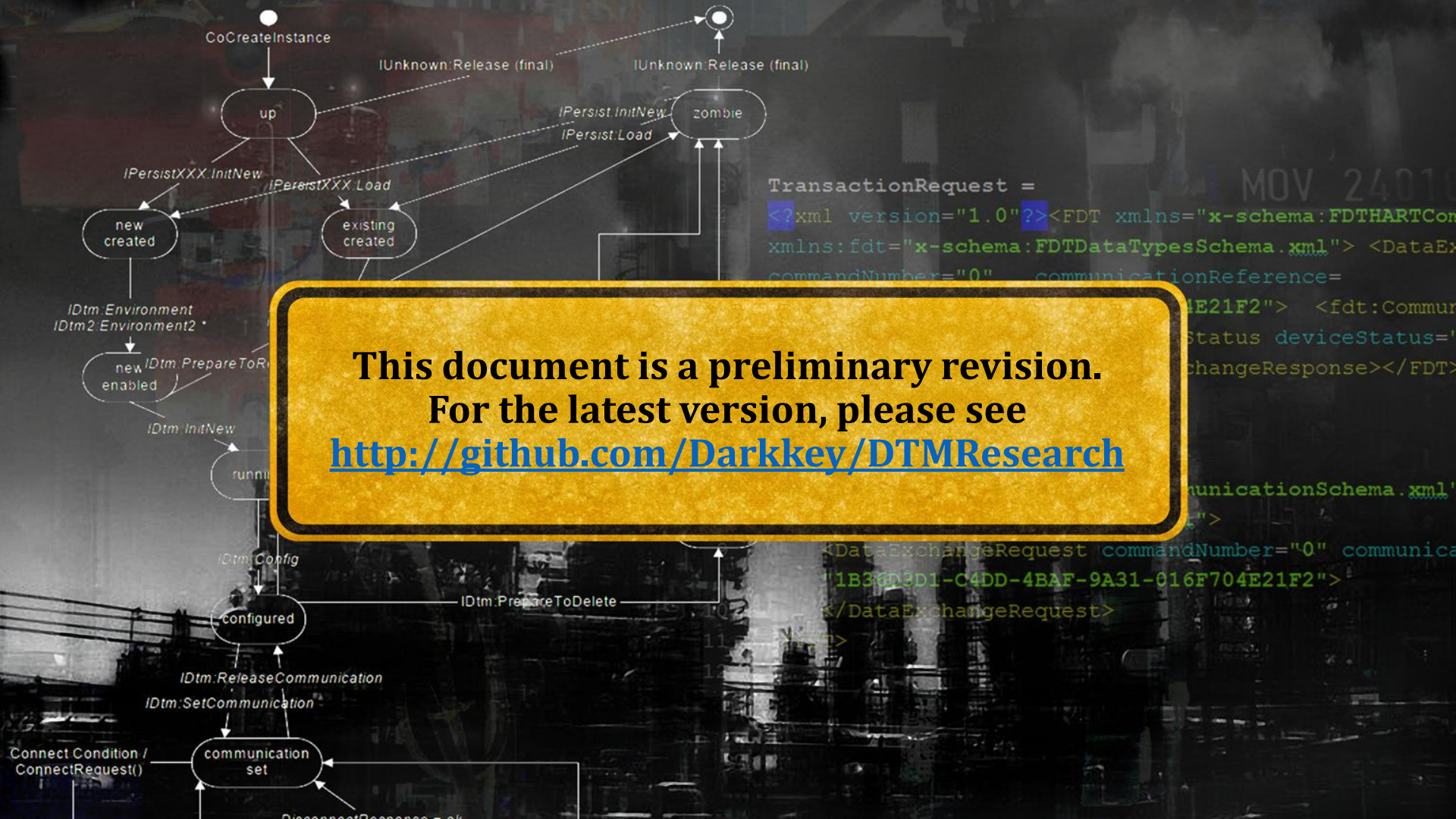
DTM COMPONENTS: SHADOW KEYS TO THE ICS KINGDOM

Alexander @dark_k3y Bolshev
Gleb @cherboff Cherbov

Digital Security

black hat EUROPE 2014



This document is a preliminary revision.
For the latest version, please see
http://github.com/Darkkey/DTMResearch

whoami: dark_k3y

Alexander Bolshev (@dark_key)

IS auditor @ Digital Security

Ph.D.

Assistant Professor @ SPbETU

Distributed systems researcher

Yet another man wearing “some-color hat”

DTM COMPONENTS: SHADOW KEYS TO THE ICS KINGDOM

whoami: cherboff

Gleb Cherbov (@cherboff)

IS researcher @ Digital Security

Information security researcher

DTM COMPONENTS: SHADOW KEYS TO THE ICS KINGDOM

Agenda

- Introduction to FDT/DTM
- Research scope
- Fuzzing technologies
- Vulnerabilities and weaknesses statistics
- Vulns && funny things
- FDT 2.0
- Conclusions

Intro to FDT/DTM

ICS 101

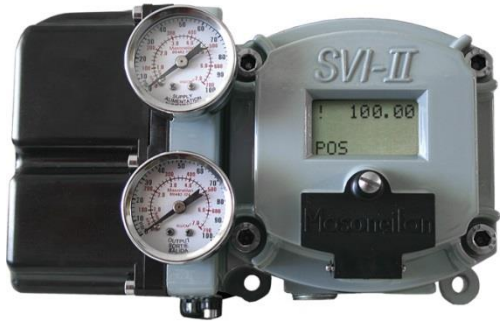
- ICS stands for Industrial Control System.
- Today, ICS infrastructures are commonly used in every factory and even in your house, too!
- ICS collects data from remote stations (also called field devices), processes them, and uses automated algorithms or operator-driven supervisory to create commands to be sent back.
- Thousands of field devices could exist at one facility.
- To control them, Plant Asset Management Systems (PAS or AMS) were invented.
- Plant Assets Management Software = tools for managing plants assets, that lie on the upper/medium levels of ICS and control/monitor/configure field devices.

DTM COMPONENTS: SHADOW KEYS TO THE ICS KINGDOM

Field protocols

- HART (current loop, 4-20 mA)
- Profibus DP (RS-485)
- Profibus PA (MBP)
- Modbus (RS-485)
- Foundation Fieldbus H1 (MBP)
- …

Field devices



DTM COMPONENTS: SHADOW KEYS TO THE ICS KINGDOM

What is FDT/DTM?

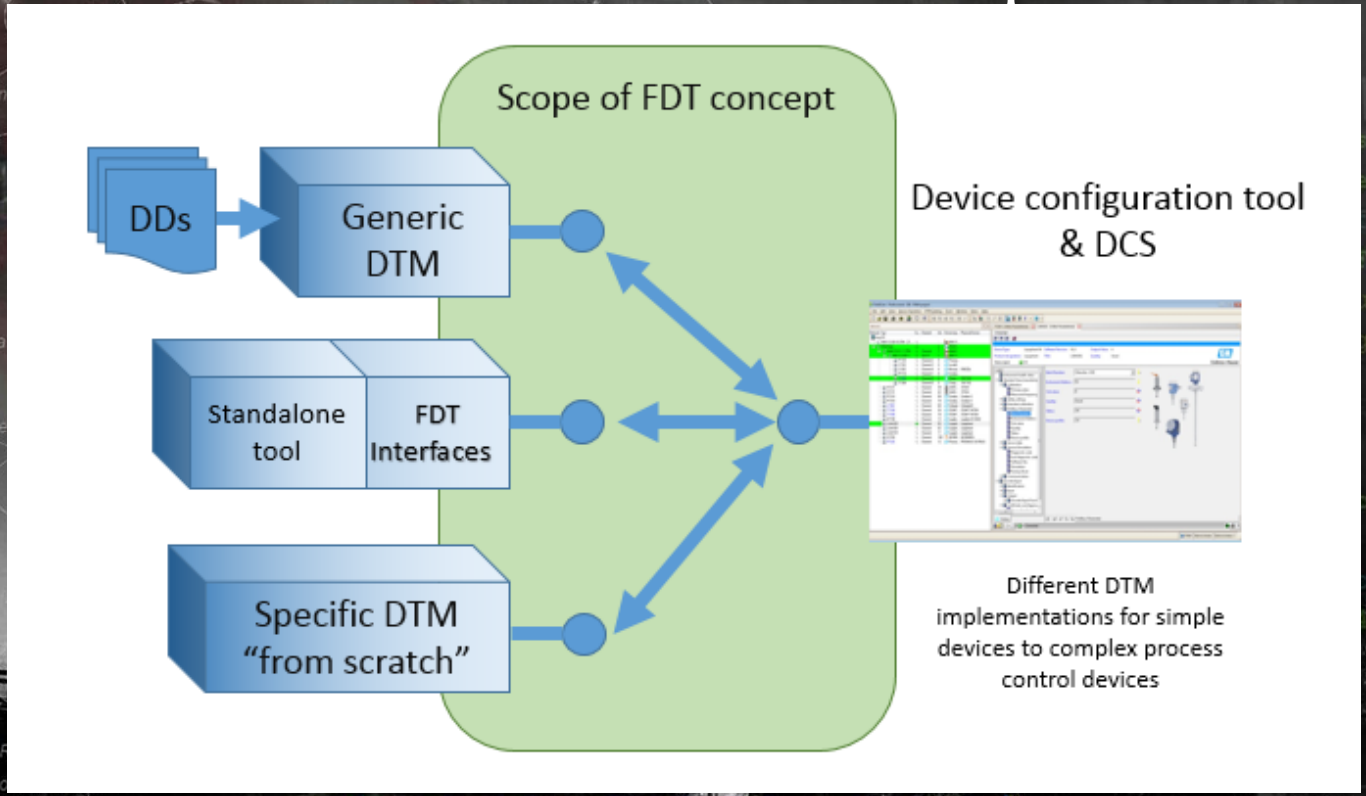
- “The FDT concept defines the interfaces between device-specific software components provided by the device supplier and the engineering tool of the control system manufacturer. The device-specific software component is called DTM (Device Type Manager).” © FDT Group, maintainer of FDT/DTM specification

In short:

- FDT standardizes the communication and configuration interface between all field devices and host systems
- DTM provides a unified structure for accessing device parameters, configuring and operating the devices, and diagnosing problems

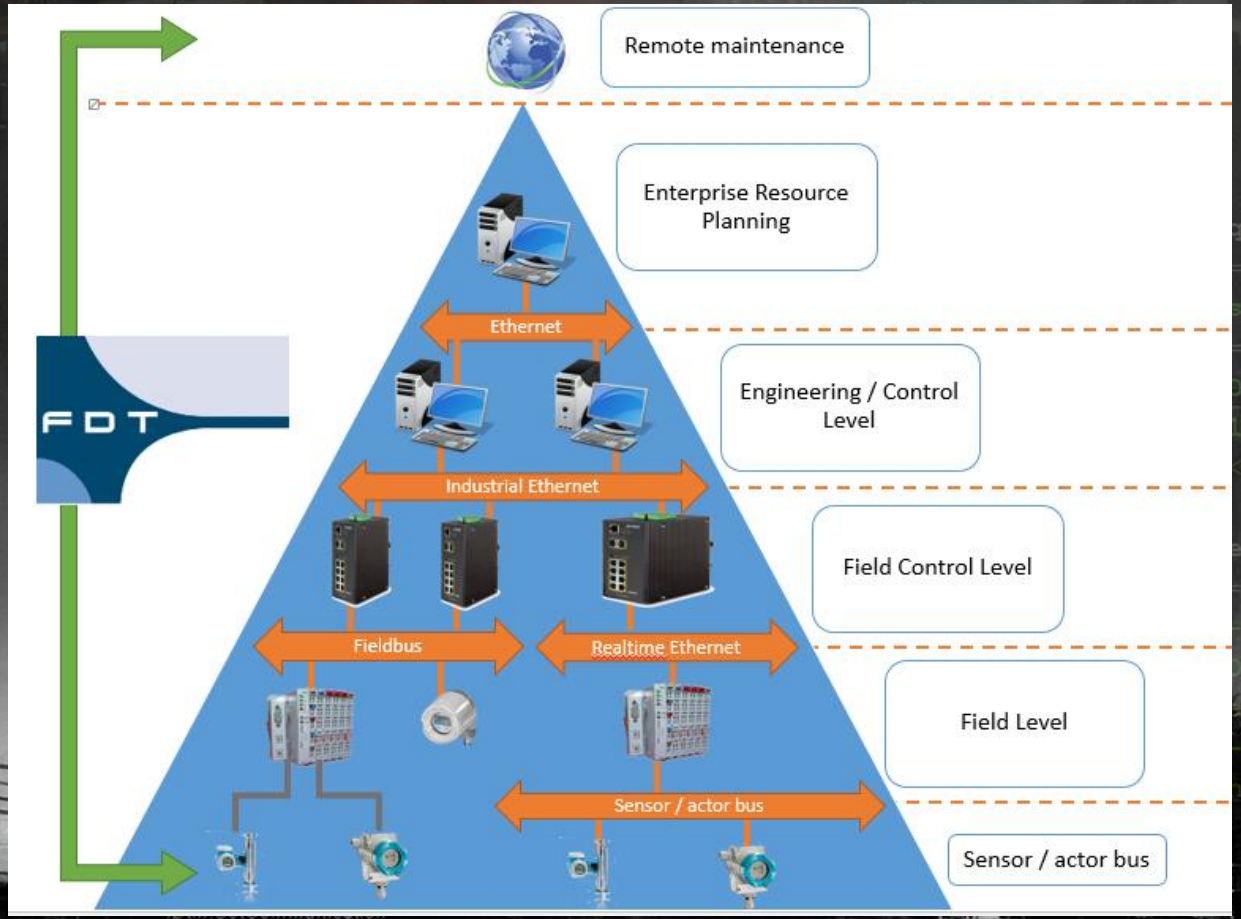
DTM COMPONENTS: SHADOW KEYS TO THE ICS KINGDOM

DTM implementations



DTM COMPONENTS: SHADOW KEYS TO THE ICS KINGDOM

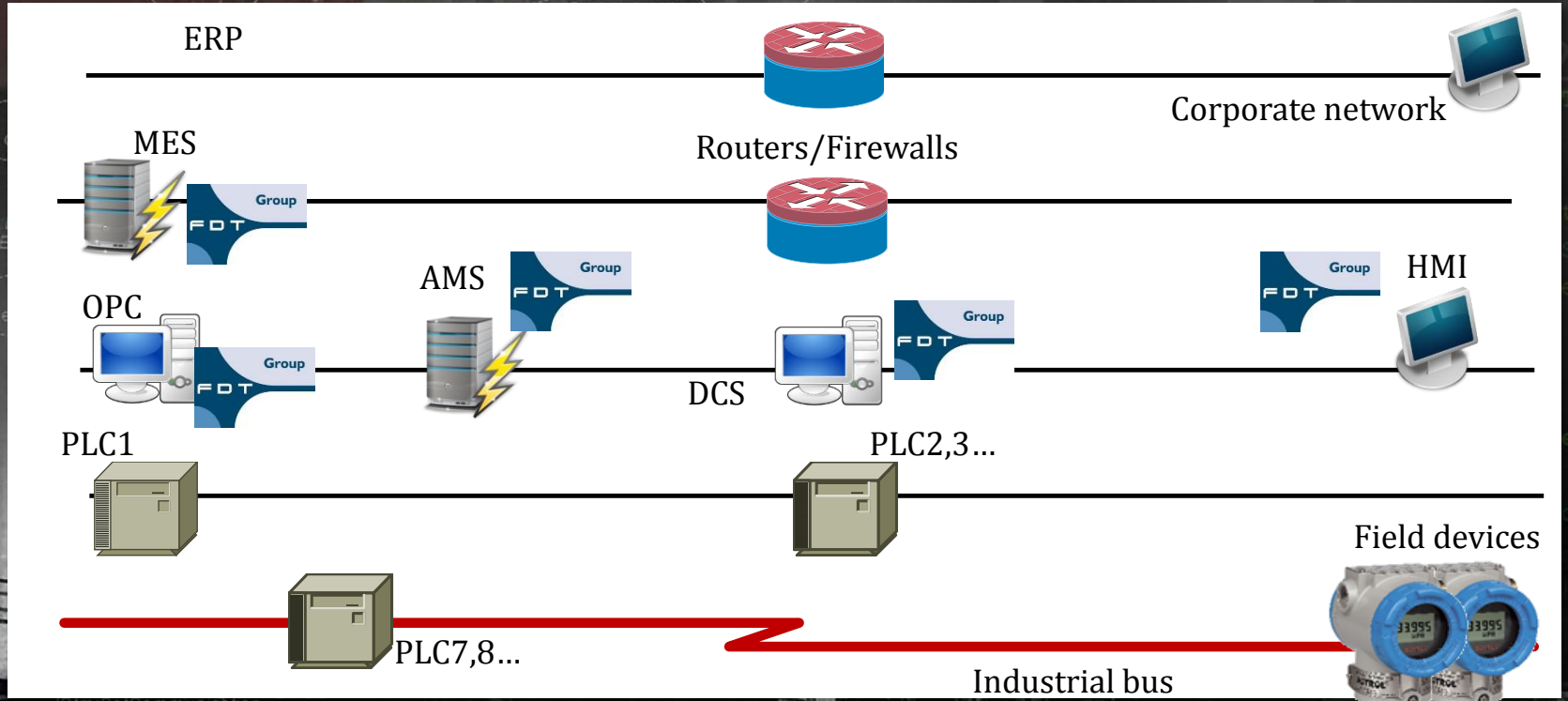
FDT/DTM layers*



*Picture from http://www.automationworld.com/fdt-group-wants-your-input-yes-yours

DTM COMPONENTS: SHADOW KEYS TO THE ICS KINGDOM

Typical places of DTMs in modern ICS systems



DTM COMPONENTS: SHADOW KEYS TO THE ICS KINGDOM

DTM components key concepts

- It is generally no standalone tool
- ActiveX interfaces defined by the FDT-Spec.
- All rules of the device known
- All user dialogs contained
- Automatic generation of dependent parameters
- Reading and writing of parameters from/to the field device
- Diagnostic functions customized for the device
- No direct connection to any other device
- No information on the engineering environment
- Support for one or more device types

FDT/DTM simplified

PAS

Frame Application

DeviceDTM

CommDTM

COM Components

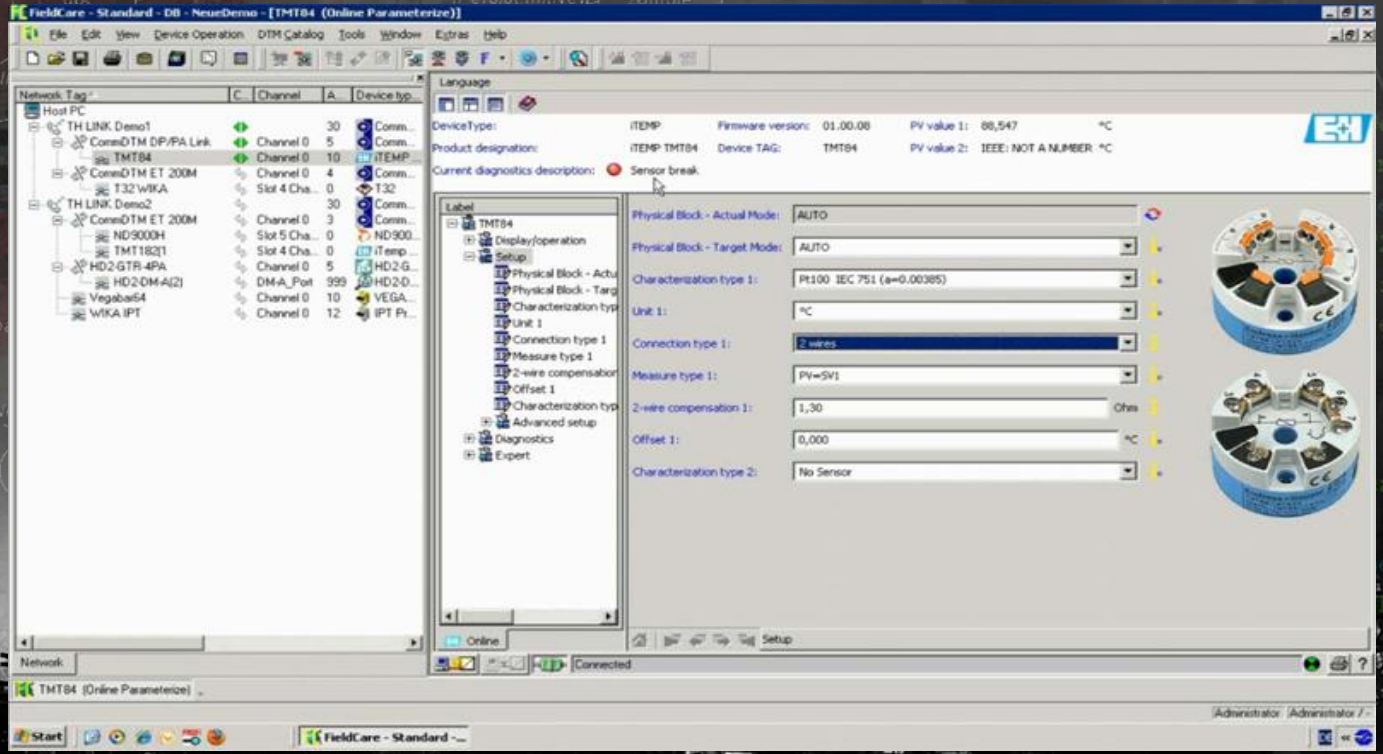
Modem/
Gateway

Transmitters && I/O

Industrial bus

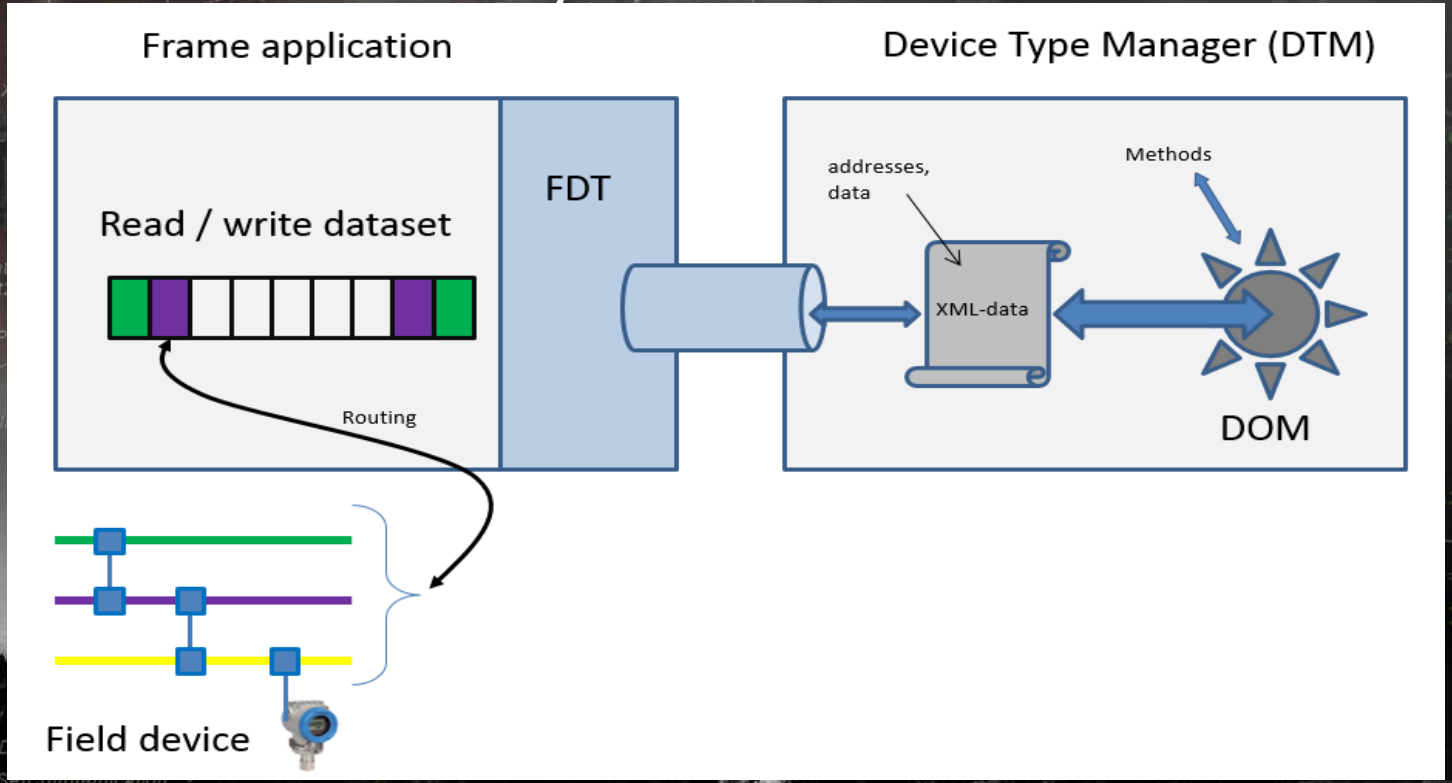
DTM COMPONENTS: SHADOW KEYS TO THE ICS KINGDOM

E&H FieldCare (PAS) – a typical frame application



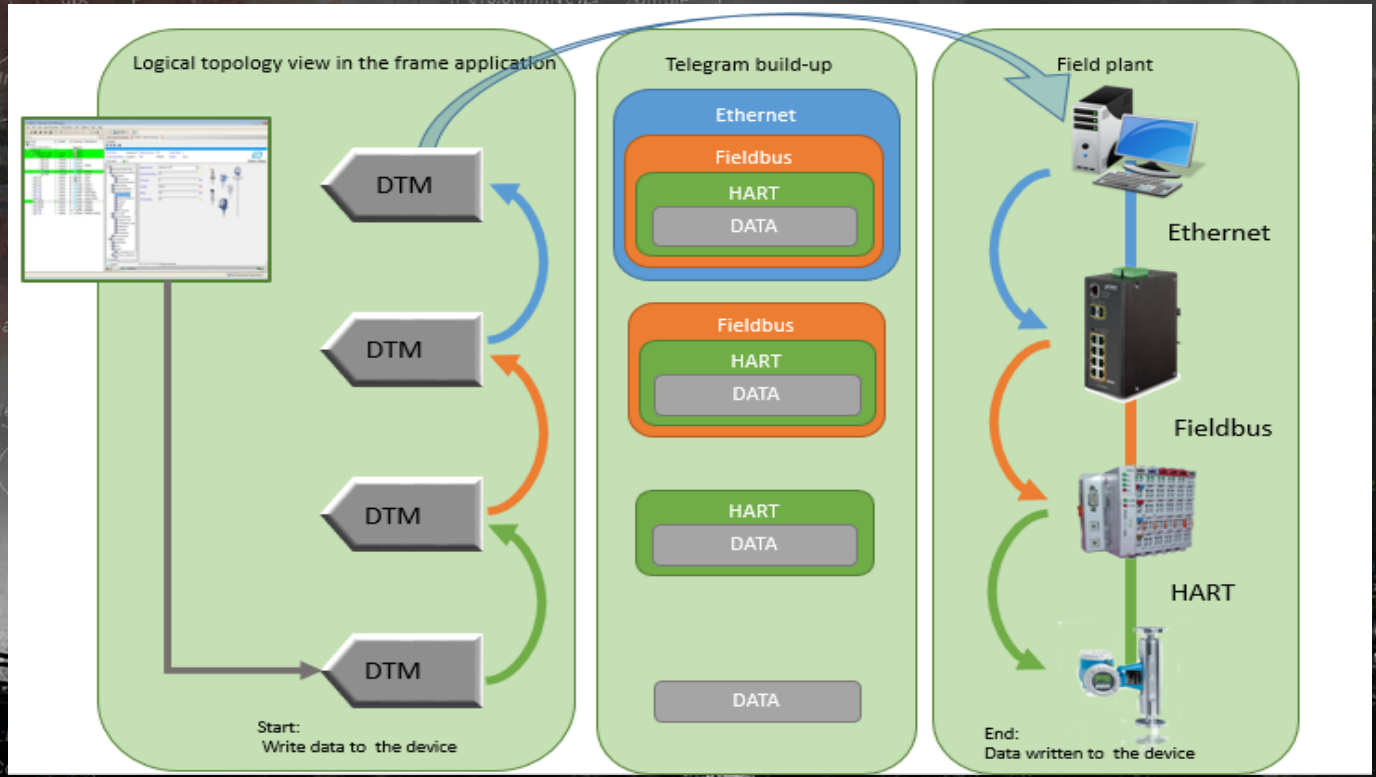
DTM COMPONENTS: SHADOW KEYS TO THE ICS KINGDOM

FDT/DTM: architecture internals



DTM COMPONENTS: SHADOW KEYS TO THE ICS KINGDOM

DTM multilayer concept



DTM COMPONENTS: SHADOW KEYS TO THE ICS KINGDOM

DTM implementations

- All of this sounds great, but in reality, DTM components are based on such technologies and use such “features” as:
  - OLE32
  - ActiveX
  - Visual Basic 6.0
  - .Net
  - COM
  - XML
  - STA
  - RPC

FDT/DTM architecture

Developers dream…

vs.

…cruel reality.

DTM COMPONENTS: SHADOW KEYS TO THE ICS KINGDOM

Research scope

Our research goals and scope

- In our research, we want to answer these questions:
  - Why is FDT/DTM architecture weak?
  - What kind of vulnerabilities in DTM components could cause a compromise of ICS infrastructure?
  - What about FDT 2.0 security?
- Also, we want to take some sample of all DTMs and find out how much of them have weaknesses and/or vulnerabilities
- Certified DTMs can be found in the catalog at http://www.fdtgroup.org/product-catalog/certified-dtms
- There are tons of DTMs
- We've decided to stick only to HART protocol and analyze ~100 DTMs

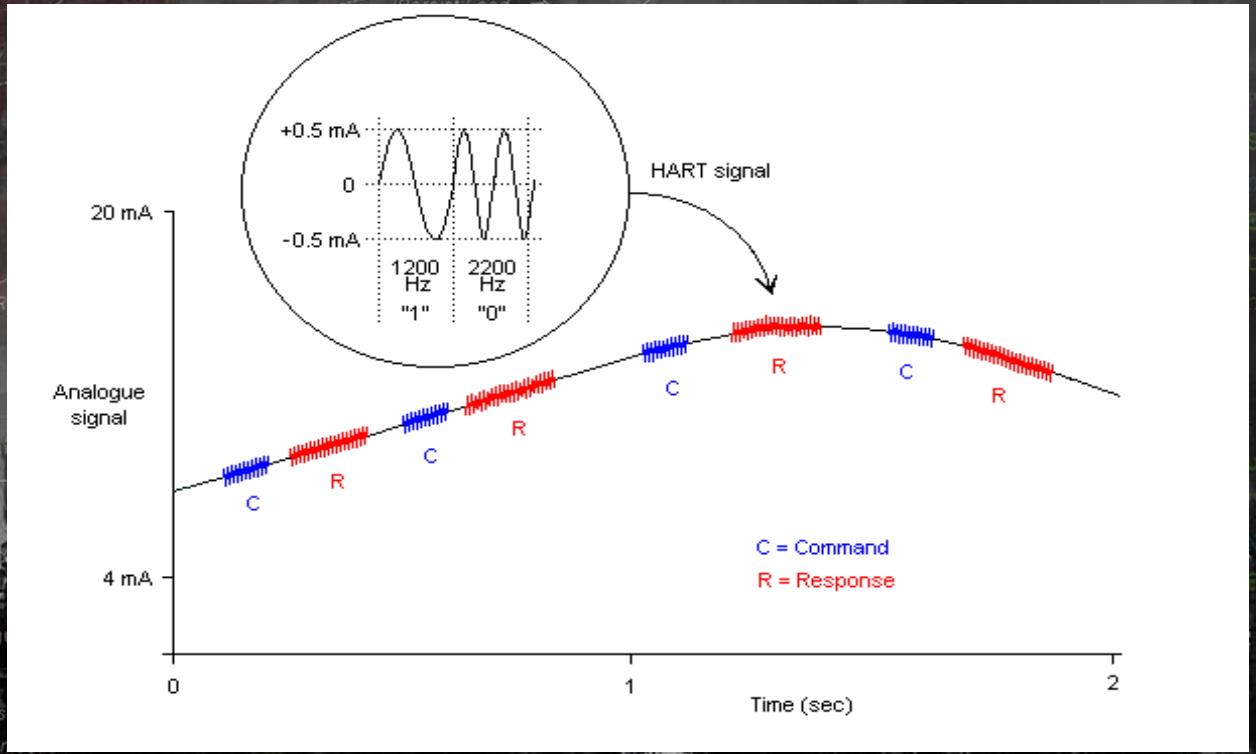
Why only DTMs for HART devices?

- We are familiar with this protocol
- We have hardware tools to work with and attack HART devices
- HART is used in critical industries, such as power plants, chemical factories, oil & gas, etc.

HART in two slides: first

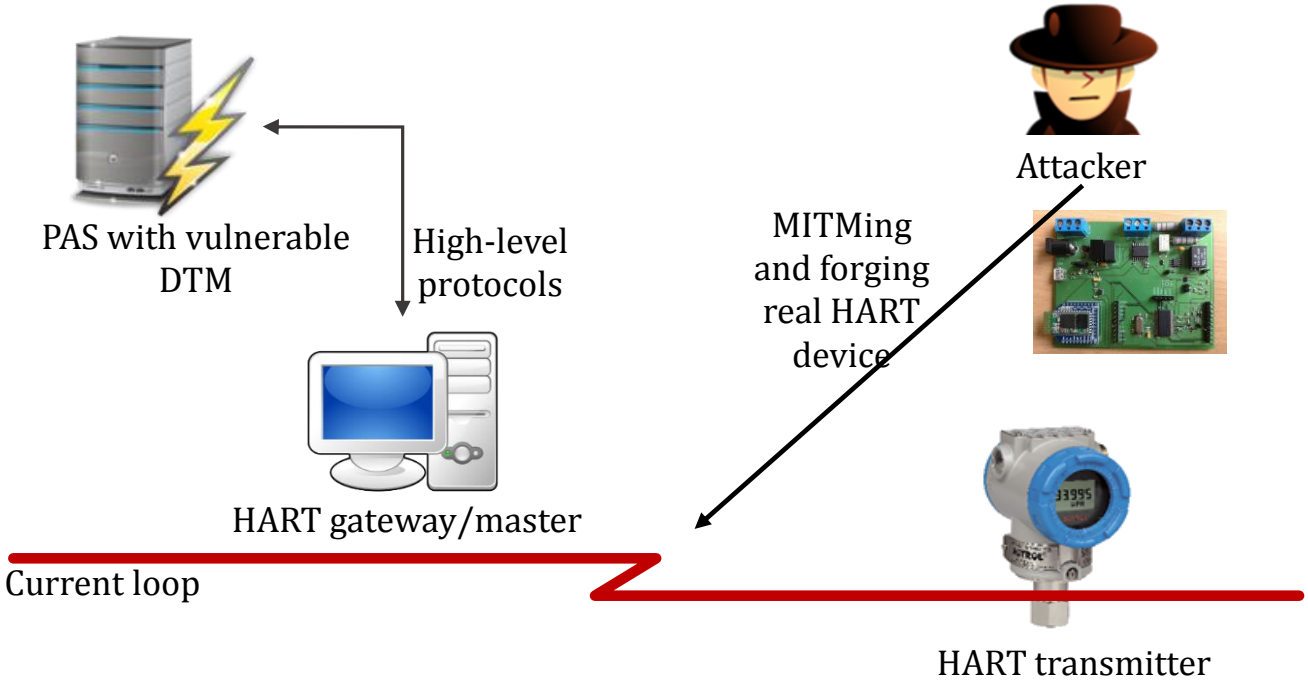
- Highway Addressable Remote Transducer Protocol
- Developed by Rosemount in mid-1980s
- Physical layer: FSK (copper wiring, 4-20 mA current loop)
- Current loop line length can reach 3 km => possible physical security problem
- Master-slave, half-duplex, 2200 Hz, 1200 bps
- No Authentication/Authorization/Cryptography (*wired)
- HART over IP version exists
- Max packet length – 255 B (standard), ~8 kB (reality).

HART in two slides: second



DTM COMPONENTS: SHADOW KEYS TO THE ICS KINGDOM

Attack model 1: through current loop



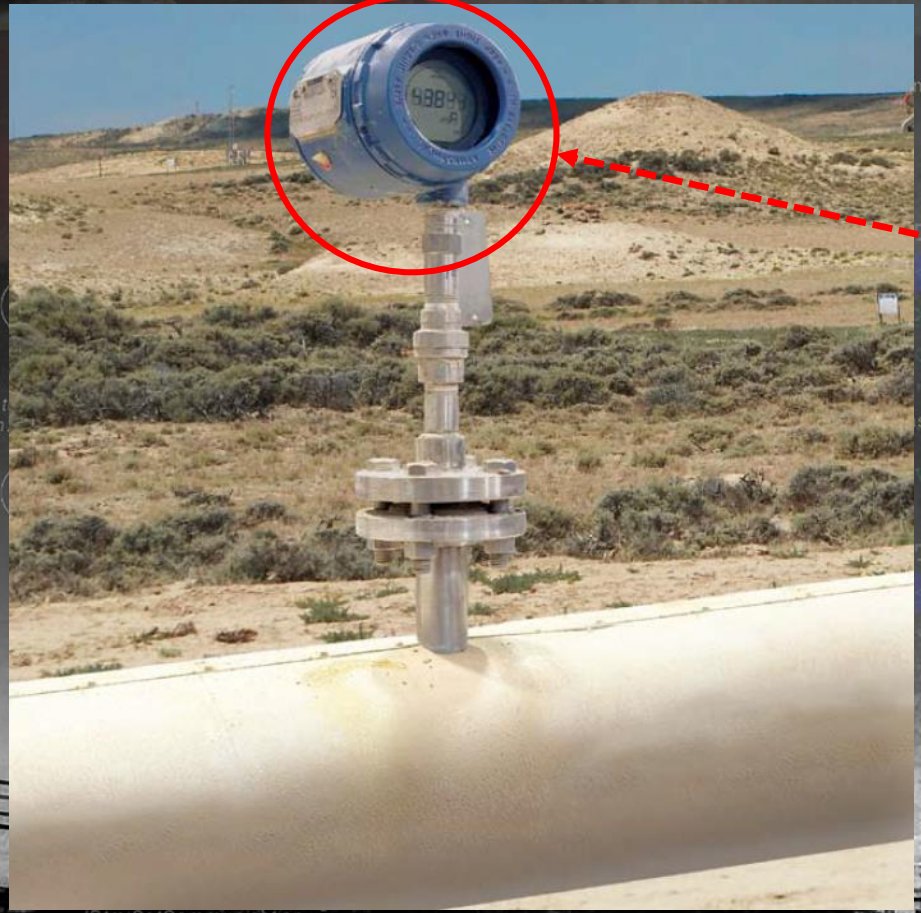
DTM COMPONENTS: SHADOW KEYS TO THE ICS KINGDOM

Real world

HART transmitter connected to current loop

DTM COMPONENTS: SHADOW KEYS TO THE ICS KINGDOM

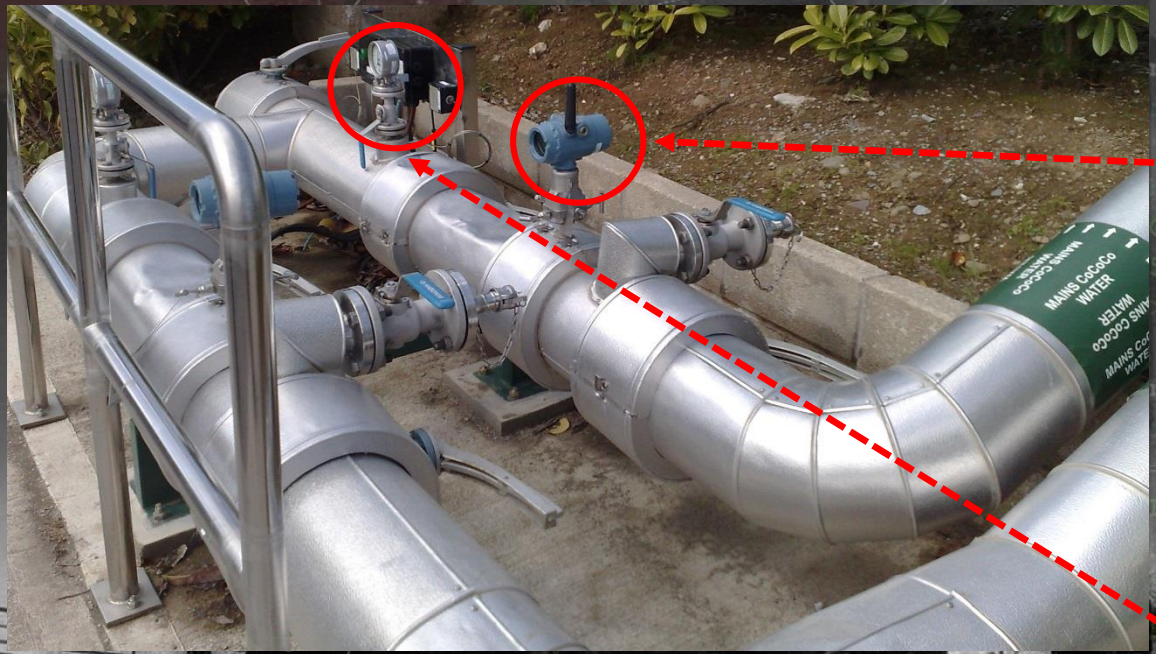
Real world



HART transmitter connected to current loop

DTM COMPONENTS: SHADOW KEYS TO THE ICS KINGDOM

Real world



Wireless HART transmitter

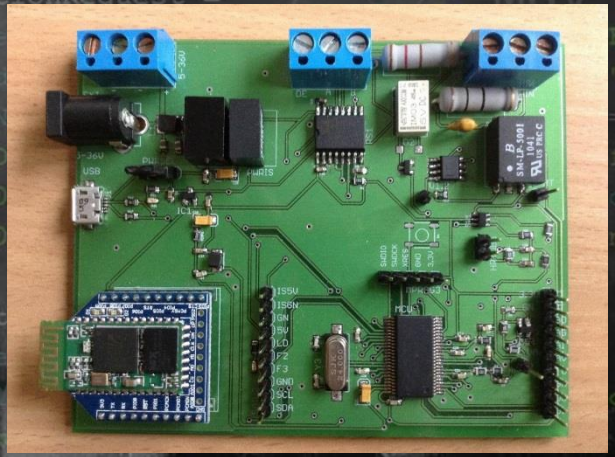
Wired HART transmitter

DTM COMPONENTS: SHADOW KEYS TO THE ICS KINGDOM

Tools and methods for MITMing HART CL

HRTShield for Arduino

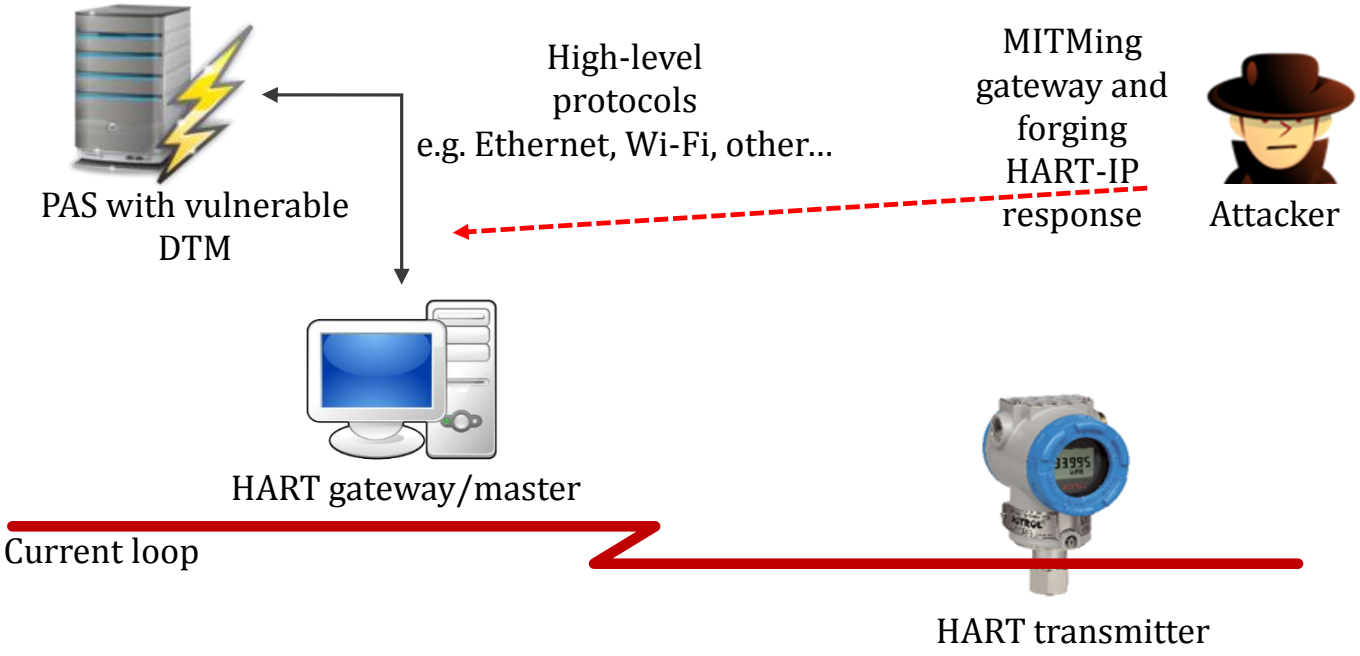
ICSCorsair



For more info on the topic, see: "HART as an attack vector: from current loop to application layer" (S4x14) and "ICSCorsair: how I will PWN your ERP from 4-20mA current loop" (BH USA'14).

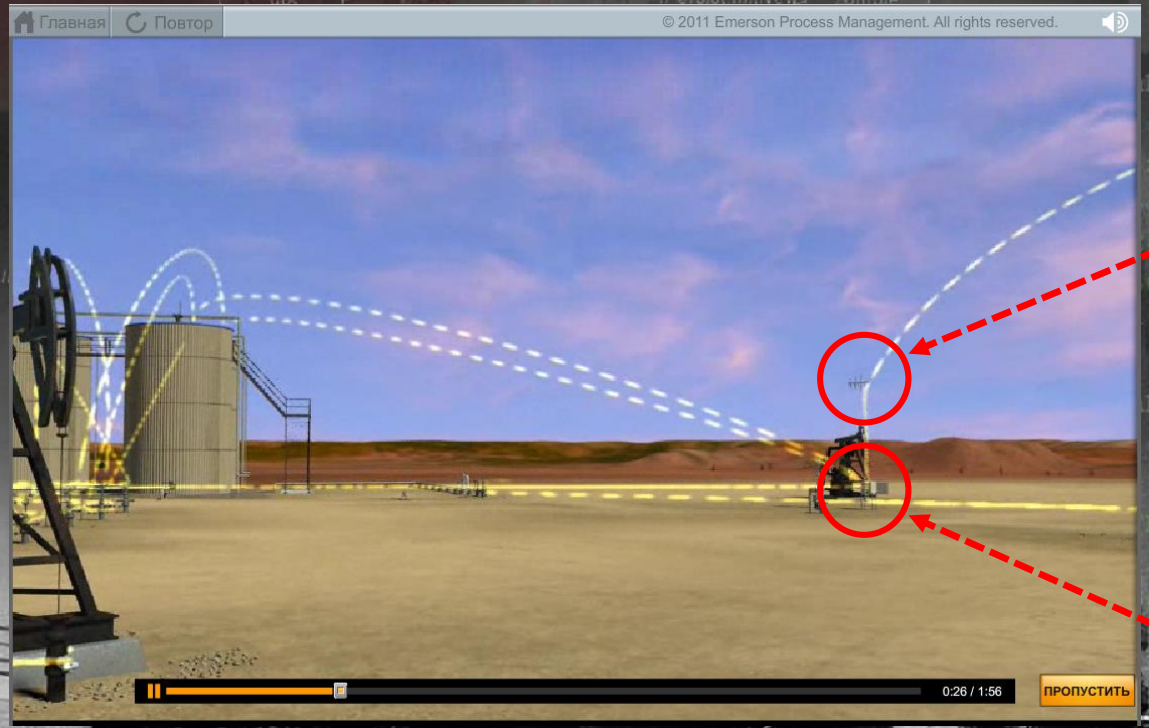
DTM COMPONENTS: SHADOW KEYS TO THE ICS KINGDOM

Attack model 2: through upper levels



DTM COMPONENTS: SHADOW KEYS TO THE ICS KINGDOM

Real world: Emerson marketing demo

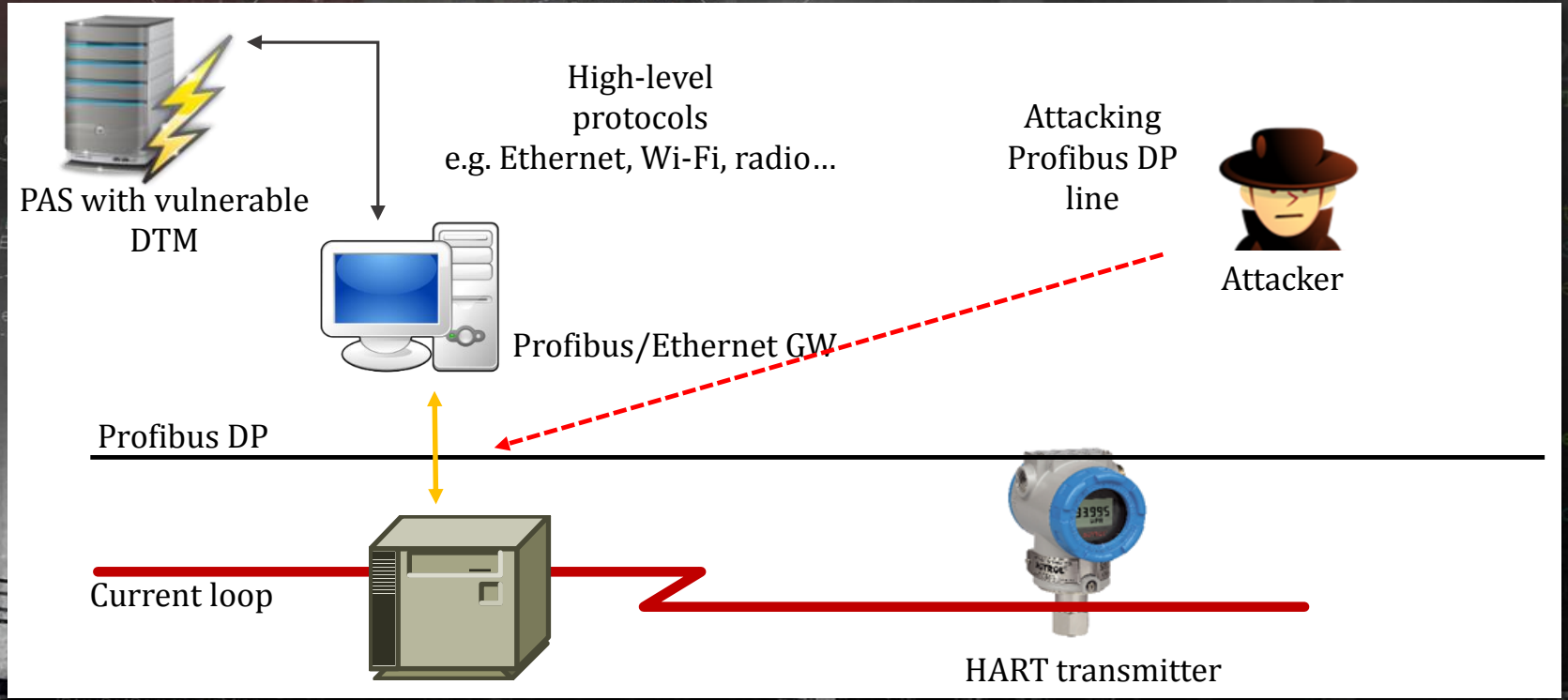


Broadband radiochannel to ICS DCS

Wireless HART transmitters, Wireless HART GWs to radiochannel

DTM COMPONENTS: SHADOW KEYS TO THE ICS KINGDOM

Attack model 3: through other low-lvl protocols



Research scope in one slide

114 DTMs from 24 Vendors for 752 Devices

DTM COMPONENTS: SHADOW KEYS TO THE ICS KINGDOM

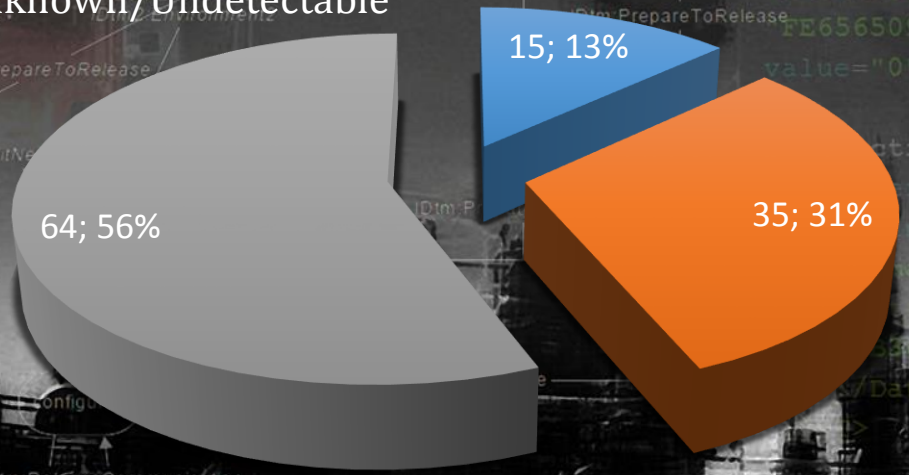
Some vendors

DTM COMPONENTS: SHADOW KEYS TO THE ICS KINGDOM

Two frameworks

DTMStudio/DTMLibrary/CoDIA

Other/Unknown/Undetectable



dtmManager/dtmGenerator

DTM COMPONENTS: SHADOW KEYS TO THE ICS KINGDOM


CoCreateInstance
IUnknown:Release (final)
IUnknown:Release (final)
up
zombie
IPersist:InitNew
IPersist:Load
IPersistXXX:InitNew
IPersistXXX:Load
new created
existing created
IDtm:Environment
IDtm2:Environment2 *
new enabled
IDtm:PrepareToR
IDtm:InitNew
IDtm:Config
IDtm:PrepareToDelete
configured
IDtm:ReleaseCommunication
IDtm:SetCommunication
Connect Condition / ConnectRequest()
communication set
TransactionRequest =
<?xml version="1.0"?><FDT xmlns="x-schema:FDTHARTCom
xmlns:fdt="x-schema:FDTDataTypesSchema.xml"> <DataEx
commandNumber="0" communicationReference=
4E21F2"> <fdt:Commur
Status deviceStatus="
changeResponse></FDT>
unicationSchema.xml"
">
<DataExchangeRequest commandNumber="0" communica
"1B36D3D1-C4DD-4BAF-9A31-016F704E21F2">
</DataExchangeRequest>
>

Fuzzing

How have we fuzzed?

DTM components may be written on different languages and use different runtimes, process model, e.t.c. Thus, we've used three different fuzzing methods:

1. Emulate CommDTM and put fuzzed protocol data directly into DeviceDTM (fastest)
2. Emulate device through virtual serial port.
3. Emulate device with hardware (HRTshield, ICSCorsair, e.t.c.). (slowest)

Tools that we've created for fuzzing

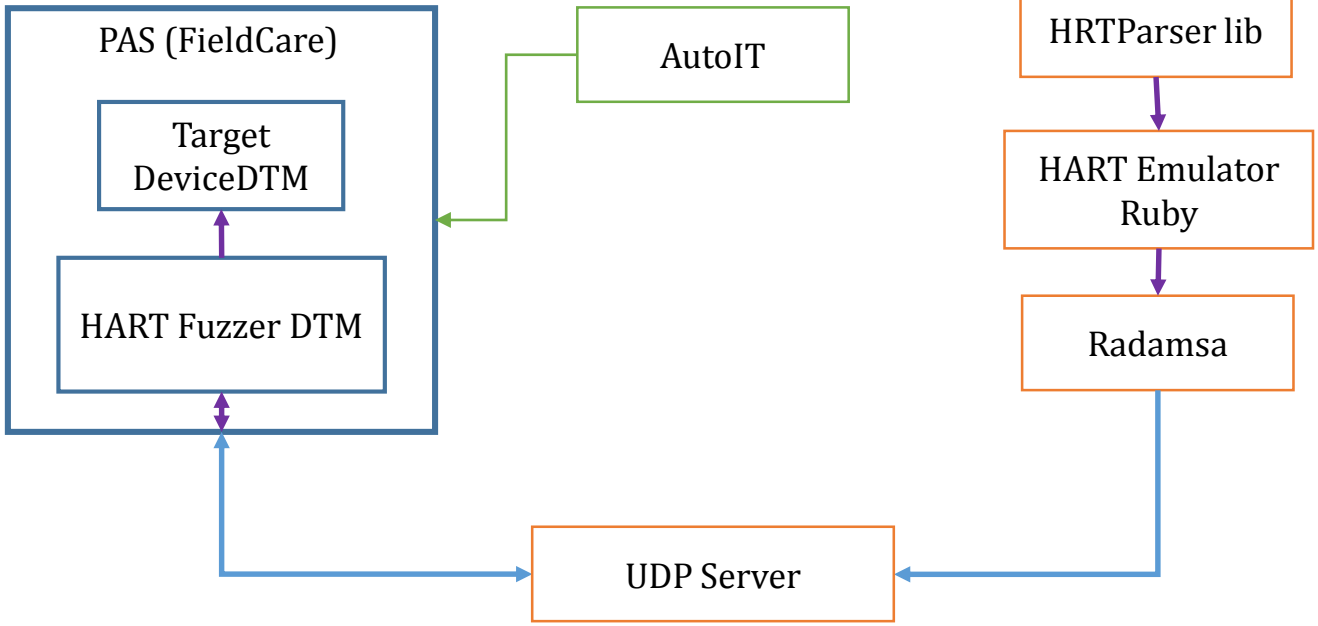
Software:

- HRTParser (HART packet creation/parsing library)
- Ruby HART emulator
- HART DTM Fuzzer (CommDTM)
- FuzzFrame (FDT Frame emulation)
- DTMSpy (logging DTM call stack/XML dataflow).

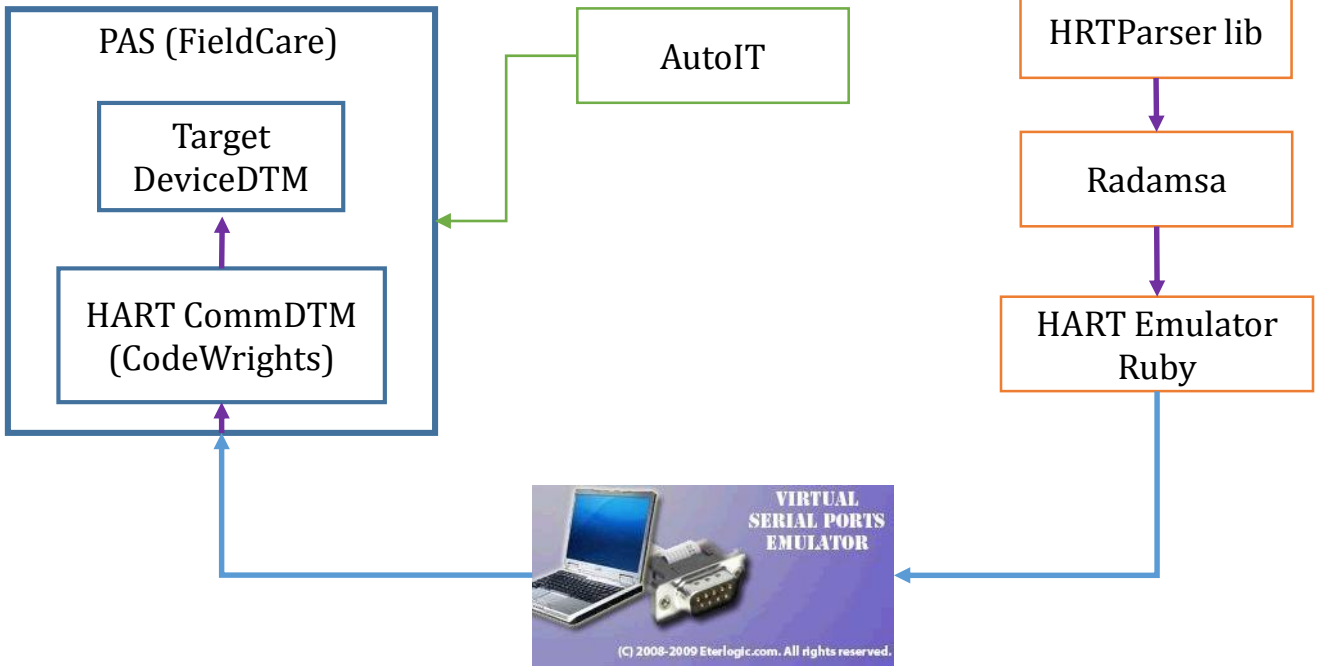
Hardware:

- ICSCorsair
- HRTShield

Fuzzing with special CommDTM component

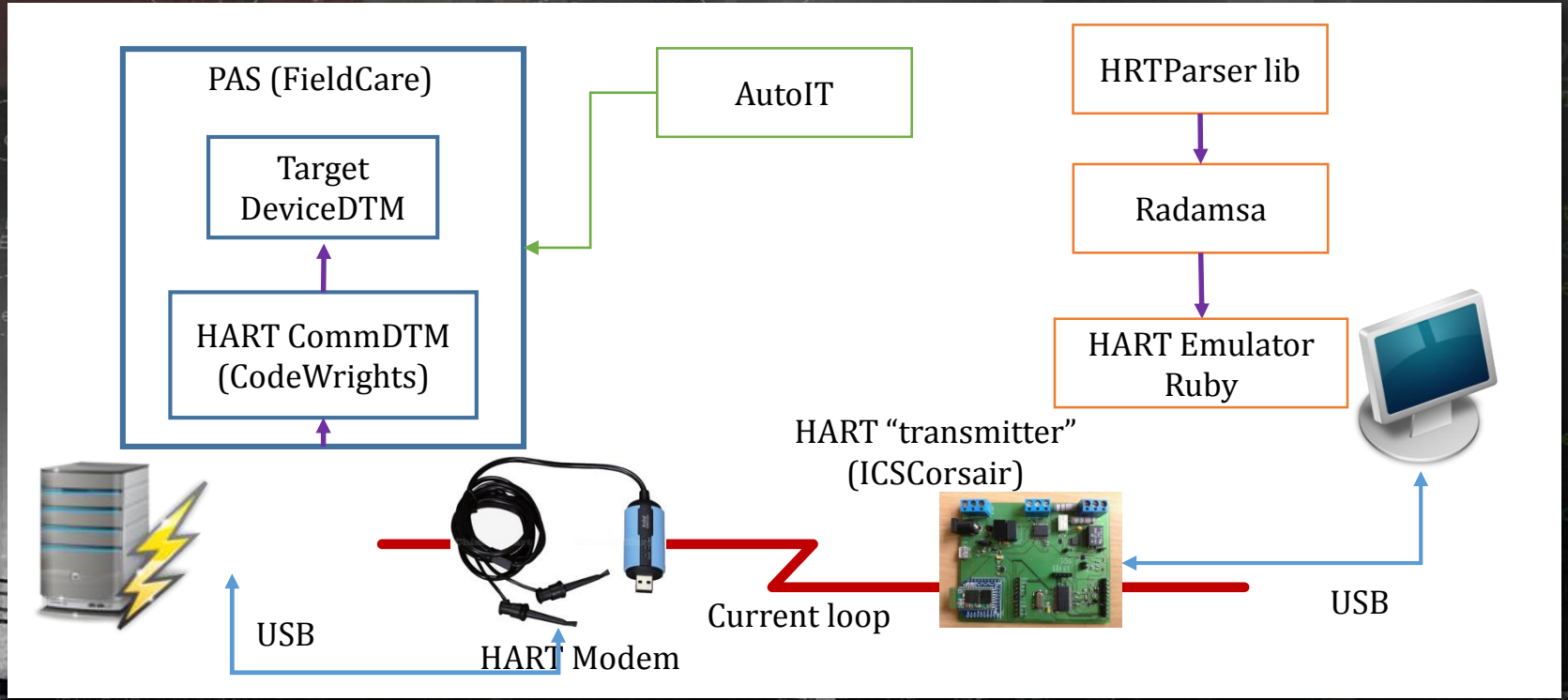


Fuzzing with Virtual Serial Ports



DTM COMPONENTS: SHADOW KEYS TO THE ICS KINGDOM

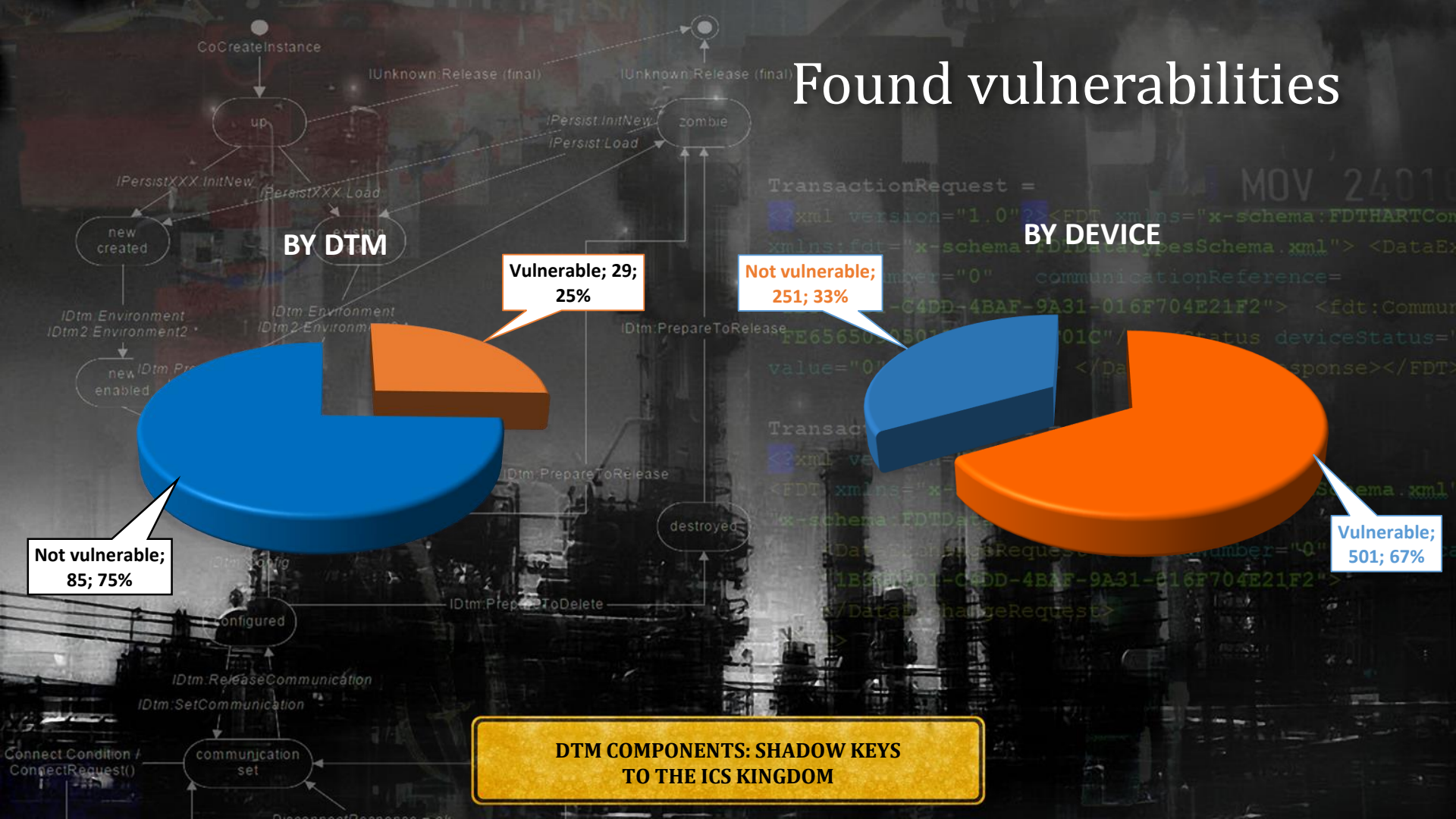
Fuzzing with hardware tools



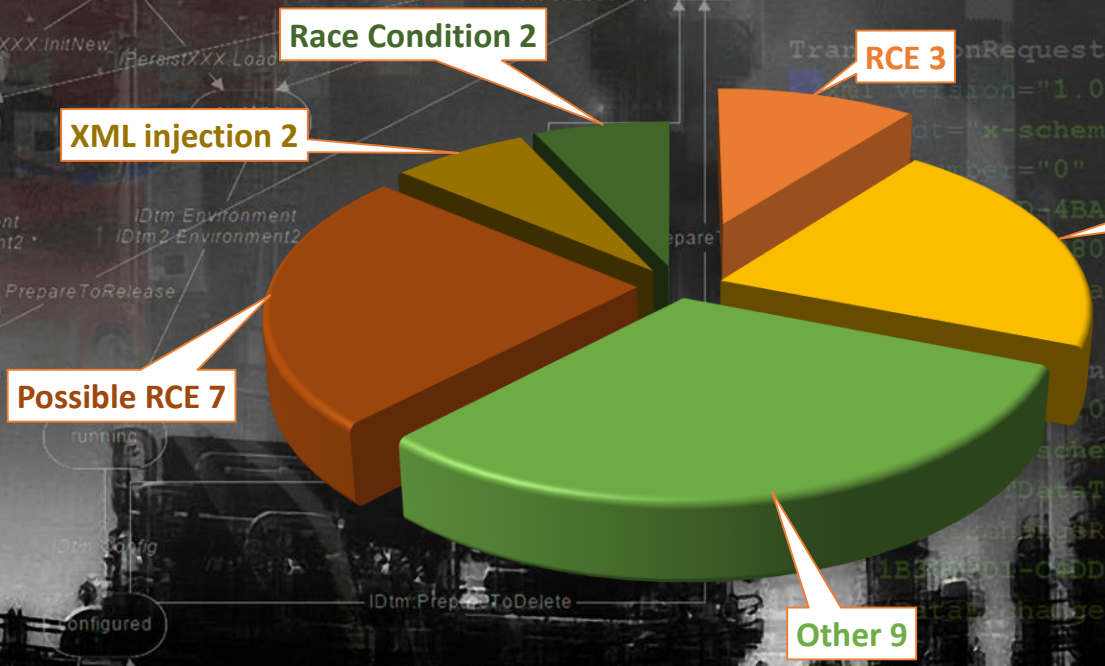
DTM COMPONENTS: SHADOW KEYS TO THE ICS KINGDOM

Results & statistics

Found vulnerabilities



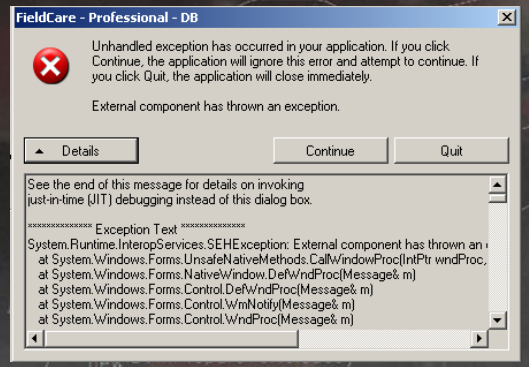
Types of found vulnerabilities (by DTM)

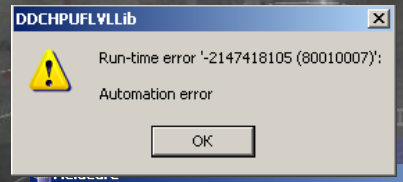


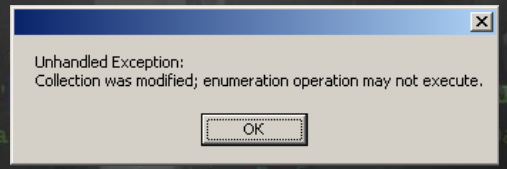
Total: 29 vulnerabilities

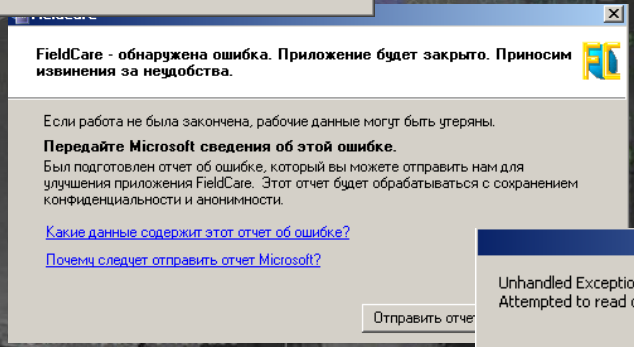
DTM COMPONENTS: SHADOW KEYS TO THE ICS KINGDOM

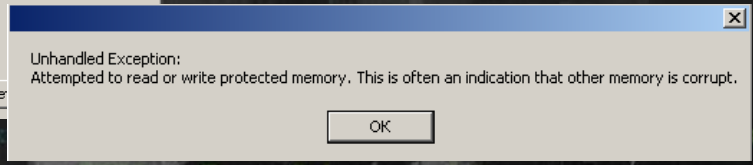
Tons of DoS and the like

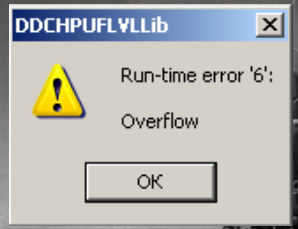


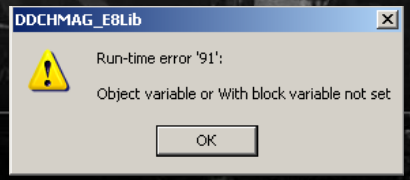


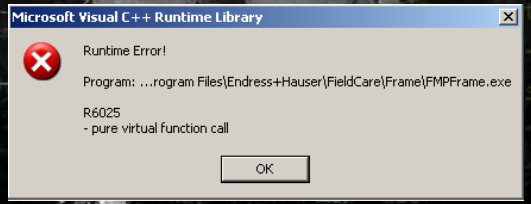


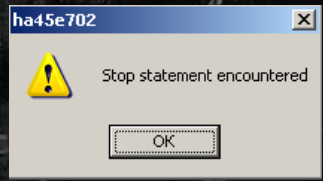












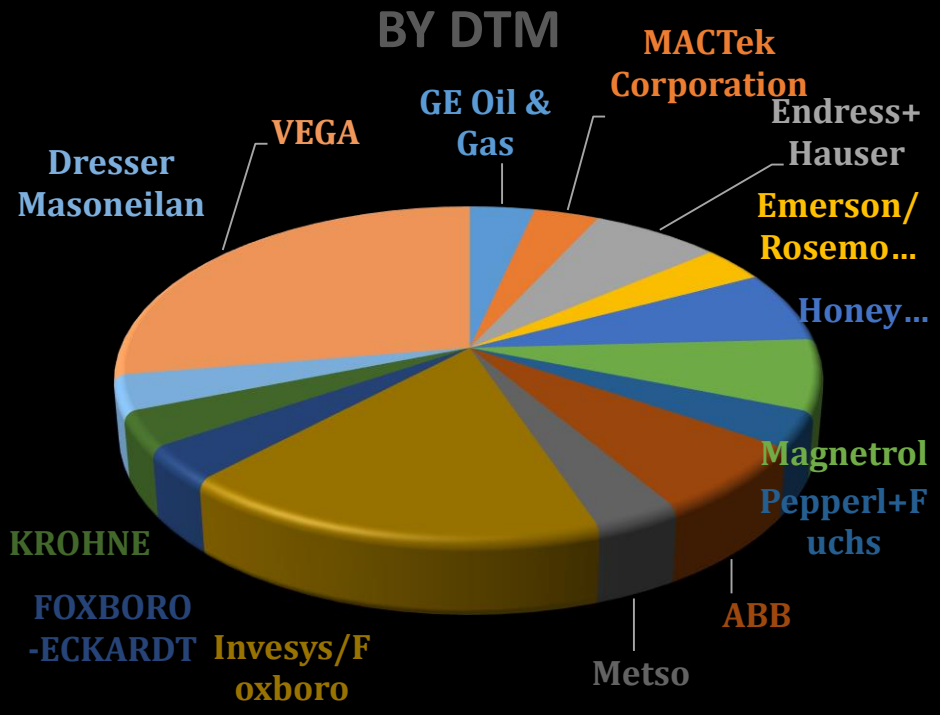
DTM COMPONENTS: SHADOW KEYS TO THE ICS KINGDOM

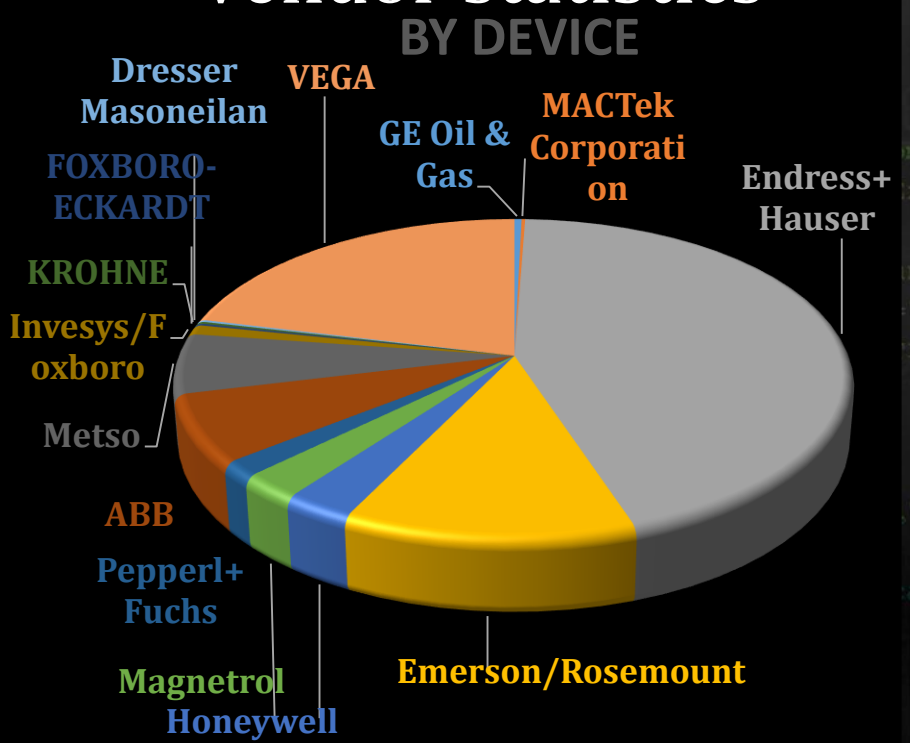
But…

SoZ, Responsible disclosure!

DTM COMPONENTS: SHADOW KEYS TO THE ICS KINGDOM

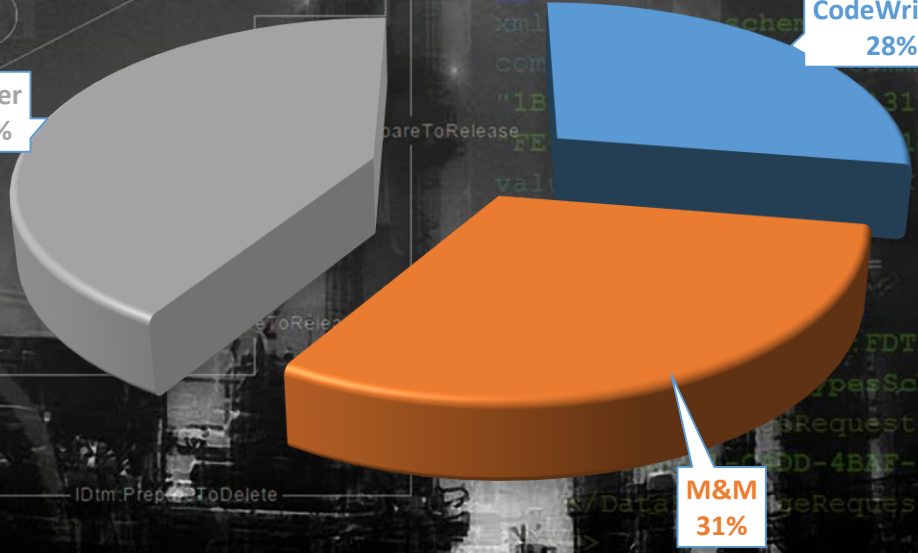
Vendor statistics





DTM COMPONENTS: SHADOW KEYS TO THE ICS KINGDOM

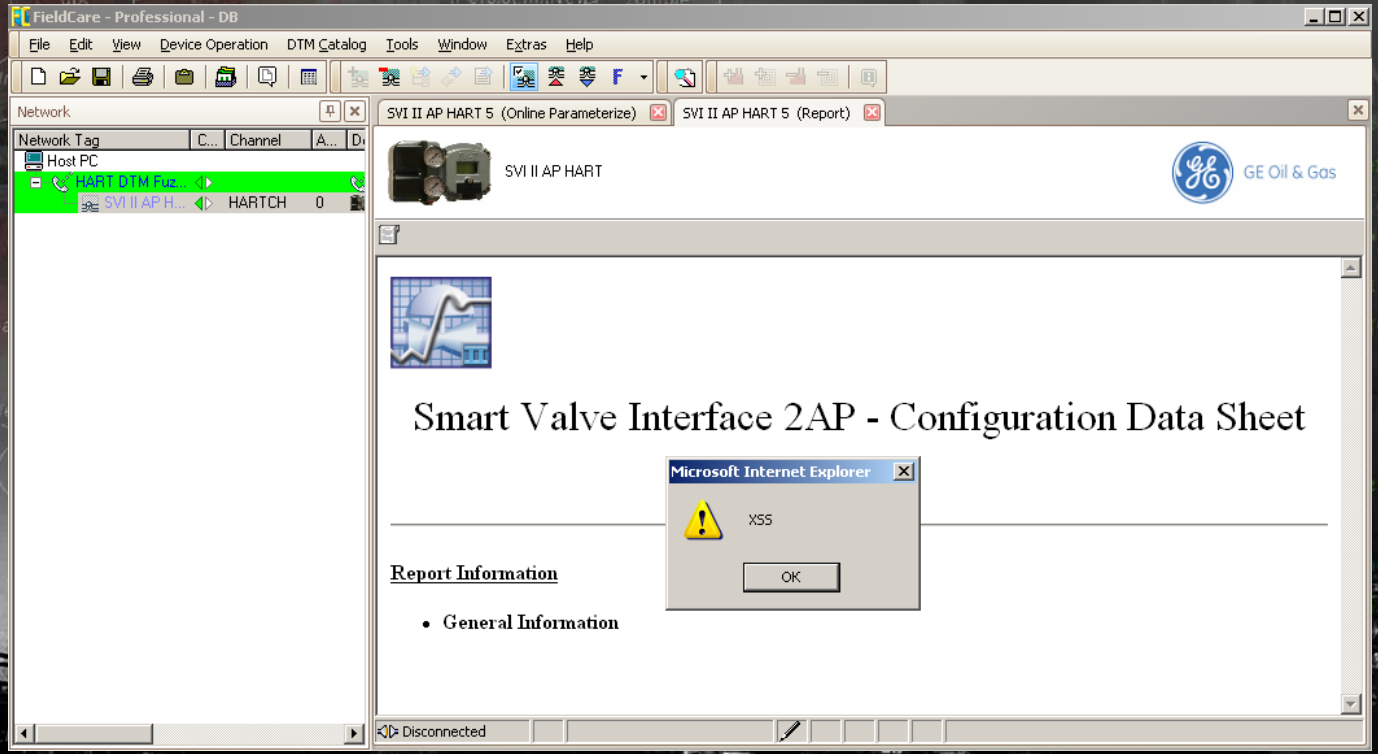
Framework statistics



DTM COMPONENTS: SHADOW KEYS TO THE ICS KINGDOM

Vulns & Funny things

XSS ☺

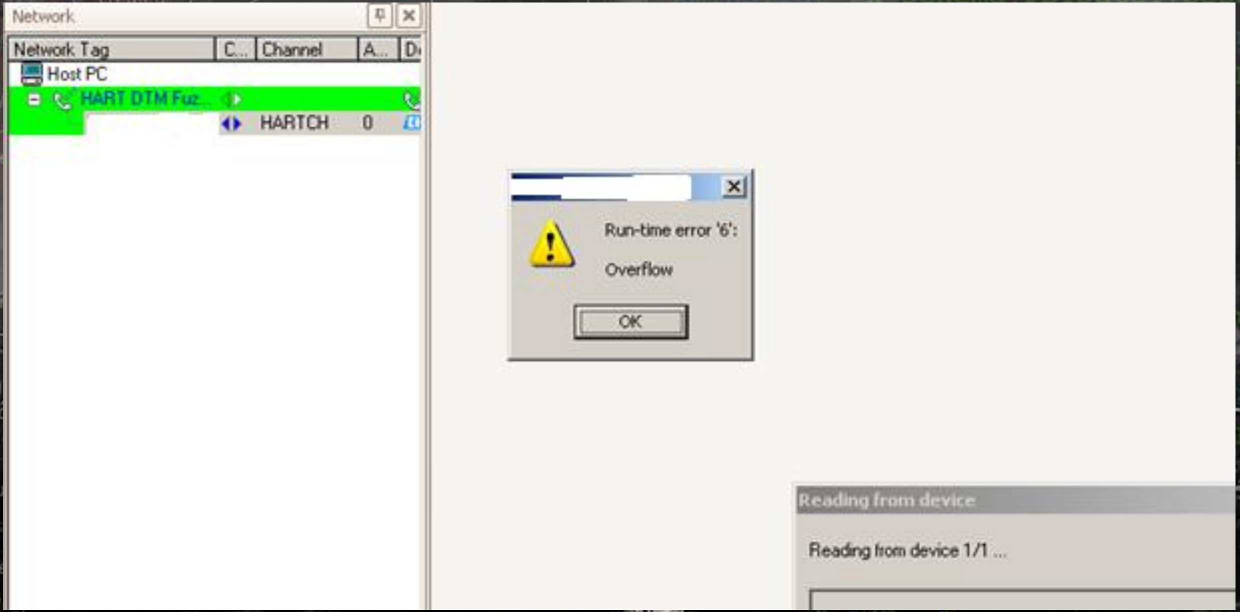


DTM COMPONENTS: SHADOW KEYS TO THE ICS KINGDOM

Too many data? E&H follow standards as always.

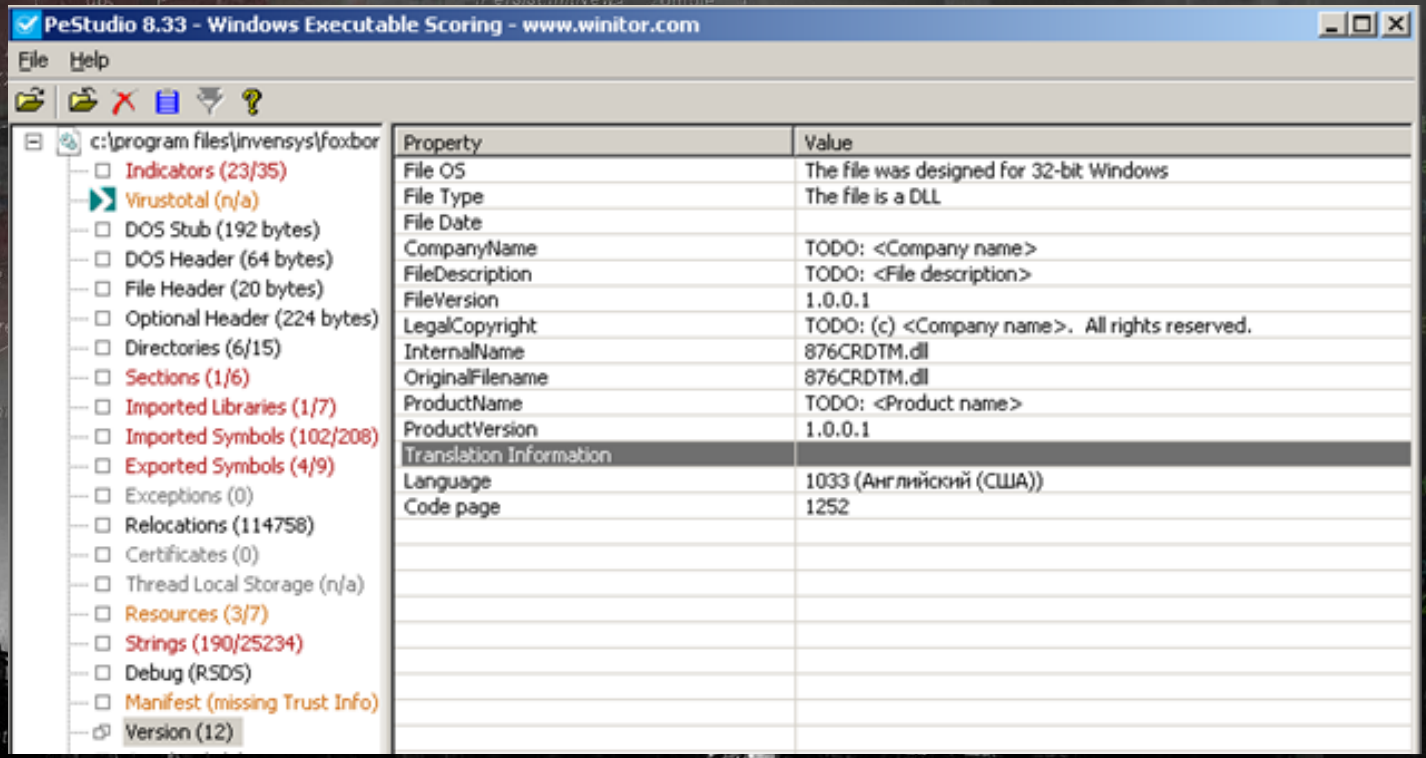
```
Sending to component:
Received: D? command_id =
Radamsing result...
Sending to component:          DADADADADADADADADADADADADADADADADADADADADADADADADADADADADADADADADADADADAD
ADADADADADADADADADADADADADADADADADADADADADADADAI          DADADADADADADADADADADADADADADADADADADADADAD
ADADADADADADADADADADADADADADADADADADADADADADADADADADADADADADADADADADADADADADADADAD
```

217 bytes < 255



DTM COMPONENTS: SHADOW KEYS TO THE ICS KINGDOM

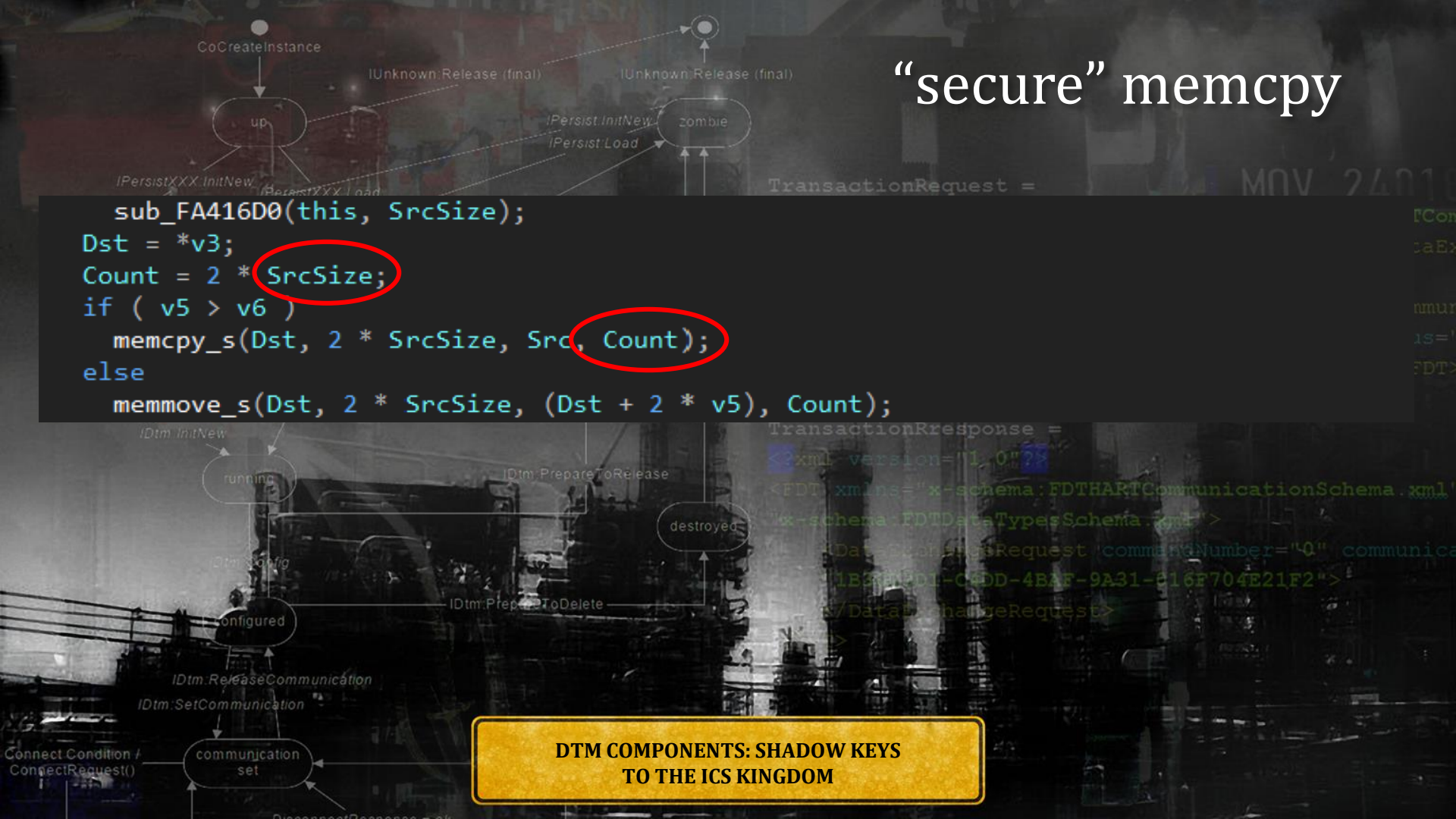
TODO: <Company name>



DTM COMPONENTS: SHADOW KEYS TO THE ICS KINGDOM

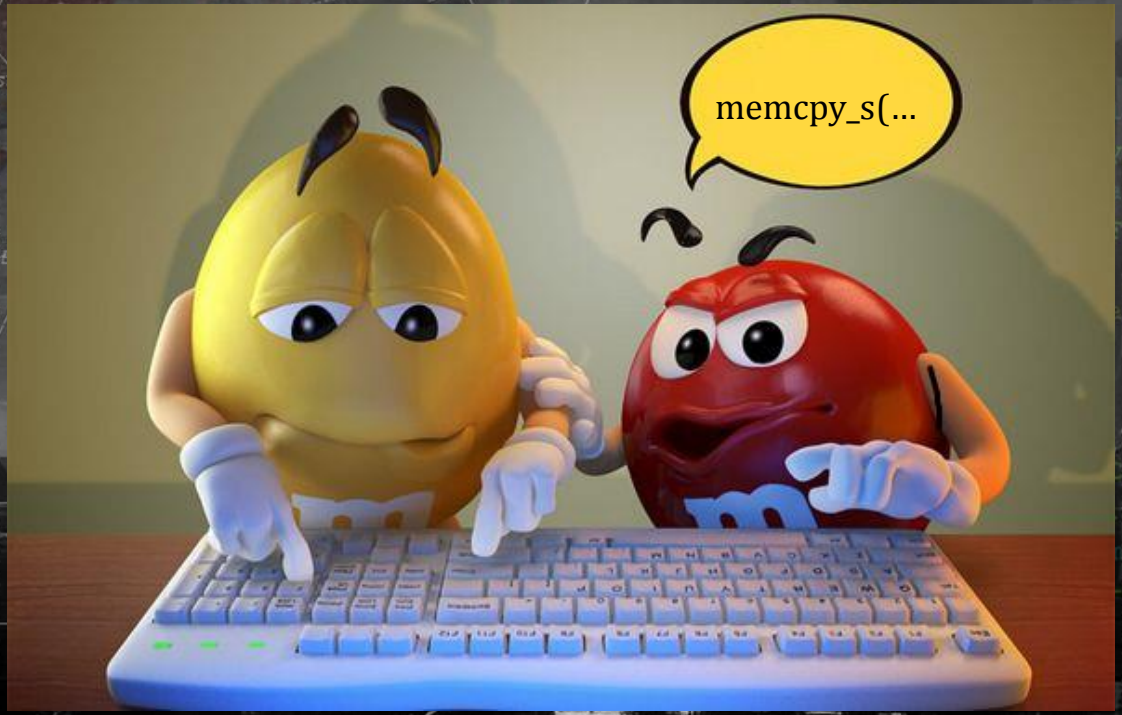
“secure” memcpy

```
  sub_FA416D0(this, SrcSize);
Dst = *v3;
Count = 2 * SrcSize;
if ( v5 > v6 )
  memcpy_s(Dst, 2 * SrcSize, Src, Count);
else
  memmove_s(Dst, 2 * SrcSize, (Dst + 2 * v5), Count);
```



DTM COMPONENTS: SHADOW KEYS TO THE ICS KINGDOM

M&M Software Gmbh.

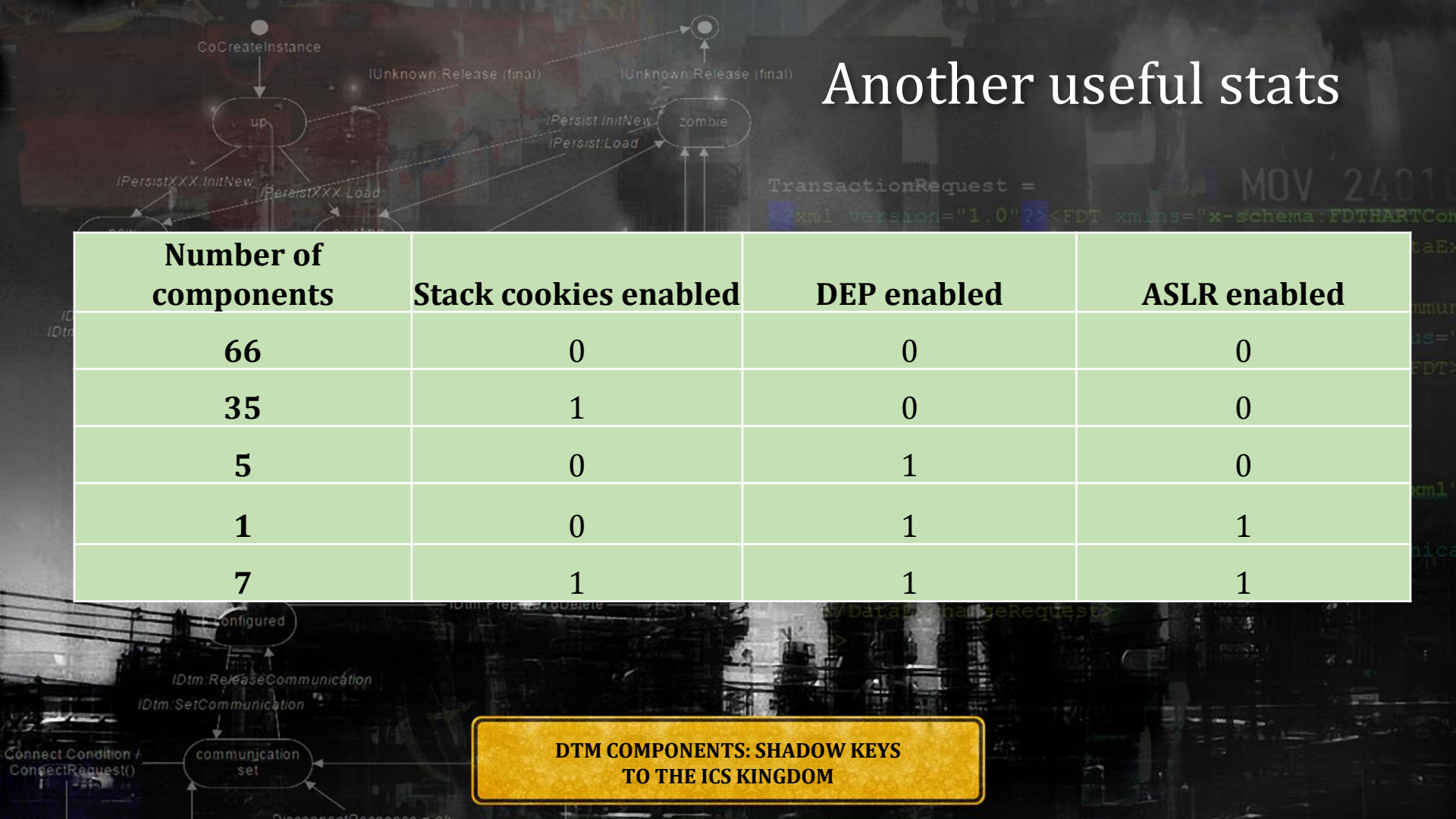


DTM COMPONENTS: SHADOW KEYS TO THE ICS KINGDOM

RCE DEMO VIDEO

Another useful stats

Number of components	Stack cookies enabled	DEP enabled	ASLR enabled
66	0	0	0
35	1	0	0
5	0	1	0
1	0	1	1
7	1	1	1



DTM COMPONENTS: SHADOW KEYS TO THE ICS KINGDOM

FDT 2.0 -- is it a solution?

FDT 2.0 new features

Recently, FDT group finally introduced a new version of FDT specification, v. 2.0. However, only a few devices support it. The key differences from 1.2.1 are:

- Interfaces are .Net-based
- Class architecture redesigned
- Increased performance
- No XML (interaction between FDT objects is based on .NET datatypes rather than XML)

Not a complete solution

FDT 2.0 problems:

- Low spread over the industry
- Backward compatibility ((de)serialization to XML for working with FDT 1.2.* could cause problems)
- Managed code will not be a complete solution if unmanaged code is still used (e.g. calling old C++ code from .Net)

Unfortunately, we could not found a real device supported by FDT 2.0 to test it; if you have one, we could borrow it for some time ;)

Conclusions

Conclusions

- During our research, we have found 29 vulnerabilities in 501 device from 14 vendors
- The quality of most DTM components is lower than medium
- FDT 2.0 could compensate some problems, but unfortunately it isn't actively used now
- Awaiting vendors' responses and hoping for the best!

Links

- ICSCorsair repository (hardware, firmware, software):

  http://github.com/Darkkey/ICSCorsair
- HRTShield repository:

  http://github.com/Darkkey/HRTShield
- HART parser repository:

  http://github.com/Darkkey/hartparser

Thanksgiving service

- **Svetlana Cherkasova && George Nosenko** for *some binary magic* and great help in reverse-engineering and creating proof-of-concept exploits
- **Andrey Abakumov** for help in finding XML injections
- **Fedor Savelyev aka Alouette** for some fuzzing ideas
- **Alexander Popov** for the great background picture

Q&A?
@dark_k3y, @cherboff