DTM COMPONENTS: SHADOW KEYS TO THE ICS KINGDOM

Alexander @dark_k3y Bolshev
Gleb @cherboff Cherbov
This document is a preliminary revision. For the latest version, please see http://github.com/Darkkey/DTMResearch
Alexander Bolshev (@dark_key)
IS auditor @ Ph.D.
Assistant Professor @ SPbETU
Distributed systems researcher
Yet another man wearing “some-color hat”

DTM COMPONENTS: SHADOW KEYS TO THE ICS KINGDOM
Gleb Cherbov (@cherboff)
IS researcher @

Information security researcher

DTM COMPONENTS: SHADOW KEYS
TO THE ICS KINGDOM
• Introduction to FDT/DTM
• Research scope
• Fuzzing technologies
• Vulnerabilities and weaknesses statistics
• Vulns && funny things
• FDT 2.0
• Conclusions
Intro to FDT/DTM
• 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.
Field protocols

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

DTM COMPONENTS: SHADOW KEYS TO THE ICS KINGDOM
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 implementations

Scope of FDT concept

Device configuration tool & DCS

- DDs
- Generic DTM
- Standalone tool
- FDT Interfaces
- Specific DTM “from scratch”

Different DTM implementations for simple devices to complex process control devices

DTM COMPONENTS: SHADOW KEYS TO THE ICS KINGDOM
FDT/DTM layers*

DTM COMPONENTS: SHADOW KEYS TO THE ICS KINGDOM

*Picture from http://www.automationworld.com/fdt-group-wants-your-input-yes-yours
Typical places of DTMs in modern ICS systems

**DTM COMPONENTS: SHADOW KEYS TO THE ICS KINGDOM**

- Corporate network
- ERP
- MES
- OPC
- AMS
- DCS
- PLC1
- PLC2,3...
- PLC7,8...
- Field devices
- Industrial bus
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

**DTM COMPONENTS: SHADOW KEYS TO THE ICS KINGDOM**

**Industrial bus**

**PAS**

**Frame Application**

**DeviceDTM**

**CommDTM**

**Modem/Gateway**

**Transmitters & I/O**

**Com Components**
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

Logical topology view in the frame application

Telegram build-up

Field plant

Start:
Write data to the device

End:
Data written to the device

DTM COMPONENTS: SHADOW KEYS TO THE ICS KINGDOM
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 DTM 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

PAS with vulnerable DTM

High-level protocols

HART gateway/master

Attacker

MITMing and forging real HART device

HART transmitter

Current loop
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

DTM COMPONENTS: SHADOW KEYS TO THE ICS KINGDOM
Tools and methods for MITMing HART CL

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).
Attack model 2: through upper levels

HART transmitter

Current loop

PAS with vulnerable DTM

HART gateway/master

High-level protocols
e.g. Ethernet, Wi-Fi, other...

Attacker

MITMing gateway and forging HART-IP response

DTM COMPONENTS: SHADOW KEYS TO THE ICS KINGDOM
Real world: Emerson marketing demo

DTM COMPONENTS: SHADOW KEYS TO THE ICS KINGDOM

Broadband radiochannel to ICS DCS

Wireless HART transmitters, Wireless HART GWs to radiochannel
Attack model 3: through other low-lvl protocols

DTM COMPONENTS: SHADOW KEYS

TO THE ICS KINGDOM

Profibus DP

PAS with vulnerable DTM

High-level protocols e.g. Ethernet, Wi-Fi, radio...

Attacker

Attacking Profibus DP line

Profibus/Ethernet GW

Current loop

HART transmitter

Profibus DP
Research scope in one slide

DTM COMPONENTS: SHADOW KEYS TO THE ICS KINGDOM

114 DTM from 24 Vendors for 752 Devices

XML code:
```
<XML version="1.0"?>
<x-scheme:FDTDataTypesSchema.xsd>
<x-schema:FDTDataTypesComponents.xsd>
<x-scheme:FDTDataTypesComponents.xsd>
</XML>
```
Some vendors

Endress+Hauser
People for Process Automation

Emerson

Honeywell

VEGA

Schneider Electric

ABB

DTM COMPONENTS: SHADOW KEYS TO THE ICS KINGDOM
Two frameworks

DTMStudio/DTMLibrary/CoDIA

dtmManager/dtmGenerator

DTM COMPONENTS: SHADOW KEYS TO THE ICS KINGDOM
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

DTM COMPONENTS: SHADOW KEYS TO THE ICS KINGDOM
Fuzzing with special CommDTM component

DTM COMPONENTS: SHADOW KEYS TO THE ICS KINGDOM
Fuzzing with Virtual Serial Ports

DTM COMPONENTS: SHADOW KEYS TO THE ICS KINGDOM

PAS (FieldCare)

Target DeviceDTM

HART CommDTM (CodeWrights)

AutoIT

HRTParser lib

Radamsa

HART Emulator Ruby

VIRTUAL SERIAL PORTS EMULATOR

(C) 2000-2009 Etherlogix.com, All rights reserved.
Fuzzing with hardware tools

DTM COMPONENTS: SHADOW KEYS TO THE ICS KINGDOM

PAS (FieldCare)

Target Device DTM

HART Comm DTM (CodeWrights)

USB

HART Modem

Current loop

HART “transmitter” (ICSCorsair)

AutoIT

HRTParser lib

Radamsa

HART Emulator Ruby

USB
Results & statistics
Found vulnerabilities

DTM COMPONENTS: SHADOW KEYS TO THE ICS KINGDOM

BY DTM

- Vulnerable; 29; 25%
- Not vulnerable; 85; 75%

BY DEVICE

- Vulnerable; 501; 67%
- Not vulnerable; 251; 33%

Vulnerable
29
25%
Not vulnerable
85
75%
Types of found vulnerabilities (by DTM)

- Race Condition 2
- XML injection 2
- Possible RCE 7
- RCE 3
- DoS 6
- Other 9

Total: 29 vulnerabilities
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
DTM COMPONENTS: SHADOW KEYS TO THE ICS KINGDOM

Vendor statistics

BY DTM

- Dresser Masoneilan
- VEGA
- GE Oil & Gas
- Endress+Hauser
- Emerson/Rosemount
- Honeywell
- Magnetrol
- Pepperl+Fuchs
- ABB
- Metso
- Invesys/Foxboro
- KROHNE
- FOXBORO-ECKARDT

BY DEVICE

- VEGA
- GE Oil & Gas
- MACTek Corporation
- Emerson/Rosemount
- KROHNE
- Invesys/Foxboro
- Metso
- ABB
- Pepperl+Fuchs
- Magnetrol
- Honeywell
- Emerson/Rosemount
- Endress+Hauser
Framework statistics

DTM COMPONENTS: SHADOW KEYS TO THE ICS KINGDOM

- CodeWrights: 28%
- M&M: 31%
- Other: 41%
DTM COMPONENTS: SHADOW KEYS TO THE ICS KINGDOM
Too many data? E&H follow standards as always.

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);
```
M&M Software Gmbh.

DTM COMPONENTS: SHADOW KEYS TO THE ICS KINGDOM

memcpy_s(...
RCE DEMO VIDEO
Another useful stats

<table>
<thead>
<tr>
<th>Number of components</th>
<th>Stack cookies enabled</th>
<th>DEP enabled</th>
<th>ASLR enabled</th>
</tr>
</thead>
<tbody>
<tr>
<td>66</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>35</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

DTM COMPONENTS: SHADOW KEYS TO THE ICS KINGDOM
FDT 2.0 -- is it a solution?
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)
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

• 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!
• 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
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