

To: DEFCON 26

Date: Aug-12 2018



Check Point®
SOFTWARE TECHNOLOGIES LTD.

Subject:

what The Fax?!



Check Point Research

Received OK?

WhoAreWe?



Yaniv Balmas

“This should theoretically work”

Security Researcher



Check Point Software Technologies

@ynvb

Eyal Itkin

“That’s cool.”

Security Researcher

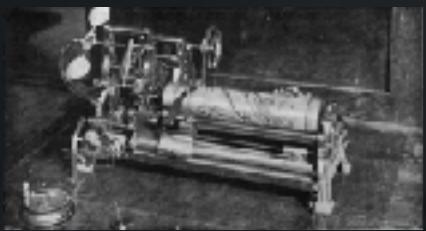


Check Point Software Technologies

@eyalitkin

FAX History

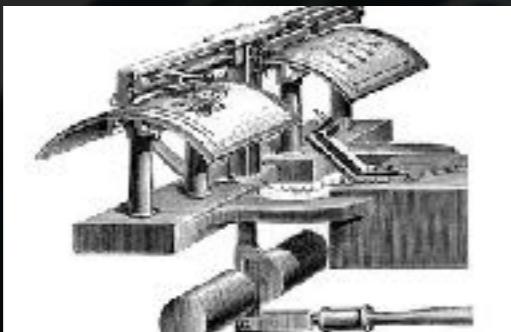
Caselli
Invents
Machine
Similar to
Today's FAX



1846



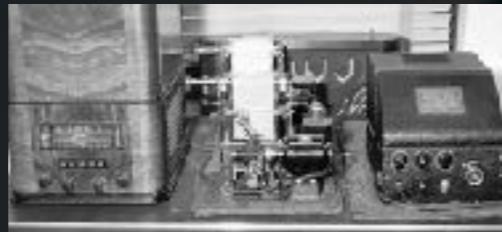
Alexander
Bain Sends
An Image
Over a Wire



1860



XEROX
Introduces
the First
Commercial
FAX Machine



1923



Enter the
RadioFAX.
Used by
Navies

1966



XEROX
Introduces
the First
Commercial
FAX Machine



1980



Group III
ITU-T Fax
Standards
T.30, T.4, T.6



GammaFAX
Brings
Computers
Into FAX
Network

BackToTheFuture

Quality

Accessibility

Reliability

Authenticity



✗

✗

✗

✗

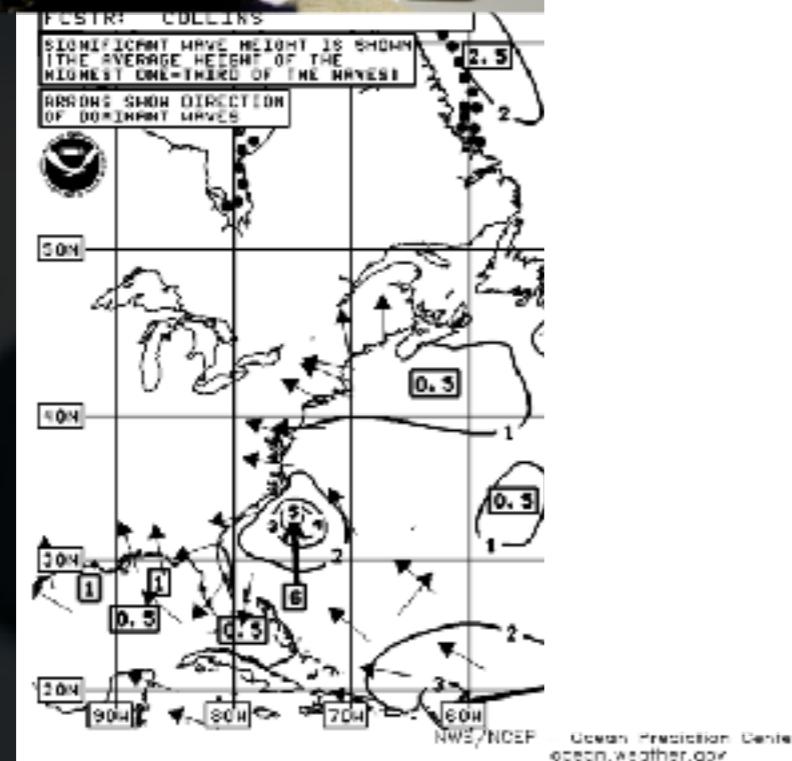
✓

✓

✓

?

BackToTheFuture



Google

"contact us" "fax"



Page 9 of about 386,000,000 results (0.33 seconds)

Contact Us, Fax, Phone, Email, Social Media Accounts

<https://fiea.ucf.edu/home/contact-us/> ▾

Jun 27, 2018 - All contact us information for FIEA, the Florida Interactive Entertainment Academy. Fax, email, social media accounts.

contact us - Financial Ombudsman

www.financial-ombudsman.org.uk/contact/index.html ▾

Jump to [switchboard and fax](#) - 020 7964 1000 (switchboard); +44 20 7964 1000 (for calls from outside the UK); 020 7964 1001 (main fax)

Contact Us | Ace Hardware on The FAX

www.aceonthefax.com/contact-us ▾

Contact Ace Hardware on The FAX by sending us an email, or give us a call at (720) 484-8585. We love to hear your feedback!

Contact Us | Lund University

<https://www.lunduniversity.lu.se/about/contact-us> ▾

Jun 20, 2018 - Contact Us. University building in spring with pink ... Tel: +46 (0)46 222 0000 Fax: +46 (0)46 222 4720. Postal address: Lund University

BackToTheFuture



Microsoft

Office

Windows

Surface

Microsoft Limited

Microsoft Campus

Thames Valley Park

Reading

Berkshire

RG6 1WG

UNITED KINGDOM

Phone: (+44) 0344 800 2400

Fax: (+44) 0870 60 20 100

BackToTheFuture



Office

Windows

Surface

Microsoft Limited

Microsoft Campus



HEAD OFFICE

1 FUXINGMEN NEI DAJIE,
BEIJING,
CHINA

SWIFT: BKCHCNBJ

TEL:(86) 010-66596688

FAX:(86) 010-66016871

POST CODE: 100818

WEBSITE:

www.boc.cn,www.bankofchina.com

BEIJING BRANCH

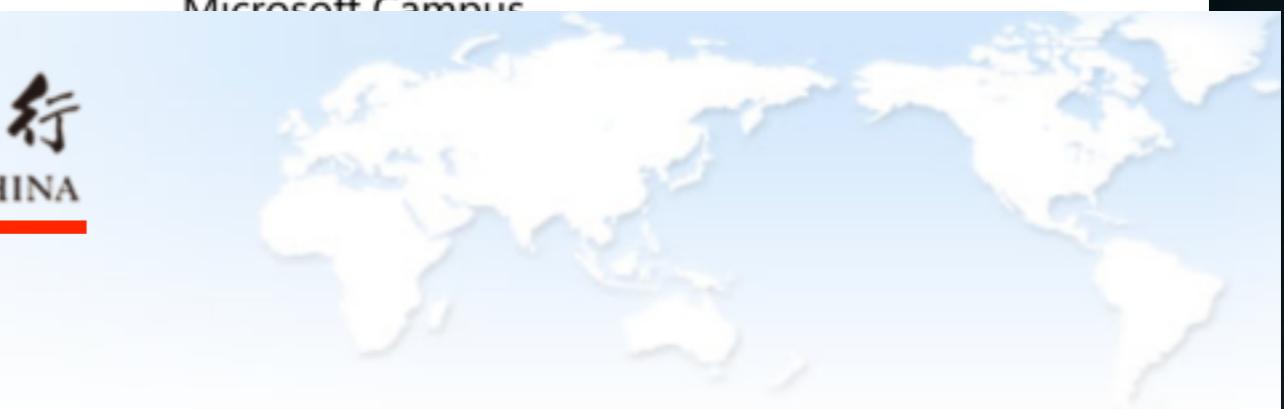
A,C,E KAIHENG CENTER,
2 CHAOYANGMEN NEI DAJIE,
DONGCHENG DISTRICT,
BEIJING,
CHINA

SWIFT: BKCHCNBJ110

TEL:(86) 010-85122288

FAX:(86) 010-85121739

POST CODE: 100010



BackToTheFuture



Office

Windows

Surface



Limited

May 18, 2012

President Barack Obama
The White House
1600 Pennsylvania Avenue, NW
Washington, DC 20500

Via facsimile: +1-202-456-2461



ING BRANCH
: KAIHENG CENTER,
AOYANGMEN NEI DAJIE,
DONGCHENG DISTRICT,
BEIJING,
CHINA
SWIFT: BKCHCNBJ110
TEL:(86) 010-85122288
FAX:(86) 010-85121739
POST CODE: 100010

CHINA
SWIFT: BKCHCNBJ
TEL:(86) 010-66596688
FAX:(86) 010-66016871
POST CODE: 100818
WEBSITE:
www.boc.cn,www.bankofchina.com

BackToTheFuture

Forms for Reporting to FDA

[SHARE](#)[TWEET](#)[LINKEDIN](#)[PIN IT](#)[EMAIL](#)[PRINT](#)

* IMPORTANT * You may continue to use Form FDA 3500 (voluntary), Form FDA 3500B (consumer-friendly), and FDA 3500A (mandatory) past the listed

For use by healthcare professionals, consumers, and patients. Submit the completed form using built-in postage-paid mailer, or fax.

Instructions for Completing Form FDA 3500

Via facsimile:

Reader, or just print the blank form and fill it out by hand. The Voluntary Form FDA 3500 features a postage-paid pre-addressed mailer.

TER,
EI DAJIE,
ICT,

- Form FDA 3500 - Voluntary Reporting**

For use by healthcare professionals, consumers, and patients. Submit the completed form using built-in postage-paid mailer, or fax.

[Instructions for Completing Form FDA 3500](#)

10
288
739
)

WTF?!

COPY

FaxToday

- Modern FAX is no longer a simple “FAX Machines”
- The same old FAX technology is now wrapped inside newer technologies
- ALL-IN-ONE printers are EVERYWHERE

Send and Receive Faxes by Email Today!

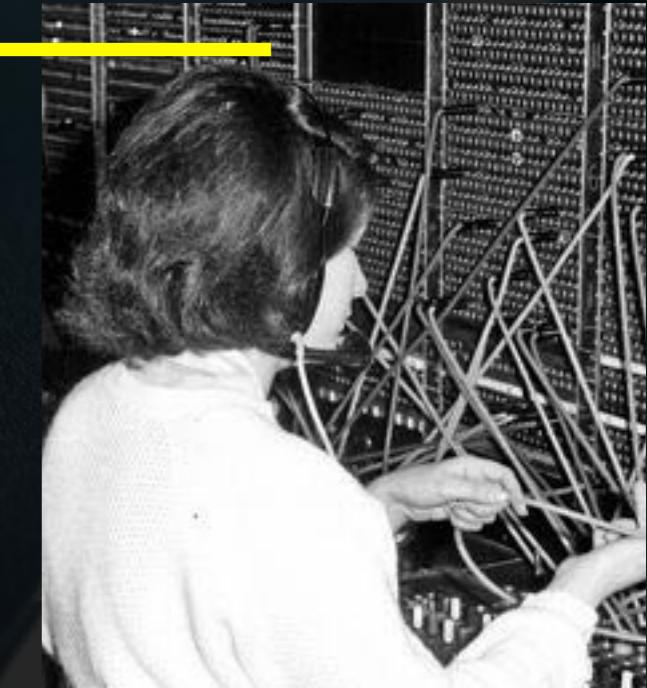
- No more fax machines, paper or toner
- All you need is an email account
- Send faxes instantly - no more waiting

Sign up now and start faxing immediately!



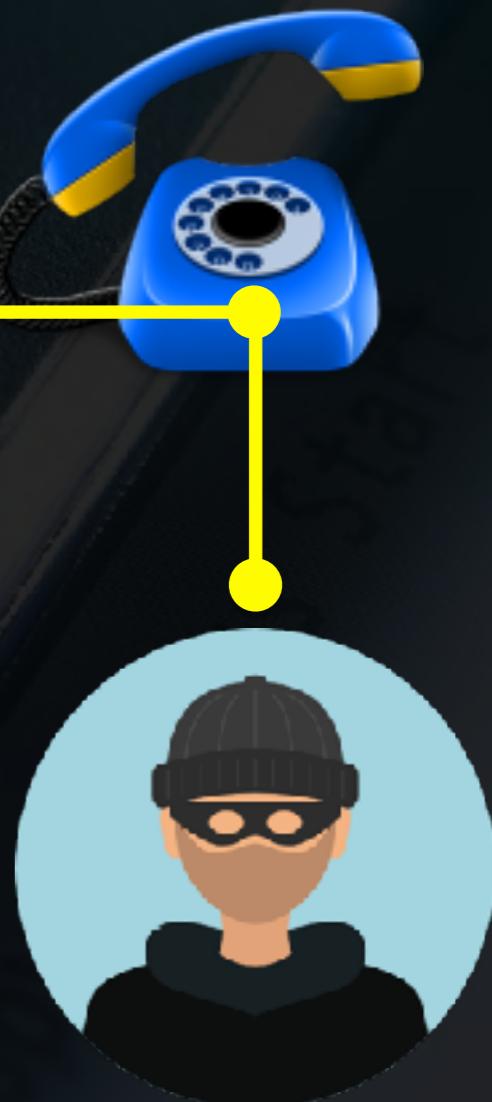
The Security View

ALL-IN-ONE Printers



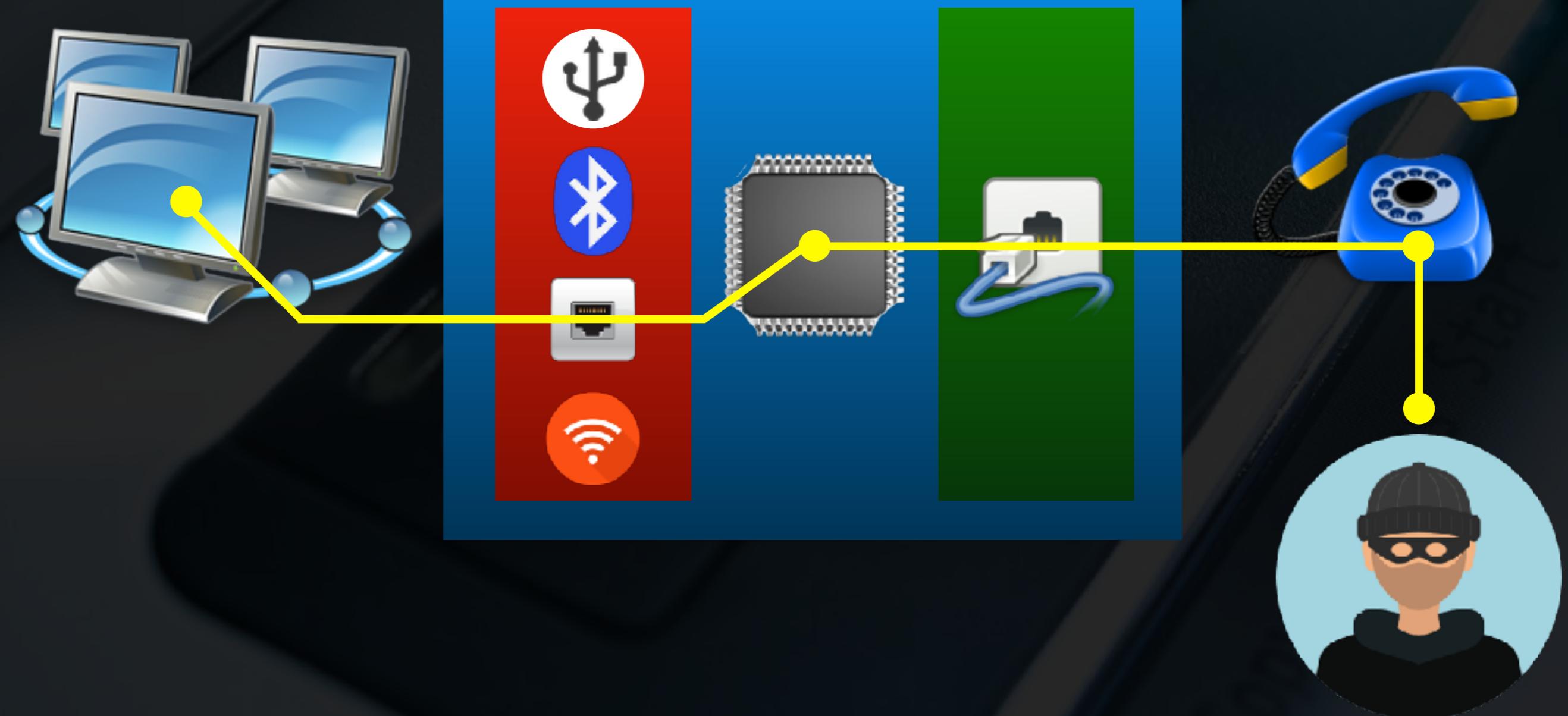
FAX Attack

ALL-IN-ONE Printers



FAX Attack

ALL-IN-ONE Printers



**Challenge
Accepted**

What is the Target?

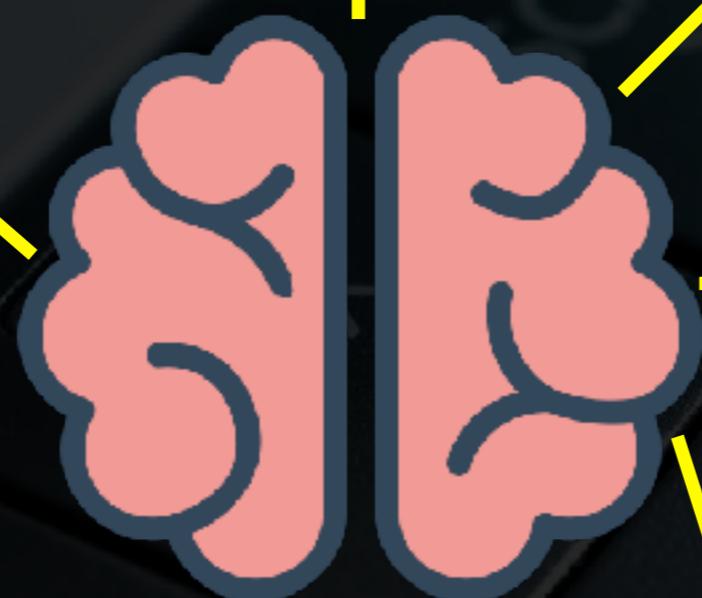
How can we Debug it?

How to Obtain the Code?

What is The OS?

How Does FAX Even Work?

Where to look for vulns?

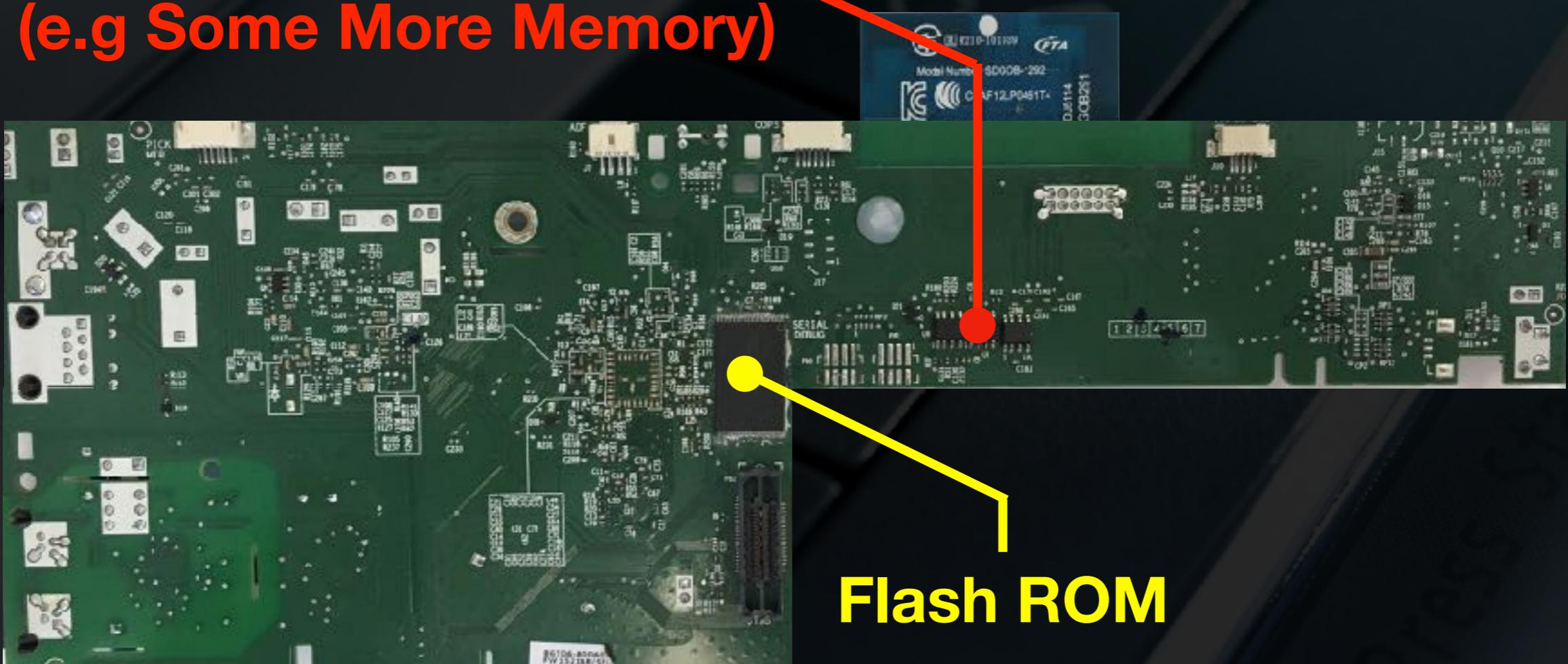


And The Winners Is



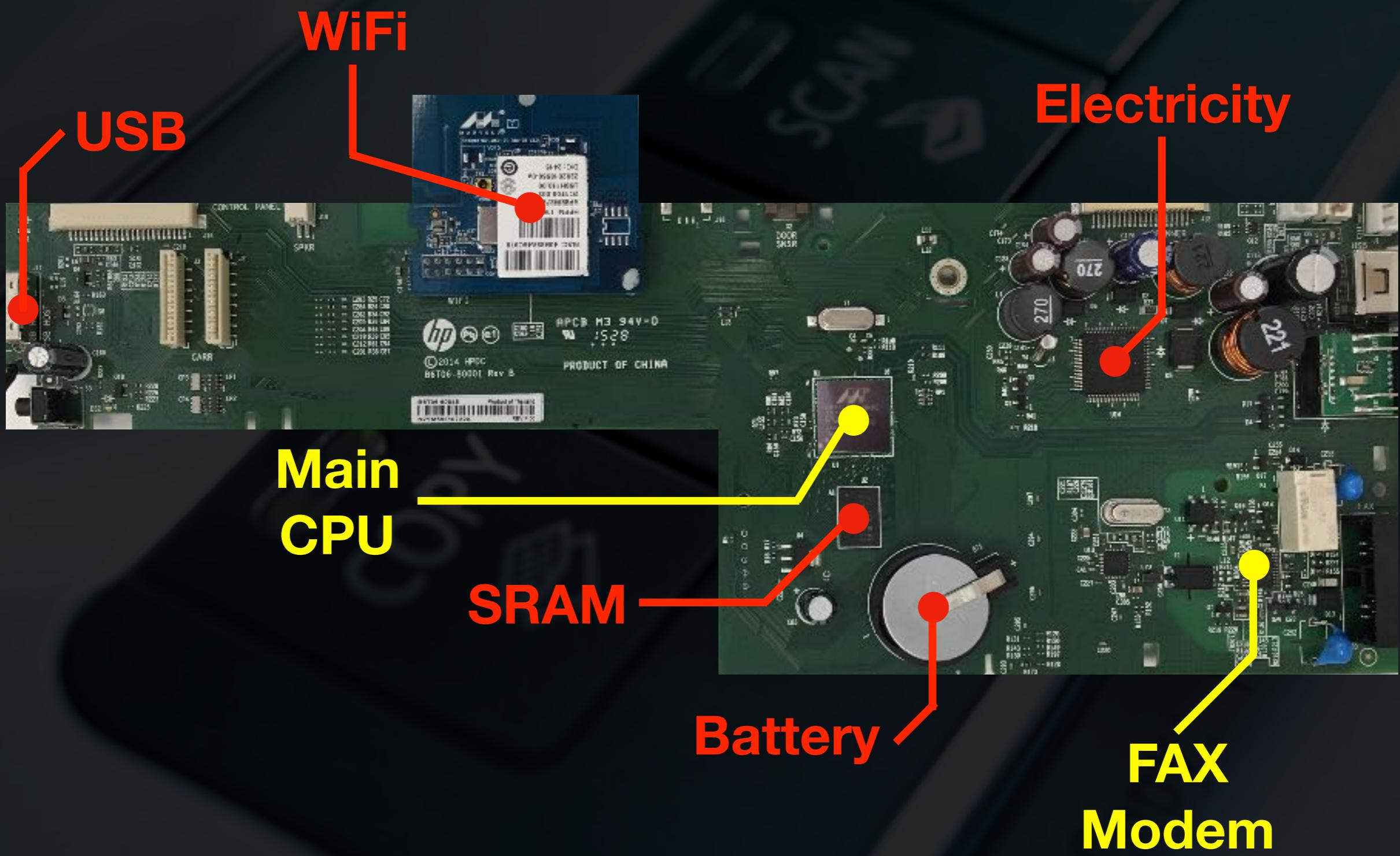
BreakingHW

SRAMs
(e.g Some More Memory)

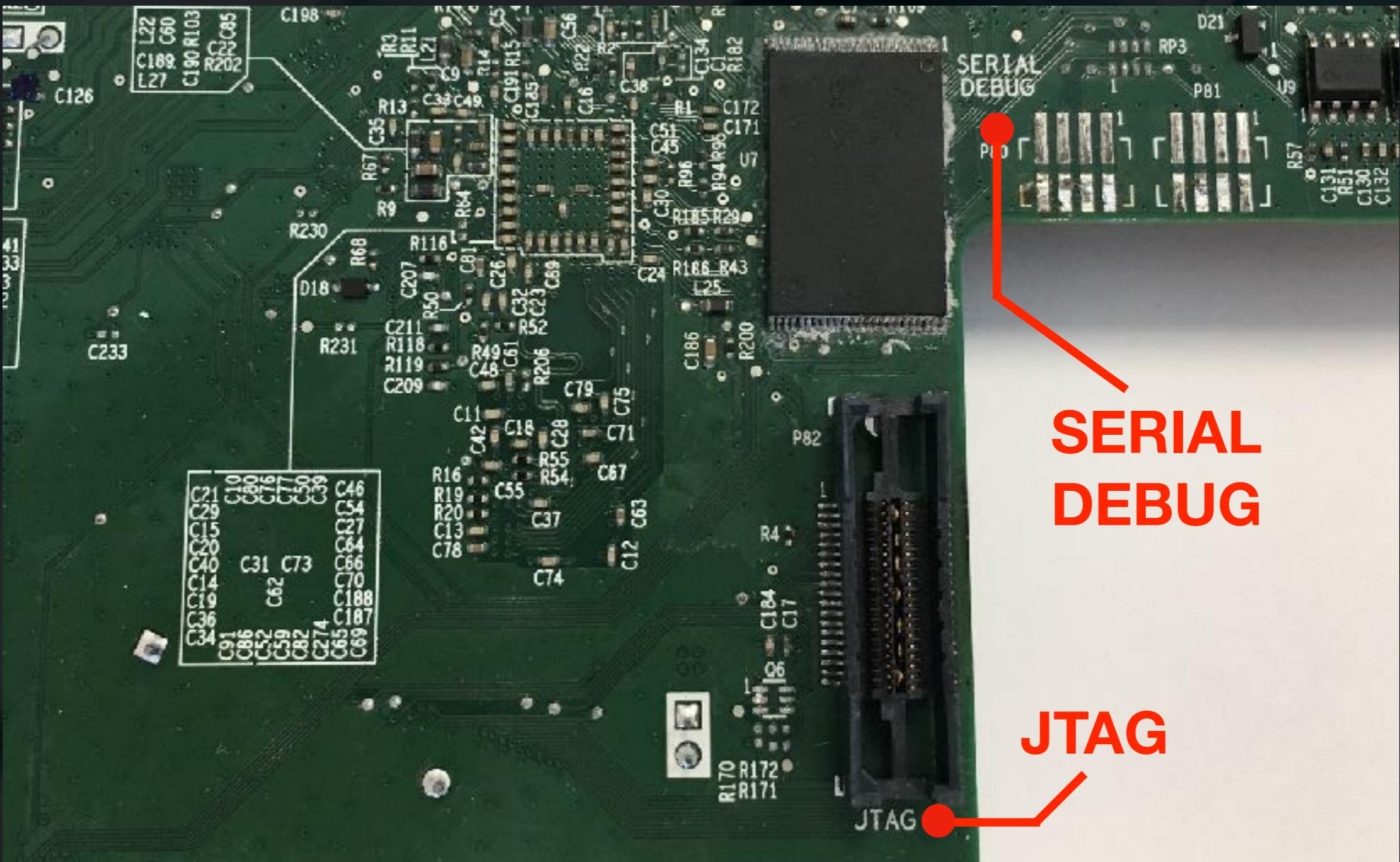


Flash ROM

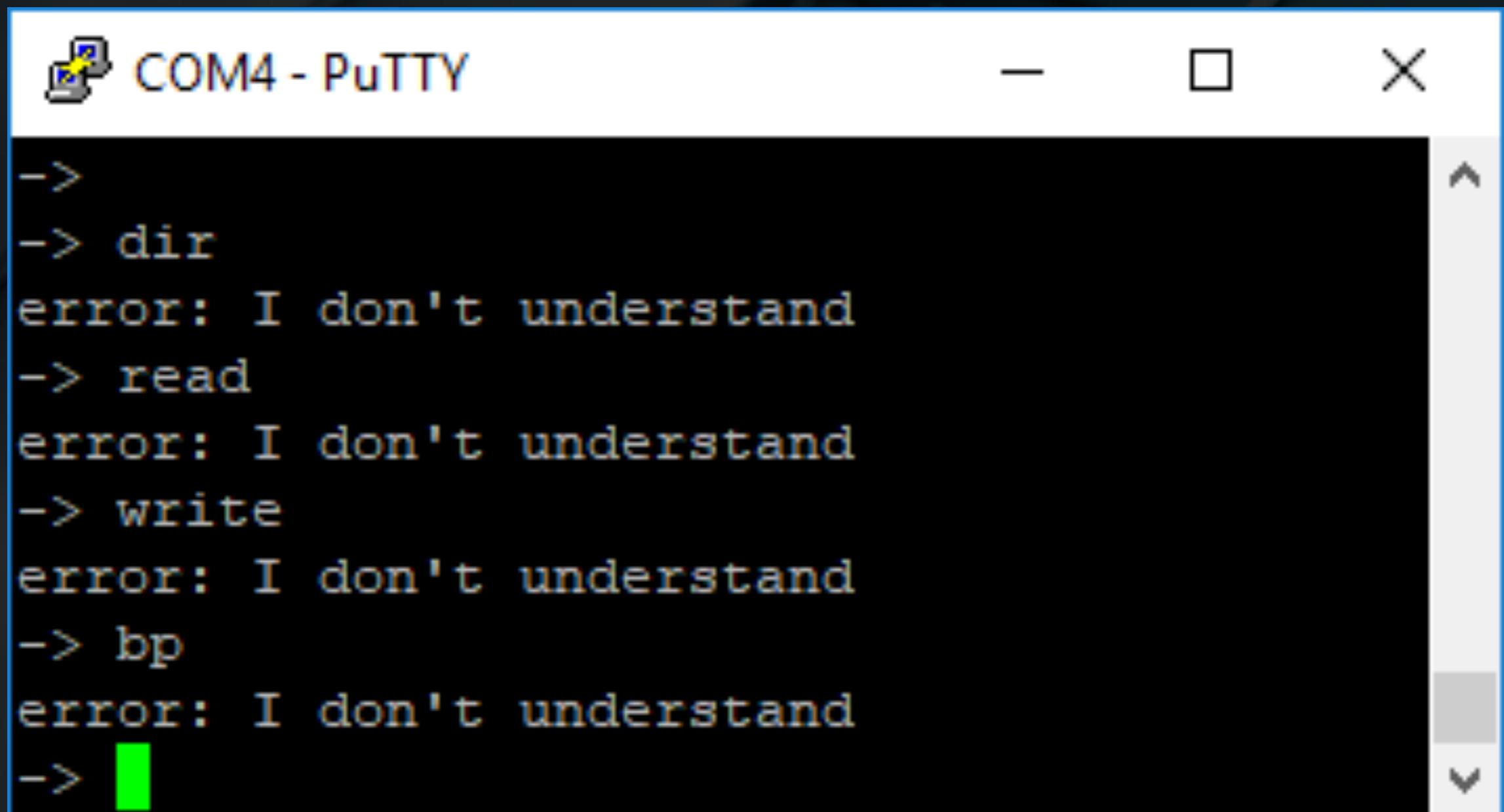
BreakingHW



ShowMeYourFirmware!



TooEasy?



COM4 - PuTTY

```
->
-> dir
error: I don't understand
-> read
error: I don't understand
-> write
error: I don't understand
-> bp
error: I don't understand
-> █
```

Firmware Upgrade

Index of /pub/

[\[parent directory\]](#)

	Name	Size	Date Modified
📁	IPG/		11/22/08, 2:00:00 AM
📁	RPOS/		7/12/16, 3:00:00 AM
📁	all_in_one/		3/6/09, 2:00:00 AM
📁	alphaserver/		11/23/08, 2:00:00 AM
📁	automatic/		6/25/09, 3:00:00 AM
📁	c-products/		10/29/09, 2:00:00 AM
📁	c-storage/		4/26/11, 3:00:00 AM
📁	calculators/		12/15/14, 2:00:00 AM
📁	caps-softpaq/		5/25/18, 5:42:00 AM
📁	catia/		3/6/09, 2:00:00 AM
📁	ctovideo/		11/9/07, 2:00:00 AM
📁	device_manager/		9/1/17, 3:00:00 AM
📁	diag/		11/12/07, 2:00:00 AM
📁	docs/		1/24/17, 2:00:00 AM
📁	dpne/		7/8/09, 3:00:00 AM
📁	enterprise/		9/15/07, 3:00:00 AM
📁	essentials/		9/15/07, 3:00:00 AM
📁	euro/		11/23/08, 2:00:00 AM
📁	extaccel/		1/4/10, 2:00:00 AM
📁	faxes/		9/15/07, 3:00:00 AM
📁	futuresmart/		6/27/11, 3:00:00 AM
📁	graham/		8/31/17, 3:00:00 AM
📁	gsb/		7/7/14, 3:00:00 AM
📁	gsy/		3/6/09, 2:00:00 AM
📁	handheld_computers/		9/15/07, 3:00:00 AM
📁	hp_StoreOnce/		12/8/14, 2:00:00 AM
📁	hp_StoreOnceRMC/		5/5/16, 3:00:00 AM
📁	hp_easymenu/		9/30/11, 3:00:00 AM
📁	hpcast/		3/18/08, 2:00:00 AM
📁	laser/		11/23/08, 2:00:00 AM

How do you
upgrade a printer
firmware?!



You Print it!

PCL XL Feature Reference Protocol Class 2.1 Supplement



PCL XL Feature Reference Protocol Class 2.1 Supplement

Revision: 1.0

Revision Date: August 9, 2000

Author:

Word for Windows File: xl_ref21.doc

Document Revision History

Rev	Revision Description	Date	Author
1.0	Release	09Aug2000	



You Print it!

PrintingTheFirmware

1B	25	2D	31	32	33	34	35	58	40	50	4A	4C	20	43	4F	.%-12345X@PJL CO
4D	4D	45	4E	54	20	28	6E	75	6C	6C	29	0A	40	50	4A	MMENT (null).@PJ
4C	20	45	4E	54	45	52	20	4C	41	4E	47	55	41	47	45	L ENTER LANGUAGE
3D	46	57	55	50	44	41	54	45	0A	1B	45	54	68	69	73	=FWUPDATE..ETHis
20	64	65	76	69	63	65	20	64	6F	65	73	20	6E	6F	74	device does not
20	73	75	70	70	6F	72	74	20	46	57	55	50	44	41	54	support FWUPDAT
45	21	0D	0A	1B	2A	72	74	31	36	33	38	34	73	41	1B	E!...*rt16384sA.
2A	62	31	36	35	34	32	59	1B	2A	62	2B	30	59	1B	2A	*b16542Y.*b+0Y.*
62	31	36	33	38	34	56	53	41	32	37	30	32	30	32	30	b16384VSA2702020
31	45	43	32	45	34	41	30	30	30	38	30	31	30	30	38	1EC2E4A000801008
46	41	35	36	43	32	33	42	30	45	34	36	30	36	41	41	FA56C23B0E4606AA
42	36	38	41	33	33	32	34	42	37	46	34	37	37	31	37	B68A3324B7F47717
42	41	39	33	41	45	39	30	36	33	35	38	46	39	44	37	BA93AE906358F9D7
30	38	31	31	38	46	32	33	32	0A	53	41	32	37	41	32	08118F232.SA27A2
39	43	32	33	32	33	30	31	43	31	39	46	30	41	46	42	9C232301C19F0AFB
44	41	30	35	44	35	37	30	35	31	38	36	44	41	39	37	DA05D5705186DA97
45	39	39	41	45	32	41	44	36	38	41	46	31	43	42	39	E99AE2AD68AF1CB9
45	46	34	42	34	36	41	36	44	43	37	43	46	41	39	41	EF4B46A6DC7CFA9A
30	35	39	39	30	38	39	43	30	44	32	33	0A	53	41	32	0599089C0D23.SA2

PrintingTheFirmware



NULL Decoder

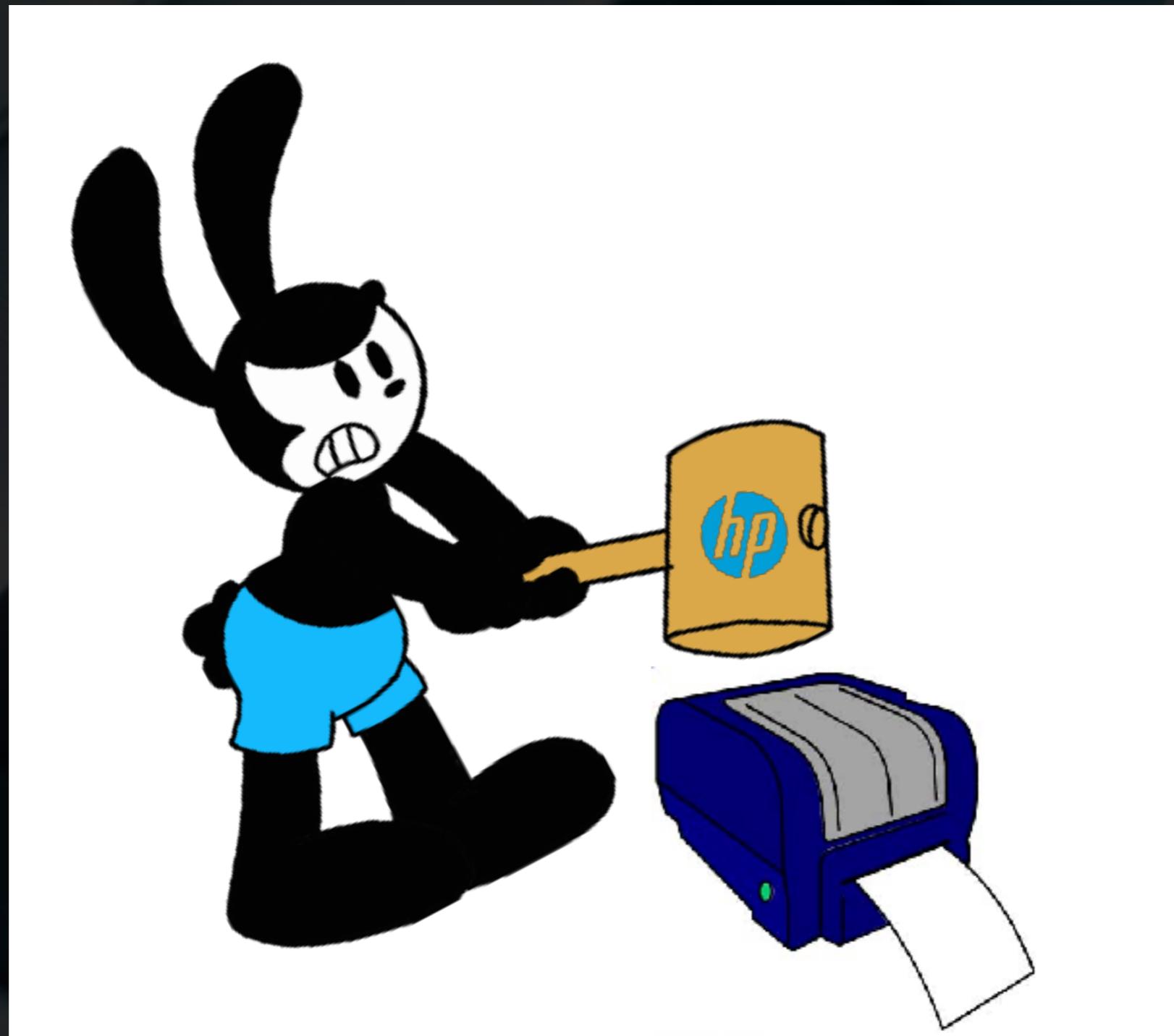


TIFF Decoder



Delta Raw Decoder

When You're a Hammer...



Sections

The screenshot shows a debugger interface with multiple tabs at the top: Program Segmentation, Hex View-1, Enums, Imports, and Exports. The main window displays a table of section information. A yellow bracket on the left groups the first four columns, which are labeled "Loading Address" in a yellow box below. A yellow bracket on the right groups the last two columns, which are labeled "Location in Binary" in a yellow box below. The middle column, "Section Name", is also highlighted with a yellow box.

secinfo:267784A4	section_load_addr <0x210F4088, romrevstr_rodata, 0x2C>; "PNP1CN1521AR"	secinfo:267784B0	section_load_addr <0xA007CD20, romnosi_text, 0>	secinfo:267784BC	section_load_addr <0x810, cromload_text, 0>;
	section_load_addr <0x60, romnosi_text, 0x7B0>		section_load_addr <0x810, romfast_imem1_text, 0x310>		section_load_addr <0xB20, romfast_imem2_text, 0x2BC>
	section_load_addr <0xDDC, romfast_imem3_text, 0>		section_load_addr <0x2009FD8C, crom_load_rodata, 0x277F>		section_load_addr <0xA007CD60, cromload_data, 0x3CFA>
	section_load_addr <0x60, 88A0, cromload_nodata+0x13C, 0>		section_load_addr <0x21656E70, cromfixcon_display, 0>		section_load_addr <0x210FADCC, cromfixcon_display, 0x3B6D86>
secinfo:26778594	section_load_addr <0xB20, romfast_imem2_text, 0x2BC>	secinfo:267785A0	section_load_addr <0xDDC, romfast_imem3_text, 0>	secinfo:267785AC	section_load_addr <0x2009FD8C, crom_load_rodata, 0x277F>
secinfo:267785B8	section_load_addr <0xA007CD60, cromload_data, 0x3CFA>	secinfo:267785C4	section_load_addr <0x600788A0, cromload_nodata+0x13C, 0>	secinfo:267785D0	section_load_addr <0x21656E70, cromfixcon_display, 0>
secinfo:267785DC	section_load_addr <0x210FADCC, cromfixcon_display, 0x3B6D86>				04A>
secinfo:26778630	section_load_addr <0x20B377E4, crommodule, 0x742>	secinfo:2677863C	section_load_addr <0x20B37EC0, cromfs, 0x500>	secinfo:26778648	section_load_addr <0x20B384B0, cromfsobjs, 0x32B5>
secinfo:26778654	section_load_addr <0x200A4DCC, unk_27FF47CC, 0>	secinfo:26778660	section_load_addr <0x200A4DC0, unk_27FF47CC, 0>	secinfo:2677866C	section_load_addr <0x200A4DC0, unk_27FF47CC, 0>
secinfo:26778678	section_load_addr <0x200890C0, cromload_text, 0x11BE9>	secinfo:26778684	section_load_addr <0x200A4DCC, cromtext, 0x6D5A08>		
secinfo:26778690	section_resethole section_header <section_nosi_text, a_reset_hole, 0, 0x44, 4, 0>; ".reset_hole"	a_reset_hole DCB ".reset_hole", 0			; DATA XREF: secinfo:section_resethole+0
secinfo:267786A8					
secinfo:267786B4		section_nosi_text section_header <section_nosi_rodata, a_nosi_text, 0x60, 0x7B0, 1, 0>			; DATA XREF: secinfo:section_resethole+0
secinfo:267786B4					
secinfo:267786B4					

I Don't Understand

DF	25	34	2E	B9	34	D4	DF	3F	D4	32	2E	32	0B	B3	65	.%4..4..?..2.2..e
7F	72	72	6F	72	3A	20	49	14	B0	F7	6E	27	74	F1	A0	.rror: I...n't..
64	65	72	73	E2	E8	B0	64	32	C0	12	EF	24	E4	20	3C	ders...d2...\$. <
25	B7	73	3E	0A	2D	E5	4D	69	5A	A0	6E	9F	67	20	52	%s>.-.MiZ.n.g R
48	53	47	E4	F3	D1	01	FF	D3	06	D0	02	F2	0F	20	54	HSG..... T
EB	47	70	70	E6	58	B2	61	44	F0	B0	FF	A1	46	07	46	.Gpp.X.aD....F.F
0E	F6	4A	20	BF	10	F2	CA	20	E1	68	97	60	2F	FF	01	..Jh.^/..



65	6C	66	FF	20	72	65	63	75	72	73	69	EF	76	65	6C	elf. recursi.vel
79	AE	E0	6E	6F	6E	DF	70	6F	73	69	74	FE	30	20	73	y..non.posit.0 s
F7	69	7A	65	0E	32	76	61	72	69	FF	61	62	6C	65	2D	.ize.2vari.able-
6C	65	6E	F7	67	74	68	AD	33	00	00	56	4C	39	41	B7	len.gth.3..VL9A.
35	9E	21	66	61	69	6D	32	4D	30	97	64	65	6C	72	E0	5.!faim2M0.delr.

WhatISThis?!

- Probably a compression algorithm
- A very bad one ...
- Some mathematics

Let's Take A Look

r e c u r s i v e l y
n o n p o s i t s i z
e v a r i a b l e - l e n
g t h V L j p e g .

FF 20 72 66 63 75 72 73 69 EF 76 65 6C 79 AE E0
6E 6F 6E DF 70 6F 73 69 74 FE 30 20 73 F7 69 7A
65 0E 32 76 61 72 69 FF 61 62 6C 65 2D 6C 65 6E
F7 67 74 68 AD 33 00 00 56 4C FF 6A 70 65 67 2E

Let's Take A Look

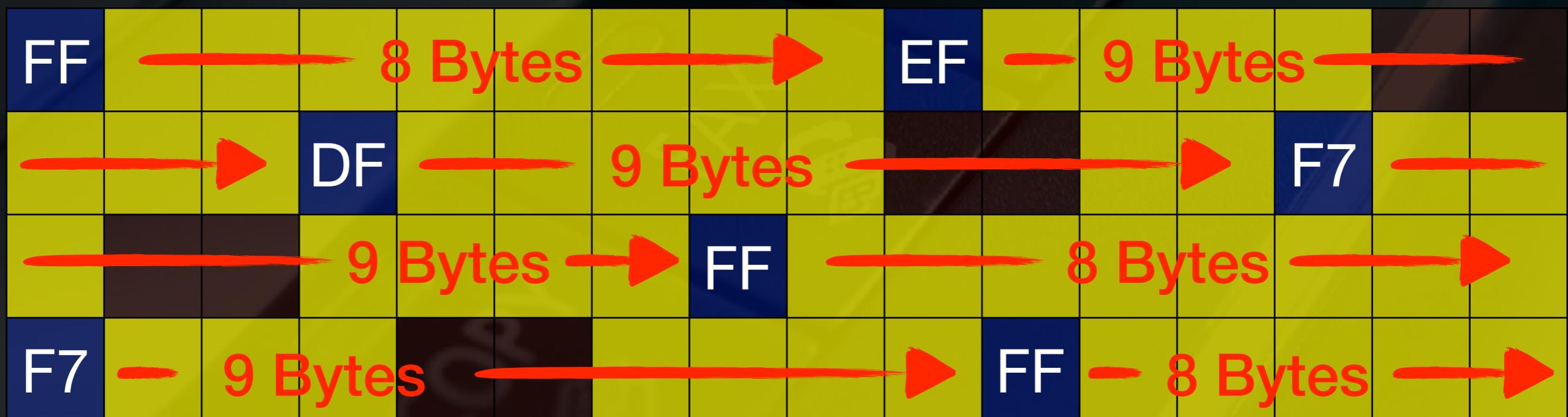
		r	e	c	u	r	s	i		v	e	l	y		
n	o	n		p	o	s	i	t			s		i	z	
e			v	a	r	i		a	b	l	e	-	l	e	n
g	t	h						V	L		j	p	e	g	.

FF	20	72	66	63	75	72	73	69	EF	76	65	6C	79	AE	E0
6E	6F	6E	DF	70	6F	73	69	74	FE	30	20	73	F7	69	7A
65	0E	32	76	61	72	69	FF	61	62	6C	65	2D	6C	65	6E
F7	67	74	68	AD	33	00	00	56	4C	FF	6A	70	65	67	2E

APattern?!

FF									EF							AE	E0
				DF					FE	30				F7			
	0E	32						FF									
F7				AD	33						FF						

A Pattern?!



Different Angle

FF	F	F	1	1	1	1	1	1	1	1	1
EF	F	E	1	1	1	1	0	1	1	1	1
DF	F	D	1	1	1	1	1	0	1	1	1
F7	7	F	1	1	1	0	1	1	1	1	1
FF	F	F	1	1	1	1	1	1	1	1	1
F7	7	F	1	1	1	0	1	1	1	1	1

The Missing Link



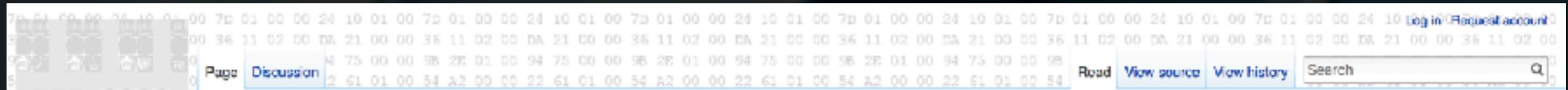
Forward / Backward Pointer

Dictionary

Sliding Window



Softdisk



Softdisk Library Format

SLIB, or Softdisk LIBRARY, compression is a container file format used by [Softdisk](#) software to compress various files used by their games, most notably the Commander Keen Dreams series of games, (Including Dangerous Dave 3 and Dangerous Dave 4)to store title images used at the beginning of each game. It was created in 1992 by Jim Row.

The data held in the file can be compressed in any one of three ways, uncompressed, LZW and LZH. The compression used is primitive and rather different from later or traditional versions of LZW/LZH. SLIB files were created by the program **S0MILIB.EXE** and as such any game that uses this format contains various segments of code in common with **S0FT1.TB** for the decompression of data.

There is a closely related format, the SHL or Softdisk Help Library format. SHL files contain only a single file. Their header is slightly different, its file signature is 'CMP1' (CoMPression of 1 file) while that of SLIB files is SLIB. The veracity of both files can be confirmed by checking for a word of value 2 at offset 4 in the file. The actual files have been given a number of extensions; .CNP (CoMPressed), .SHL (Softdisk Help Library) or the same extension.

The SLIB file can roughly be broken into a number of parts; the header, which contains data about the various data chunks, and the data chunks themselves, each containing a single file. Each chunk also has a short header.

Softdisk Library Format	
Format type	Archive
Max files	65,535
File Allocation Table (FAT)	
Filenames?	Yes, 8.3
Metadata?	None
Supports compression?	Yes
Supports encryption?	No
Supports subdirectories?	No
Hidden data?	Yes
Games	Commander Keen Dreams Dangerous Dave 3 Dangerous Dave 4

[Contents](#) | [hide](#)

- [1 Header](#)
 - [2 Data Chunks](#)
 - [3 Compression](#)
 - [3.1 LZW](#)
 - [3.2 LZH](#)
 - [4 Soflib](#)
 - [5 Data contained in Libraries](#)
 - [6 Tools](#)
 - [7 Credits](#)

Header

The file header is found only in SLIB files and is absent in SHL files, which are loaded into memory in their entirety. The SLIB header allows individual data chunks to be loaded into memory separately.

The SLIB header is a variable length header that contains information about how many chunks there are in a file as well as their location in the file and size. It is used by the game to load chunks.

COMMANDER KEEN

IN
COM
CUD
CUD

DISTRIBUTED BY
APOCEE



Press
F1
for Help

The Missing Link



Mystery Solved

Sliding Window

Input Text

A B C D A B E F G

Output Text

Mystery Solved

Sliding Window

A

Input Text



A B C D A B E F G

Output Text

A

Mystery Solved

Sliding Window

A B

Input Text

A B C D A B E F G



Output Text

A B

Mystery Solved

Sliding Window

A B C

Input Text

A B C D A B E F G



Output Text

	A	B	C								
--	---	---	---	--	--	--	--	--	--	--	--

Mystery Solved

Sliding Window

A B C D

Input Text

A B C D A B E F G



Output Text

	A	B	C	D							
--	---	---	---	---	--	--	--	--	--	--	--

Mystery Solved

Sliding Window

A B C D

Input Text

A B C D A B E F G

Output Text

A B C D

Mystery Solved

Sliding Window

A B C D

Input Text

A B C D A B E F G

Output Text

	A	B	C	D							
--	---	---	---	---	--	--	--	--	--	--	--

Mystery Solved

Sliding Window

A B C D

Input Text

A B C D A B E F G

Output Text

A B C D 00 02

Mystery Solved

Sliding Window

A B C D E

Input Text

A B C D A B E F G



Output Text

	A	B	C	D	00	02	E				
--	---	---	---	---	----	----	---	--	--	--	--

Mystery Solved

Sliding Window

A B C D E F

Input Text

A B C D A B E F G



Output Text

	A	B	C	D	00	02	E	F		
--	---	---	---	---	----	----	---	---	--	--

Mystery Solved

Sliding Window

A B C D E F G

Input Text

A B C D A B E F G



Output Text

	A	B	C	D	00	02	E	F	G		
--	---	---	---	---	----	----	---	---	---	--	--

Mystery Solved

Sliding Window

A B C D E F G

Input Text

A B C D A B E F G



Output Text

1 1 1 1 0 1 1 1

	A	B	C	D	00	02	E	F	G		
--	---	---	---	---	----	----	---	---	---	--	--

Mystery Solved

Sliding Window

A B C D E F G

Input Text

A B C D A B E F G

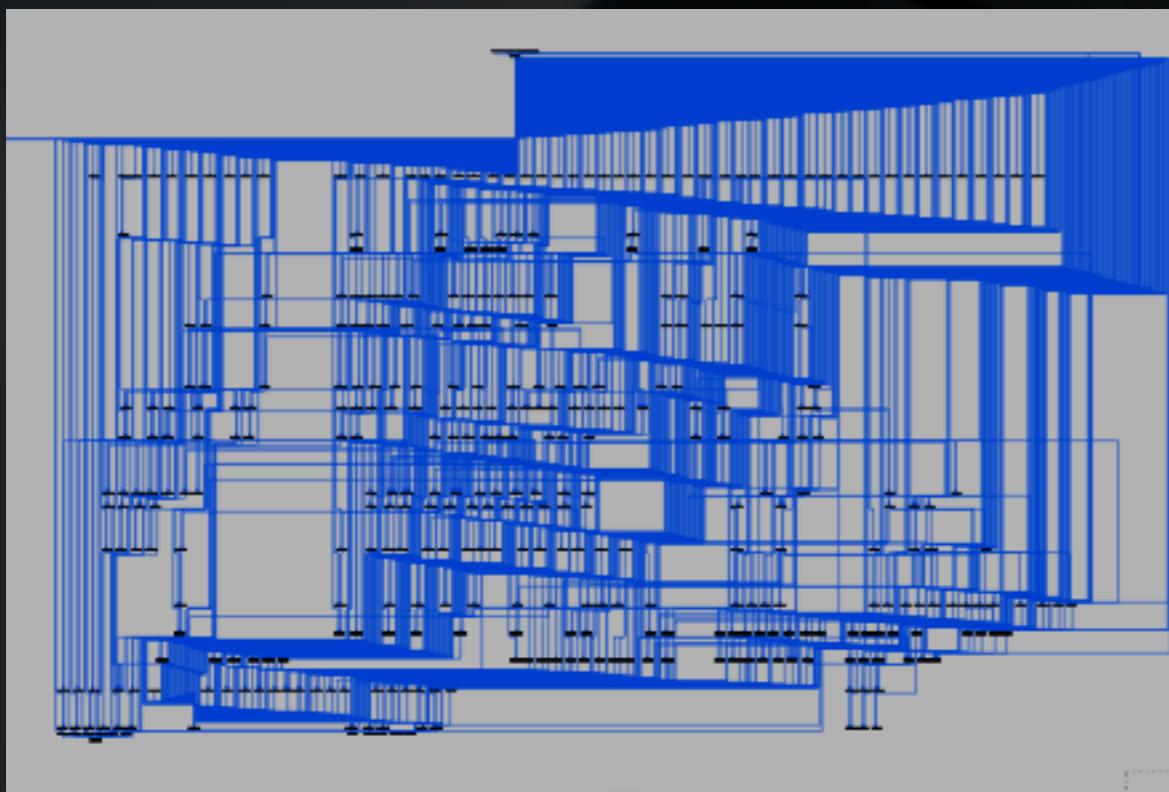


Output Text

E F A B C D 00 02 E F G

ThePrintingBeast

- 64,709 Functions
 - Most of the code not parsed by IDA
 - Indirect Calls, Dynamic Tables, BootLoader Functions



MakingSomeSense

ThreadX
- ARM9/
Green
Hills

Common Libraries

mDNSResponder

Spidermonkey

OpenSSL 1.0.1j (2014)

gSOAP 2.7

libpng 1.2.29 (2008)

System n' Stuff

Treck (IP, TCP/UDP, DNS, HTTP, ...)

2 Staged Boot Loader

Tasks

tPrintFax

tT30

tFaxLog

tModem

tTB, tHTML, ...

Making Some Sense

ThreadX
- ARM9/
Green
Hills

Common Libraries

mDNSResponder

OpenSS

Spidermonkey

gSOAP 2.7

libpng 1.2.29



Treck (IP, TCP,

2 Stage

Tasks

tPrintFax

tT30

axLog

tModem

tTB, tHTML, ...

soff

, HTTP, ...)

Loader

JSONOnAPrinter?!

- JavaScript is used in a module called PAC.
- PAC - Proxy Auto Configuration
- Used by a URL linking to a JS file in DHCP settings
- Top layer functionality was designed by HP

FakeURL

The screenshot shows a debugger interface with multiple windows open at the top, including Pseudocode, IDA Pro, and various memory dump tabs. The main window displays assembly code for a function named `yl_sub_20A23DD8`. The code is annotated with line numbers from 40 to 78. The assembly uses standard conventions like `v10`, `v9`, and `v12` for registers, and `dword_A2315C38` for memory locations. It includes calls to external functions like `sub_20A05258` and `sub_20B2E484`, and handles strings related to URLs. The code is written in a mix of C-like pseudocode and raw assembly.

```
● 40     if ( v10 != -98 )
● 41     {
● 42         sub_20A05258(v9);
● 43         sub_20A05258(dword_A2315C38);
● 44         dword_A2315C38 = -1;
● 45         return v10;
● 46     }
● 47     if ( (v12 - 301) > 2 && v12 != 307 )
● 48     {
● 49         sub_20A05258(v9);
● 50         sub_20A05258(dword_A2315C38);
● 51         result = -1;
● 52         dword_A2315C38 = -1;
● 53         return result;
● 54     }
● 55     (*(*dword_A320E290 + 12))(dword_A320E290, v9, "Location", &arg1, 257);
● 56     if ( yl strstr(&arg1, "http://") )
● 57     {
● 58         yl strncpy__(&dst, &arg1, 256);
● 59         sub_20A05258(v9);
● 60     }
● 61     else
● 62     {
● 63         v11 = YA_strlen(&dst);
● 64         sub_20B2E484(&dst, &v15, 256 - v11);
● 65     }
● 66 }
● 67     if ( !yl_sub_20A24370(&unk_A320E294, "fakeurl1234.com", "fakeurl1234.com", &a4a) )
● 68     {
● 69         if ( a4a )
● 70         {
● 71             yl_sub_20A24030(a4a, v5, v6, v7);
● 72             yl_js_free(a4a);
● 73             a4a = 0;
● 74         }
● 75     }
● 76     sub_20A05258(v9);
● 77     return 0;
● 78 }
```

0097F128 | yl_sub_20A23DD8:55 (20A23EF4)

Yep...



T30

- aka “ITU-T Recommendation T.30”
- Procedures for document facsimile transmission in the general switched telephone network
- Defined the “heavy lifting” procedures relevant for all fax sending functionality
- Designed at 1985
- Last update at 2005

DynamicHell



```
1 // a1 = 7 i think
2 int __fastcall yb_t30_main_fax_
3 {
4     unsigned int current_state;
5     int v2; // r6
6     int stop_inner_loop; // r0
7     signed int v4; // r0
8     int v5; // r0
9     int v6; // r0
10    int v7; // r0
11    int v8; // r6
12    int v9; // r0
13    some_cb_struct_t arg2; // [sp]
14
15    current_state = yl_fax_init_cc
16    yb_SetT30State_1(T30_STATE1_CU
17    v2 = 0;
18 LABEL_2:
19    stop_inner_loop = yl_g_fax_stop_1
20    if (!yl_g_fax_stop_inner_loop)
21        goto LABEL_109;
22    do
23    {
24        while ( 1 )
25    {
26        yl_g_fax_stop_inner_loop = 0;
27        while ( sub_204BF816[yb_g_fax_state_] & 1
28        ;
29        yl_yblock_put(yl_g_yblock_semail_A2E860F0);
30        current_state = yb_GetT30State_();
31        stop_inner_loop = yl_g_fax_stop_inner_loop;
32        if ( current_state == 43 )
33            break;
34    switch_start:
35        while ( !stop_inner_loop )
```

```
; DATA XREF: sub_209F5230+4rc
; yb_ChngFaxParam+21o ...
; DATA XREF: sub_209F576A+Etw
; yb_FAX_PrintConnData__+C8tw ...
; DATA XREF: sub_209F576A+14tw
; yb_FAX_PrintConnData__+80tw ...

; DATA XREF: sub_209F576A+28rc
; yb_FAX_PrintConnData__:_loc_209F58A2+r
; DATA XREF: yb_FAX_PrintConnData__+F0+r
; yb_FAX_PrintConnData__:_loc_209F5920+r
; DATA XREF: yb_FAX_PrintConnData__+42+r
; yb_FAX_PrintConnData__+58+r ...
```

```
; DATA XREF: sub_209F5B24+55rw
; sub_209F5B24+82+r ...
; DATA XREF: sub_209F5B24+53rw
; sub_209F5B24:_loc_209F5B8E+r ...
```

```
KREF: sub_209F56B2+28rc
DE56FC+1E+o ...
```

```
6 1
7 1
8 1
9 1
10 1
11 1
12 1
13 1
14 1
15 1
16 1
17 1
18 1
19 1
20 1
21 1
22 1
23 1
24 1
25 1
26 1
27 1
28 1
29 1
30 1
31 1
32 1
33 1
34 1
35 1
```

TheUndebuggable

- How do we debug this hostile environment?
- There are no native debugging facilities
- We have no control over the execution flow
- Hardware watch-dog is a serious problem

LuckyBreak

- Luck is a fundamental part of every research project
- At July 19, SENRIO published an exploit dubbed “Devil’s Ivy”
- CVE-2017-9765 - RCE in gSOAP 2.7 - 2.8.47
- And it seems our printer is vulnerable!

Devil'sIvy

```
int __fastcall yl_soap_get_pi(soap_t *soap)
{
    char *s; // r6
    signed int i; // r7
    signed int c_1; // r4
    unsigned int c; // r0
    unsigned __int8 *v6; // r6
    char buf[64]; // [sp+0h] [bp-54h]

    s = buf;
    i = 64;
    while ( 1 )
    {
        c = yl_soap_getchar(soap);
        c_1 = c;
        if ( c == -1 || c == '?' )
            break;
        if ( --i > 0 )
        {
            if ( c < 0x21 )                                // soap_blank(c)
                LOBYTE(c_1) = 0x20;
            *s++ = c_1;
        }
        *s = 0;
        if ( !yb_strncmp(buf, "xml ", 4) )
        {
            v6 = yl strstr(buf, " encoding=");
            if ( v6 )
                if ( !yb_strcmp(v6, "UTF-8") )
                    return 1;
        }
    }
}
```

Debugging Challenges

- Need to read/write memory
- Need to Execute code
- Create a network tunnel between debugger/debuggee

Debugging Challenges

- We have control over execution flow
- Need to load our own code
- Bypass memory protection
- Embed debugging stub into current firmware

Scout

- We created our own instruction based debugger
- Called - 'Scout'
- Supports x86, x64, ARM (ARM and Thumb mode)
- Embedded mode for firmware
- Linux kernel mode

How Does A FAX?



How Does A FAX?



PHASE 1



Network
Interaction

PHASE 2

Probing/
Ranging

PHASE 3

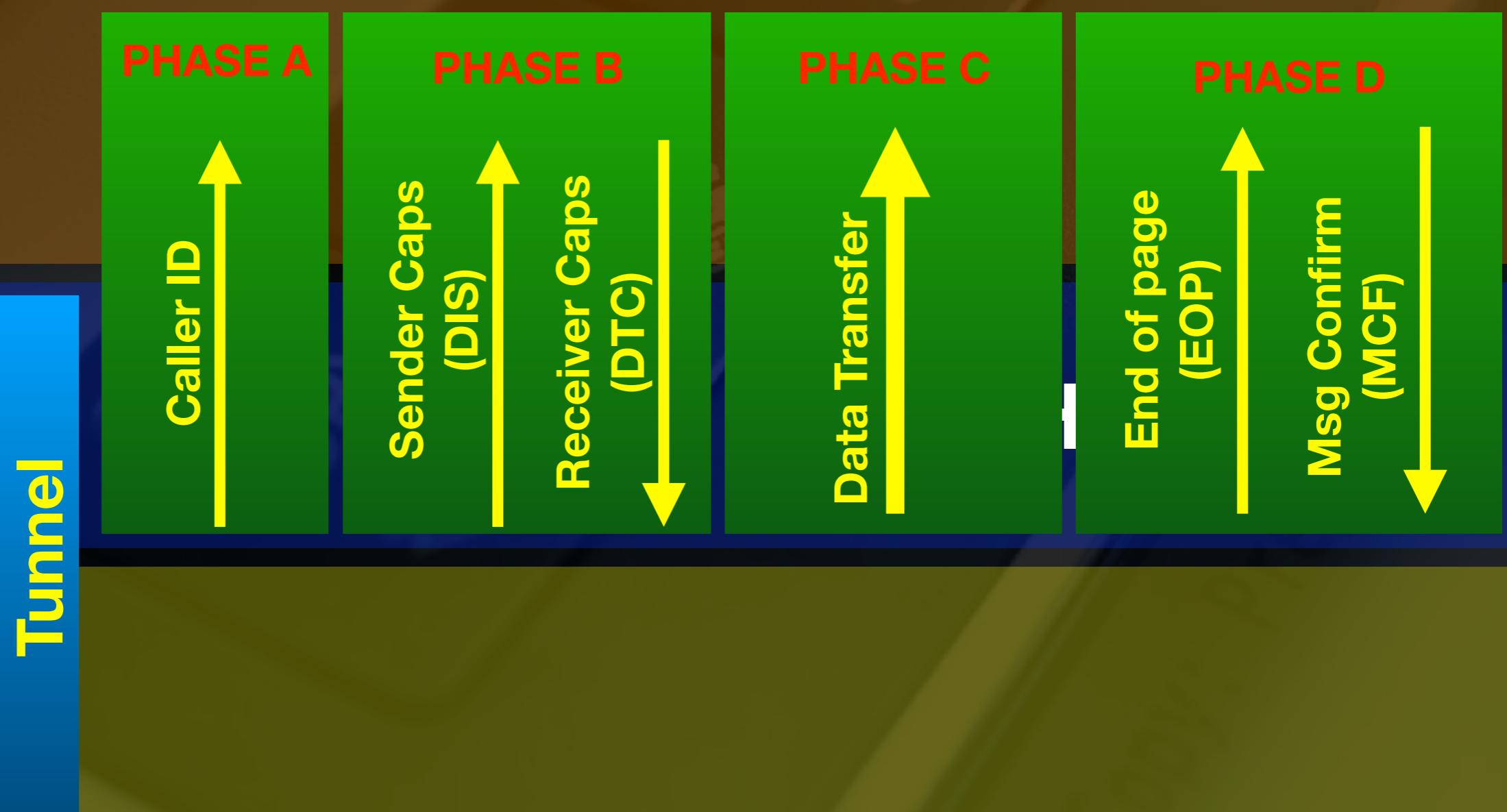
Equalizer
and
Echo
Canceller
Training

PHASE 4

Training
Phase



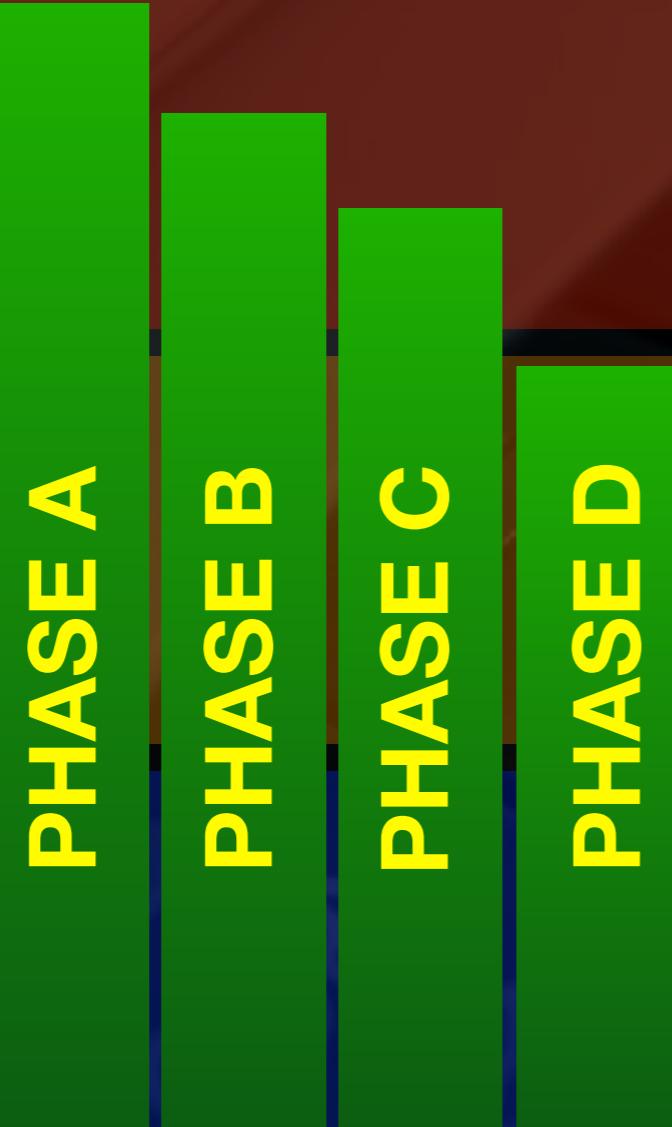
How Does A FAX?



How Does A FAX?



Tunnel



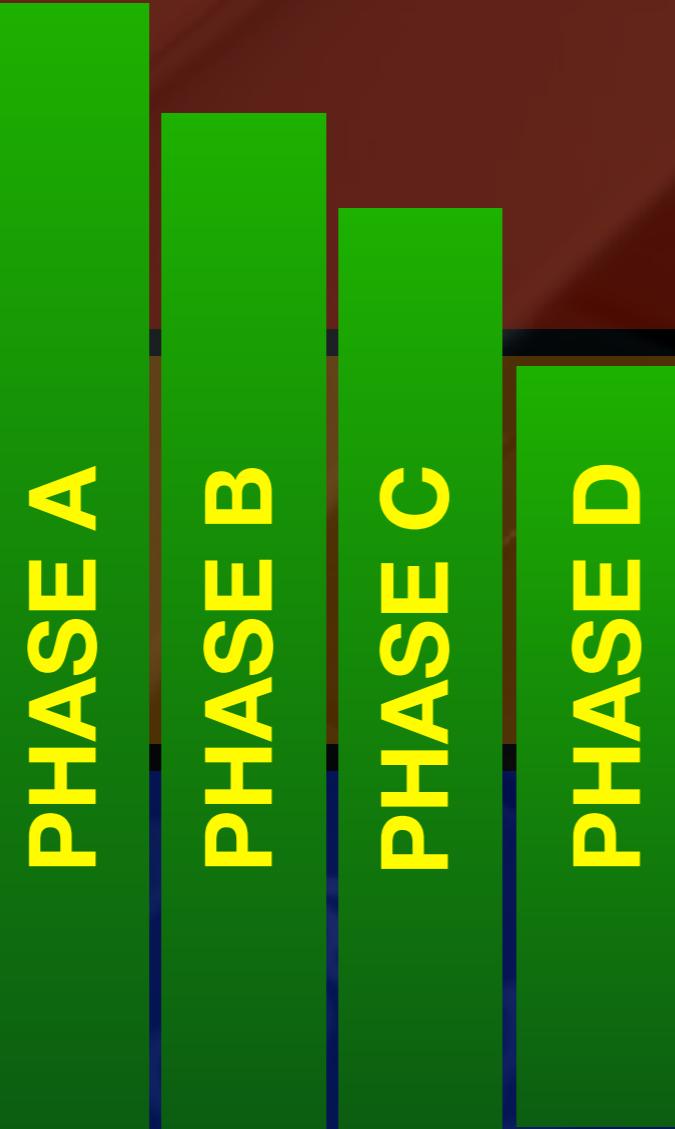
T.30

HDLC

How Does A FAX?



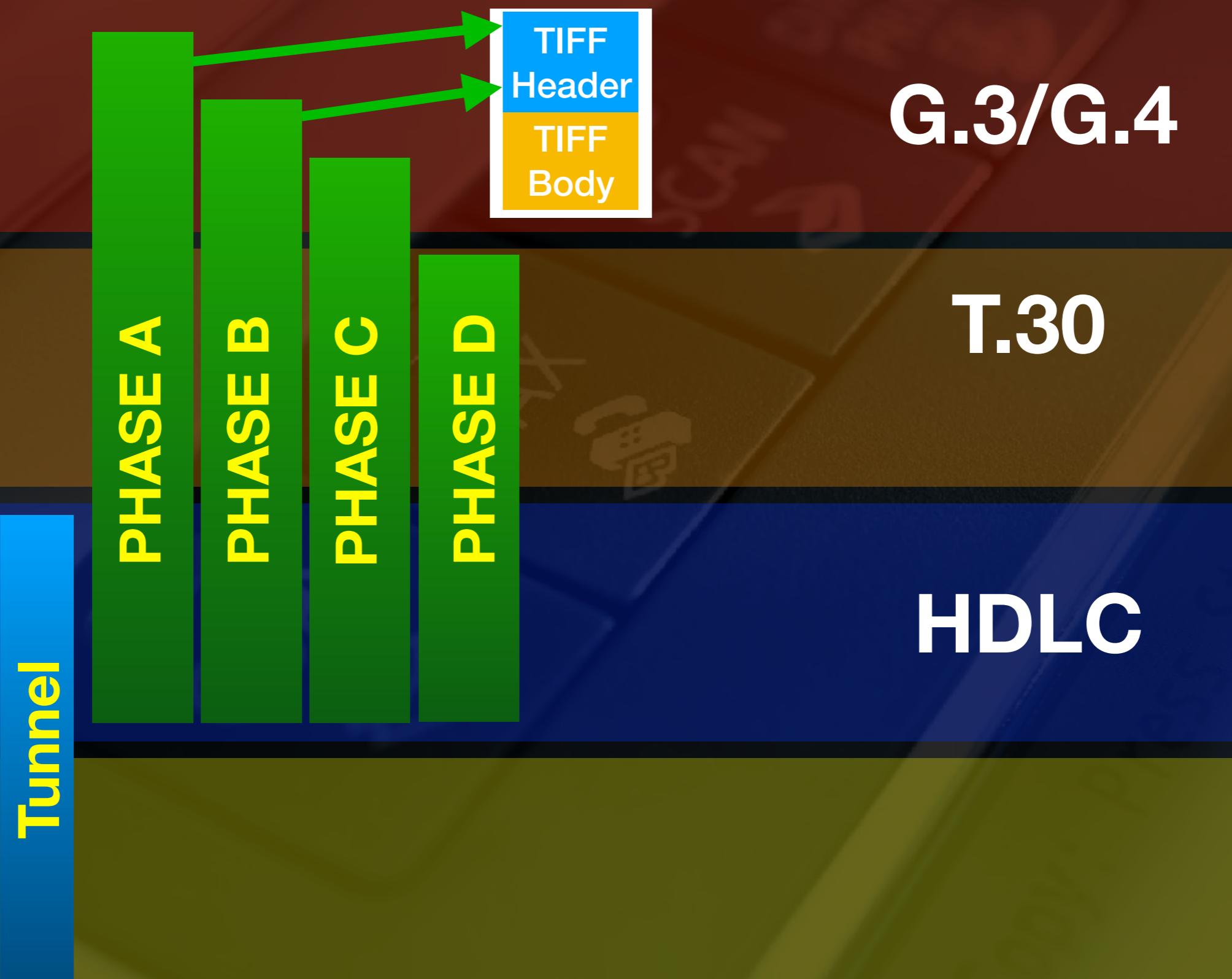
Tunnel



T.30

HDLC

How Does A FAX?



How Does A FAX?



Tunnel

PHASE A



PHASE B

PHASE C

PHASE D

HDLC

T.30
Color Extension

How Does A FAX?



Tunnel

PHASE A

PHASE B

PHASE C

PHASE D

JPEG
Header
and
Body

T.30
Color Extension

HDLC

Vulnerability

- All the layers we showed can contain possible vulnerabilities.
- The most convenient layer is the application one.
- We started by inspecting the JPEG parsing capabilities.

JPEG

FF	D8	FF	E0	00	10	4A	46	49	46	00	01	02	00	00	64JFIF....d
00	64	00	00	FF	C4	0A	02	34	D3	2A	78	80	42	6D	2B	.d.....4.*x.Bn+
FF	DA	12	28	2A	6F	2B	81	6A	16	0F	C8	9A	13	FF	D9	...(*0+.j....)

SOI - Start Of Image

APP0 - Application Specific

Size

Data

DHT - Define Huffman Table

Size

Data



SOS - Start Of Scan

Data

EOI - End Of Image

DHT

- Define Huffman Table
- Defines 4X4 comparison matrix for the JPEG Image



DHT

FF	C4	$\Sigma($	00	01	00	00	00	02	00	01	02	00	00	00
00	00	00) = 6	4	0A	02	34	D3	2A	78	80	42	6D	2B

- 4X4 Matrix values are summed
- The product is used as a size value for data bytes
- The data bytes are copied into a 256 bytes array located on the stack

DHT

FF	C4	20	00	01	00	00	00	00	02	00	01	02	00	00	00	00
00	00	00	FF	FF	C4	0A	02	34	D3	2A	78	80	42	6D	2B	

— 6 —

Stack

- 952 -

Can You Spot It?

```
v2 = EI_jpg_stream_read_byte_from_file();
v3 = v2 >> 4;
v4 = v2 & 0xF;
accumulated_sum_bound_4096 = 0;
loop_index = 0;
do
{
    read_byte = EI_jpg_stream_read_byte_from_file();
    local_buffer[loop_index] = read_byte;
    accumulated_sum_bound_4096 += read_byte;

    ++loop_index;
}
while ( loop_index <= 15 );
huge_short_minus_19 = huge_short_minus_2 - 17;
if ( huge_short_minus_19 < accumulated_sum_bound_4096 )
    break;
y1_dword_zero (local buffer 256, 64);
for ( i = 0; i < accumulated_sum_bound_4096; ++i )
    local_buffer_256[i] = EI_jpg_stream_read_byte_from_file();
huge_short_minus_2 = huge_short_minus_19 - accumulated_sum_bound_4096;
if ( v3 && v3 != 1 || v4 && v4 != 1 )
{
    EI_jpg_set_read_state_opcode(5);
    return;
}
```

DHT

FF	C4	20	FF	FF	FF	FF											
FF	FF	FF	FF	FF	C4	0A	02	34	D3	2A	78	80	42	...	2B		

Stack

-95%

DHT

FF	C4	20	FF	FF	FF	FF											
FF	FF	FF	FF	FF	C4	0A	02	34	D3	2A	78	80	42	...	2B		

— 4000 —

Stack

- 952 -

Overflow!!

ExploitChain

- Trivial stack overflow
- No constraints (“forbidden bytes”)
- ~4,000 user controlled bytes
- The file contains even more information we control...

Demo Time

Conclusions

- PSTN is still a valid attack surface in 2018!
- FAX can be used as a gateway to internal networks
- Old outdated protocols are not good for you...

What Can I Do?

- Patch your printers
- Don't connect FAX where not needed
- Segregate your printers from the rest of the network

**STOP
USING
FAX**

LittleHelpFromMyFriends



**Lior
Oppenheim**



oppenheim1



**Yannay
Livneh**



Yannayli



**Tamir
Bahar**



tmr232



**Yoav
Alon**



yoavalon

fin.



ynvb



Eyalltkin



Check Point®
SOFTWARE TECHNOLOGIES LTD.