate and **Patch** kernel
Mount file system
Unlock (edit /etc/shadow)



LINUX IS SECURE/INSECURE DEPENDING ON CONFIGURATION AND DISTRIBUTION …

UEFI DEMO



Backdoor ExitBootServices

Retrieve Memory Map

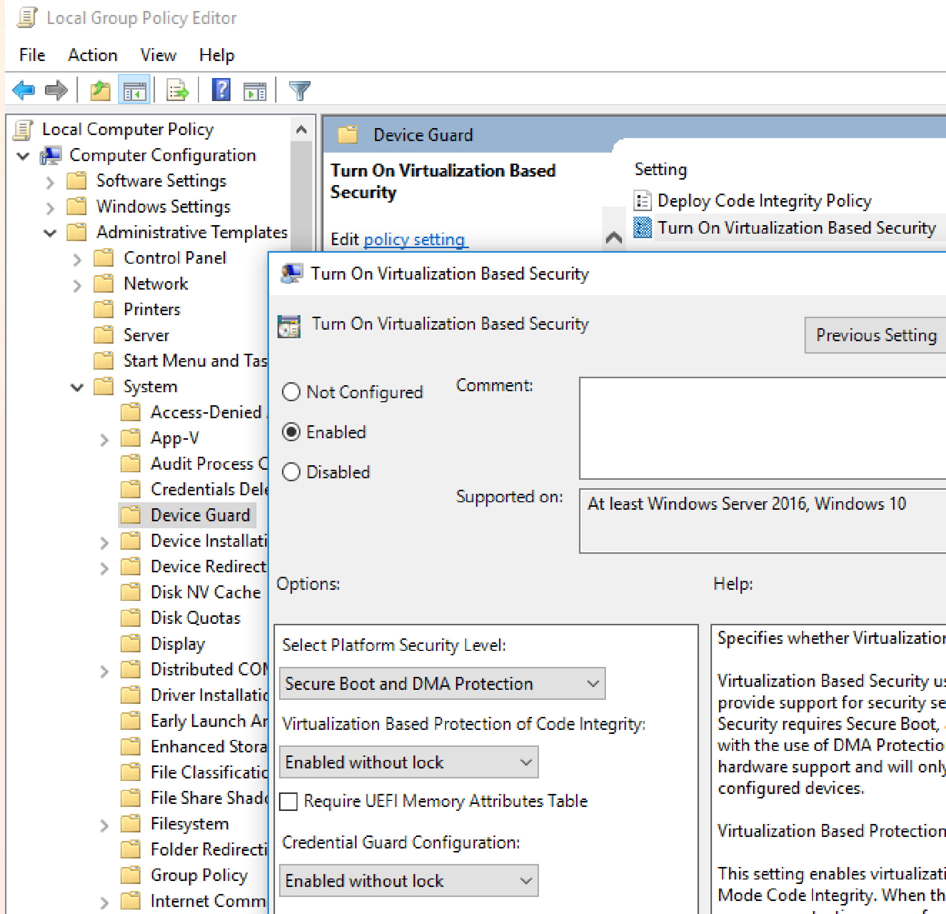
Patch ntoskrnl.exe

Windows Virtualization Based Security (VBS)

Protection of Kernel Code Integrity with help of hypervisor & secure kernel

DMA access to memory:
Hypervisor and Secure Kernel memory == no access
Normal executable pages == read only
Normal non-executable pages == read/write

VBS code integrity not yet enabled in winload.efi stage
(kernel & hypervisor not yet started)



Turn On Virtualization Based Security

Edit policy setting

Turn On Virtualization Based Security

Turn On Virtualization Based Security

○ Not Configured
● Enabled
○ Disabled

Comment:

Supported on: At least Windows Server 2016, Windows 10

Options:

Select Platform Security Level: Secure Boot and DMA Protection

Virtualization Based Protection of Code Integrity: Enabled without lock

□ Require UEFI Memory Attributes Table

Credential Guard Configuration: Enabled without lock

Help:

Specifies whether Virtualizatio

Virtualization Based Security u
provide support for security se
Security requires Secure Boot,
with the use of DMA Protectio
hardware support and will only
configured devices.

Virtualization Based Protection

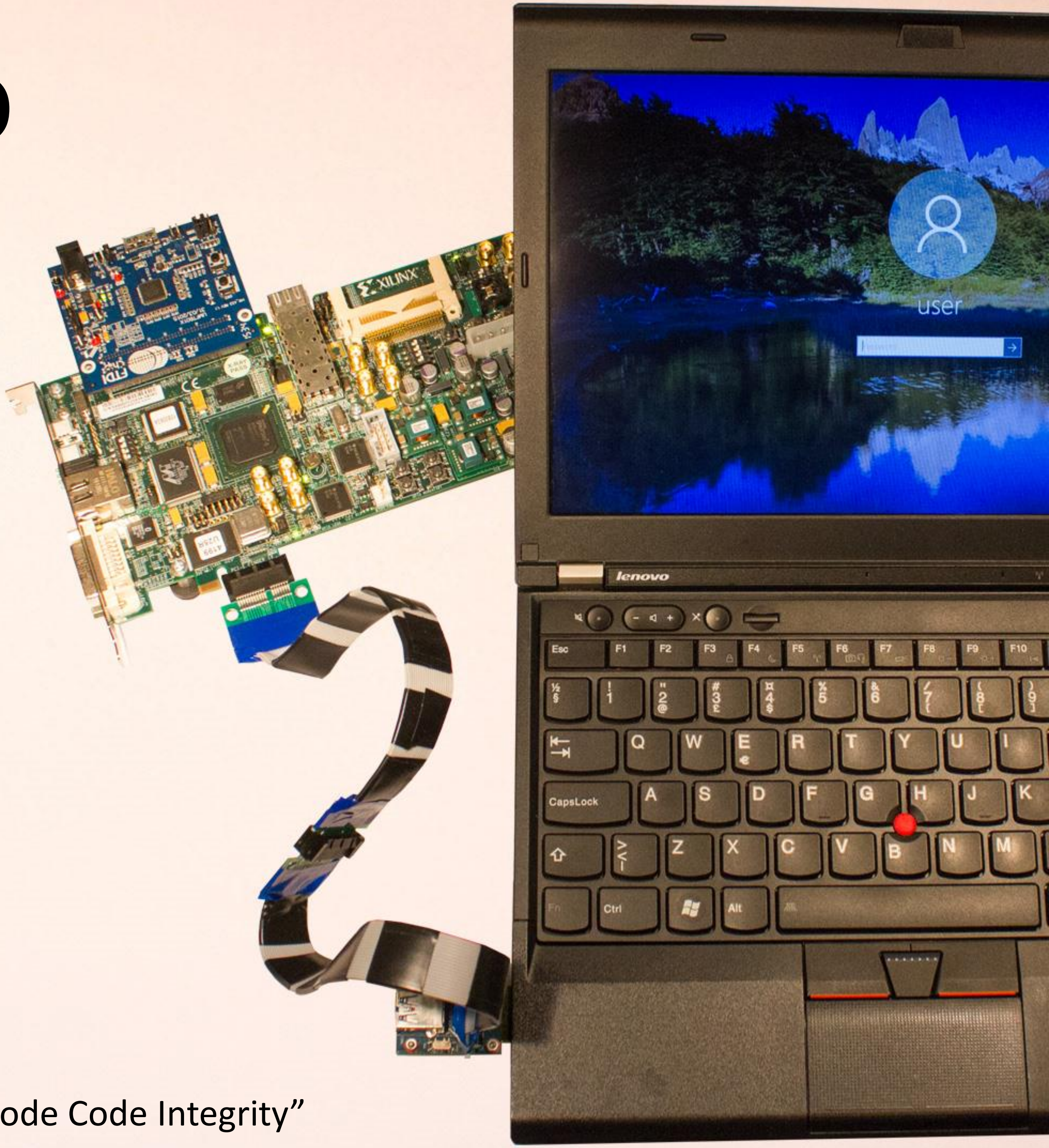
This setting enables virtualizat
Mode Code Integrity. When th

WINDOWS DEMO

Bypass VBS* from compromised **UEFI**

Excute Code and **Spawn Shell**

Dump memory



*) Virtualization Based Security, ”Device Guard” with ”Kernel Mode Code Integrity”

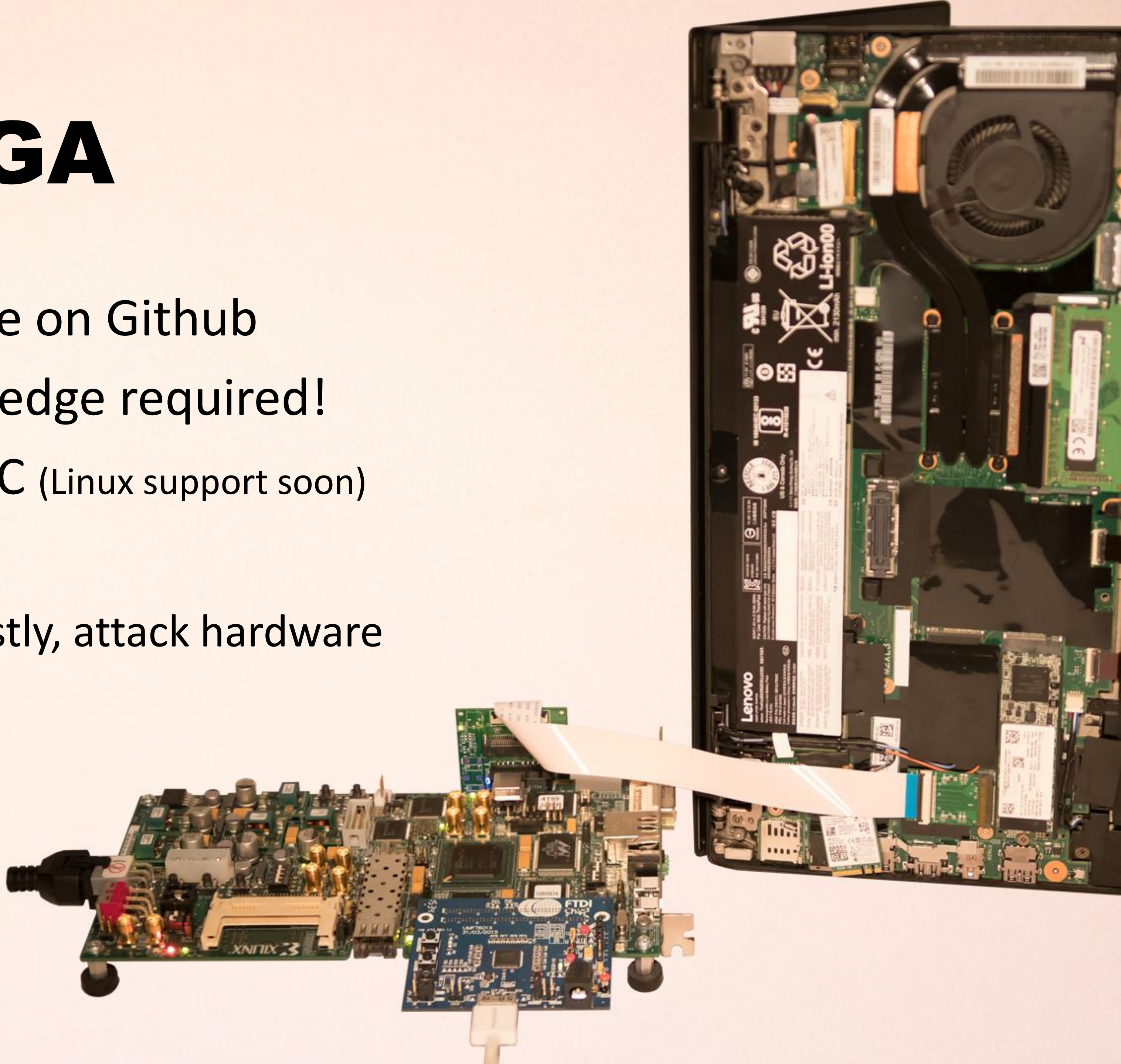
PCILeech FPGA

Source and binaries available on Github

Easy to use! No FPGA knowledge required!

Windows only on attacker PC (Linux support soon)

Future support for more, less costly, attack hardware



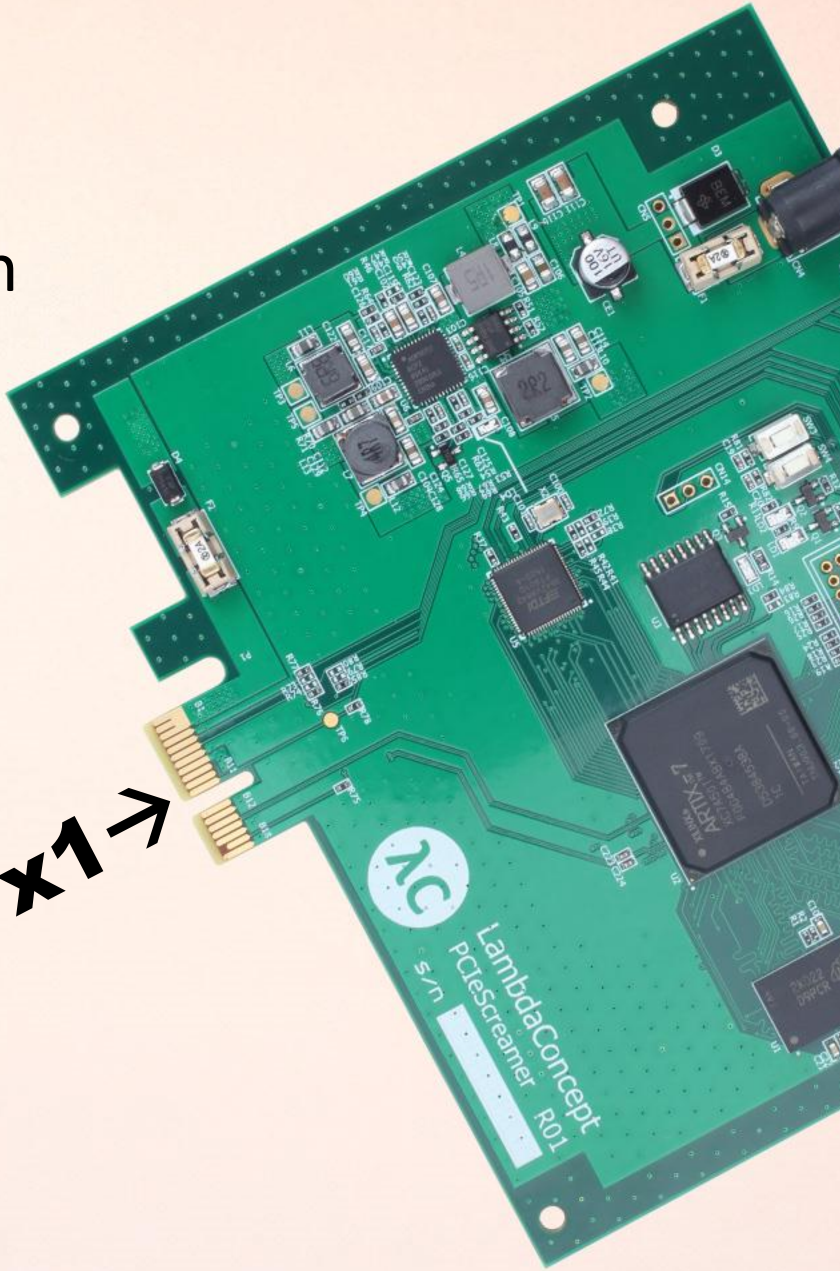
PCIeScreamer

New HW by @key2fr - Ramtin Amin

Easier to use
less costly
more capable

PCILeech support

Early 2018



Summary

Affordable FPGA DMA attacking is the reality of today!

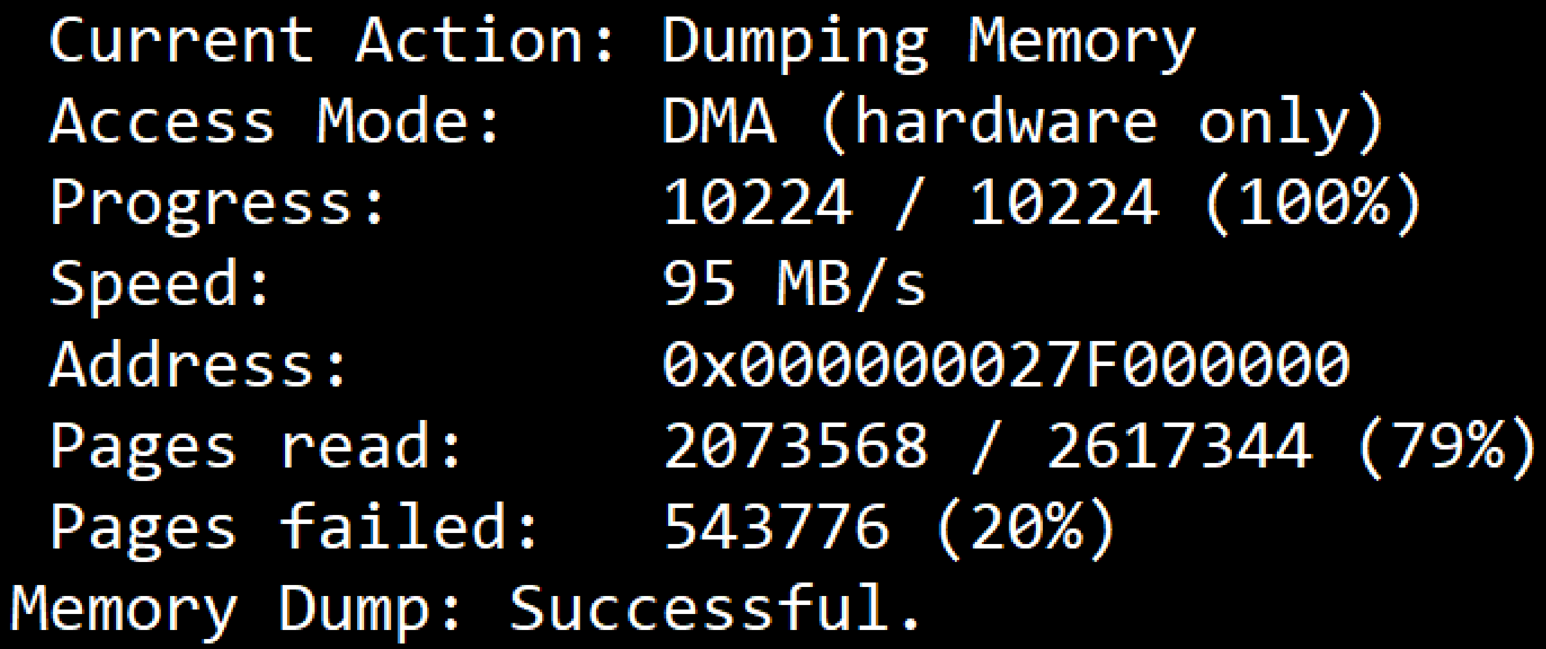
Physical Access is still an issue

IOMMUs are there but they might not be used!

More **research to be done** in the area

Hopefully my tools will be useful

Thank You!



```
 Current Action: Dumping Memory
 Access Mode:    DMA (hardware only)
 Progress:       10224 / 10224 (100%)
 Speed:          95 MB/s
 Address:        0x000000027F000000
 Pages read:     2073568 / 2617344 (79%)
 Pages failed:   543776 (20%)
Memory Dump: Successful.
```

github.com/ufrisk/pcileech-fpga

🐦 UlfFrisk