Fax?!

What The Fax?!
Who Are We?

Yaniv Balmas
“This should theoretically work”
Security Researcher
Check Point Software Technologies
@

Eyal Itkin
“That’s cool.”
Security Researcher
Check Point Software Technologies
@

@ynvb
@eyalitkin
1846
Alexander Bain Sends An Image Over a Wire

1860
1860
Caselli Invents Machine Similar to Today’s FAX

1923
1923
Enter the RadioFAX. Used by Navies

1966
1966
XEROX Introduces the First Commercial FAX Machine

1980
1980
Group III ITU-T Fax Standards T.30, T.4, T.6

1985
1985
GammaFAX Brings Computers Into FAX Network
<table>
<thead>
<tr>
<th>Quality</th>
<th>Accessibility</th>
<th>Reliability</th>
<th>Authenticity</th>
</tr>
</thead>
<tbody>
<tr>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
</tbody>
</table>

Back To The Future
BackToTheFuture
Back To The Future

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
May 18, 2012

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

Via facsimile: +1-202-456-2461
Back To The Future

Forms for Reporting to FDA

* 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.

- **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
WTF?!
Fax Today

- 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
The Security View

ALL-IN-ONE Printers
FAX Attack

ALL-IN-ONE Printers
Challenge Accepted
What is the Target?

How to Obtain the Code?

What is The OS?

How Does FAX Even Work?

How can we Debug it?

Where to look for vulns?
And The Winner Is
Breaking HW

SRAMs
(e.g. Some More Memory)

Flash ROM
Breaking HW

Main CPU

WiFi

USB

Electricity

Battery

FAX Modem

SRAM
ShowMeYourFirmware!

SERIAL DEBUG

JTAG
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
-> 

### Index of /pub/

<table>
<thead>
<tr>
<th>Name</th>
<th>Size</th>
<th>Date Modified</th>
</tr>
</thead>
<tbody>
<tr>
<td>[parent directory]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IPG/</td>
<td></td>
<td>11/22/08, 2:00:00 AM</td>
</tr>
<tr>
<td>RPOS/</td>
<td></td>
<td>7/12/16, 3:00:00 AM</td>
</tr>
<tr>
<td>all_in_one/</td>
<td></td>
<td>3/6/09, 2:00:00 AM</td>
</tr>
<tr>
<td>alphaserver/</td>
<td></td>
<td>11/23/08, 2:00:00 AM</td>
</tr>
<tr>
<td>automatic/</td>
<td></td>
<td>6/25/09, 3:00:00 AM</td>
</tr>
<tr>
<td>e products/</td>
<td></td>
<td>10/29/09, 2:00:00 AM</td>
</tr>
<tr>
<td>e storage/</td>
<td></td>
<td>4/26/11, 3:00:00 AM</td>
</tr>
<tr>
<td>calculators/</td>
<td></td>
<td>12/15/14, 2:00:00 AM</td>
</tr>
<tr>
<td>caps-softpaq/</td>
<td></td>
<td>5/25/18, 5:42:00 AM</td>
</tr>
<tr>
<td>catia/</td>
<td></td>
<td>3/6/09, 2:00:00 AM</td>
</tr>
<tr>
<td>ctovideo/</td>
<td></td>
<td>11/9/07, 2:00:00 AM</td>
</tr>
<tr>
<td>device_manager/</td>
<td></td>
<td>9/1/17, 3:00:00 AM</td>
</tr>
<tr>
<td>diag/</td>
<td></td>
<td>11/12/07, 2:00:00 AM</td>
</tr>
<tr>
<td>docs/</td>
<td></td>
<td>1/24/17, 2:00:00 AM</td>
</tr>
<tr>
<td>dpnc/</td>
<td></td>
<td>7/8/09, 3:00:00 AM</td>
</tr>
<tr>
<td>enterprise/</td>
<td></td>
<td>9/15/07, 3:00:00 AM</td>
</tr>
<tr>
<td>essentials/</td>
<td></td>
<td>9/15/07, 3:00:00 AM</td>
</tr>
<tr>
<td>euro/</td>
<td></td>
<td>11/23/08, 2:00:00 AM</td>
</tr>
<tr>
<td>extacell/</td>
<td></td>
<td>1/4/10, 2:00:00 AM</td>
</tr>
<tr>
<td>faxes/</td>
<td></td>
<td>9/15/07, 3:00:00 AM</td>
</tr>
<tr>
<td>futures/mart/</td>
<td></td>
<td>6/27/11, 3:00:00 AM</td>
</tr>
<tr>
<td>graham/</td>
<td></td>
<td>8/31/17, 3:00:00 AM</td>
</tr>
<tr>
<td>gsb/</td>
<td></td>
<td>7/7/14, 3:00:00 AM</td>
</tr>
<tr>
<td>gsy/</td>
<td></td>
<td>3/6/09, 2:00:00 AM</td>
</tr>
<tr>
<td>handheld_computers/</td>
<td></td>
<td>9/15/07, 3:00:00 AM</td>
</tr>
<tr>
<td>hp_StoreOnce/</td>
<td></td>
<td>12/8/14, 2:00:00 AM</td>
</tr>
<tr>
<td>hp_StoreOnceRMC/</td>
<td></td>
<td>5/5/16, 3:00:00 AM</td>
</tr>
<tr>
<td>hp_easytools/</td>
<td></td>
<td>9/30/11, 3:00:00 AM</td>
</tr>
<tr>
<td>hpcast/</td>
<td></td>
<td>3/18/08, 2:00:00 AM</td>
</tr>
</tbody>
</table>
How do you upgrade a printer firmware?! You Print it!
How do you upgrade a printer firmware?

You Print it!
Printing The Firmware

| 1B 25 2D 31 | 32 33 34 35 | 58 40 50 4A | 4C 20 43 4F | .-%-12345X@PJL COMMENT (null).@PJL ENTER LANGUAGE =FWUPDATE..ETHis device does not support FWUPDAT
| 4D 4D 45 4E | 54 20 28 6E | 75 6C 6C 29 | 0A 40 50 4A |
| 4C 20 45 4E | 54 45 52 20 | 4C 41 4E 47 | 55 41 47 45 |
| 3D 46 57 55 | 50 44 41 54 | 45 0A 1B 45 | 54 68 69 73 |
| 20 64 65 76 | 69 63 65 20 | 64 6F 65 73 | 20 6E 6F 74 |
| 20 73 75 70 | 70 6F 72 74 | 20 46 57 55 | 50 44 41 54 |
| 45 21 0D 0A | 1B 2A 72 74 | 31 36 33 38 | 41 32 37 30 |
| 2A 62 31 36 | 35 34 32 59 | 1B 2A 62 2B | 30 59 1B 2A |
| 62 31 36 33 | 38 34 56 53 | 41 32 37 30 | 32 30 32 30 |
| 31 45 43 32 | 45 34 41 30 | 30 30 30 30 | 30 30 32 30 |
| 46 41 35 36 | 43 32 33 42 | 30 45 34 36 | 30 36 41 41 |
| 42 36 38 41 | 33 33 32 34 | 42 37 46 34 | 37 31 37 |
| 42 41 39 33 | 41 45 39 30 | 41 33 35 38 | 46 39 44 37 |
| 30 38 31 31 | 38 46 32 33 | 32 0A 53 41 | 32 37 41 32 |
| 39 43 32 33 | 32 33 30 31 | 43 31 39 46 | 30 41 46 42 |
| 44 41 30 35 | 44 35 37 30 | 35 31 38 36 | 44 41 39 37 |
| 45 39 39 41 | 45 32 41 44 | 36 38 41 46 | 31 43 42 39 |
| 45 46 34 42 | 34 36 41 36 | 44 43 37 43 | 46 41 39 41 |
| 30 35 39 39 | 30 38 39 43 | 30 44 32 33 | 0A 53 41 32 |
| 058808890D23 SA2 |
Printing The Firmware

NULL Decoder

TIFF Decoder

Delta Raw Decoder
When You’re a Hammer...
Sections

<table>
<thead>
<tr>
<th>Loading Address</th>
<th>Section Name</th>
<th>Location in Binary</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x60</td>
<td>romnosi_text</td>
<td>0x7B0</td>
</tr>
<tr>
<td>0x810</td>
<td>romfast_imem1_text</td>
<td>0x310</td>
</tr>
<tr>
<td>0xB20</td>
<td>romfast_imem2_text</td>
<td>0x2BC</td>
</tr>
<tr>
<td>0xDDC</td>
<td>romfast_imem3_text</td>
<td></td>
</tr>
<tr>
<td>0x2009FD8C</td>
<td>crom_load_rodata</td>
<td>0x277F</td>
</tr>
<tr>
<td>0xA007CD60</td>
<td>cromload_data</td>
<td>0x3CFA</td>
</tr>
<tr>
<td>0x210FADCC</td>
<td>cromfixcon_display</td>
<td>0x3B6D86</td>
</tr>
<tr>
<td>0x200A4DCC</td>
<td>cromtext</td>
<td></td>
</tr>
</tbody>
</table>
I Don't Understand…
What is this?!

- Probably a compression algorithm
- A very bad one …
- Some mathematics
Let's Take A Look

recursively
non positive size
variable length

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**

<table>
<thead>
<tr>
<th>recursion</th>
<th>very</th>
<th>non</th>
<th>positive</th>
<th>size</th>
</tr>
</thead>
<tbody>
<tr>
<td>variable-length</td>
<td>JPEG</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<p>| 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 |</p>
<table>
<thead>
<tr>
<th>FF</th>
<th>DF</th>
<th>FF</th>
<th>EF</th>
<th>EF</th>
<th>FF</th>
<th>AE</th>
<th>E0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0E</td>
<td>32</td>
<td>FF</td>
<td>FE</td>
<td>30</td>
<td>F7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F7</td>
<td>AD</td>
<td>33</td>
<td>FF</td>
<td>FF</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
A Pattern?

<table>
<thead>
<tr>
<th>FF</th>
<th>8 Bytes</th>
<th>EF</th>
<th>9 Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>9 Bytes</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>9 Bytes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F7</td>
<td>9 Bytes</td>
<td>FF</td>
<td>8 Bytes</td>
</tr>
<tr>
<td>F7</td>
<td>9 Bytes</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The Missing Link

- Forward / Backward Pointer
- Dictionary
- Sliding Window
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 tile images used at the beginning of each game. It was created in 1992 by Jim Boy.

The data held in the file can be compressed in any one of three ways, uncompressed, LZW and LZX. The compression used is primitive and rather different from later or traditional versions of LZW/LZH. SLIB files were created by the program SOFTLIB.EXE and can be read by any game that uses this format contains various segments of code in common with SOFTLIB.EXE 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 'CMPL' (Compressed file) while that of SLIB files is SLIB. The versatility 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: .CMPL (Compressed), .SLIB (Softdisk Help Library) or the game 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.

### 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.
COMMANDEER KEEN
in "GOODBYE, GALAXY!"
DISTRIBUTED BY APOGEES

Press F1 for Help
<table>
<thead>
<tr>
<th>Window Location</th>
<th>Data Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>2771</td>
<td>3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>A</th>
<th>D</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

AD 33
Mystery Solved

Sliding Window

|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |

Input Text

ABCDABEFG

Output Text
Mystery Solved

Sliding Window

A

Input Text

ABCDA

Output Text

A
Mystery Solved

Sliding Window

A B

Input Text

ABCDA
BEFG

Output Text

A B
Mystery Solved

Sliding Window

A B C

Input Text

A B C

Output Text

A B C
Mystery Solved

Sliding Window

A B C D

Input Text

A B C D

Output Text

A B C D
Mystery Solved

Sliding Window

Input Text

Output Text
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

ABCDABEFG

Output Text

A B C D 00 02 E
Mystery Solved

Sliding Window

A B C D E F

Input Text

ABCDABEFG

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 | 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

ABCDEF

Input Text

ABCDAB

Output Text

EFABCD0002EFG
The Printing Beast

- 64,709 Functions
- Most of the code not parsed by IDA
- Indirect Calls, Dynamic Tables, BootLoader Functions
<table>
<thead>
<tr>
<th>System n’ Stuff</th>
<th>Common Libraries</th>
<th>Tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>mDNSResponder</td>
<td>tPrintFax</td>
</tr>
<tr>
<td></td>
<td>Spidermonkey</td>
<td>tT30</td>
</tr>
<tr>
<td></td>
<td>OpenSSL 1.0.1j (2014)</td>
<td>tFaxLog</td>
</tr>
<tr>
<td></td>
<td>gSOAP 2.7</td>
<td>tModem</td>
</tr>
<tr>
<td></td>
<td>libpng 1.2.29 (2008)</td>
<td>tTB, tHTML, ...</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ThreadX - ARM9/Green Hills</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Making Some Sense

ThreadX - ARM9/Green Hills

Common Libraries
- mDNSResponder
- OpenSSL 1.0.1j
- libpng 1.2.29
- gSOAP 2.7

Tasks
- tPrintFax
- tT30
- FaxLog
- tModem
- tTB, tHTML, ...

Spidermonkey

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

2 Stage Boot Loader
JSOnAPrinter?! 

- 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
if ( v10 != -98 )
{
    sub_20A05253(v9);
    sub_20A05253(dword_A2315C38);
    dword_A2315C38 = -1;
    return v10;
}
if ( (v12 - 301) > 2 && v12 != 307 )
{
    sub_20A05253(v9);
    sub_20A05253(dword_A2315C38);
    result = -1;
    dword_A2315C38 = -1;
    return result;
}
/*(dword_A320E290 + 12)(dword_A320E290, v9, "Location", &arg1, 257);
if ( yl_strstr(&arg1, "http://") )
{
    yl_strncpy(&dst, &arg1, 256);
    sub_20A05253(v9);
} else
{
    v11 = YA_strlen(&dst);
    sub_20B2E484(&dst, &v15, 256 - v11);
}
if (!yl_sub_20A24370(sunk_A320E294, "fakeurl1234.com", "fakeurl1234.com", &a4a ))
{
    if ( a4a )
    {
        yl_sub_20A24030(a4a, v5, v6, v7);
        yl_js_free(a4a);
        a4a = 0;
    }
    sub_20A05253(v9);
    return 0;
}
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

```c
/* snip */
```
The Undebuggable

• 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
Lucky Break

- 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!
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 )
                LOBYTE(c_1) = 0x20;
            *s++ = c_1;
        }
    }
    *s = 0;
    if ( !yb_strncmp(buf, "xml ", 4) )
    {
        v6 = yl_strstr(buf, " encoding=");
        if ( v6 )
            // soap_blank(c)
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 Work?

**PHASE A**
- Caller ID

**PHASE B**
- Sender Caps (DIS)
- Receiver Caps (DTC)

**PHASE C**
- Data Transfer

**PHASE D**
- End of page (EOP)
- Msg Confirm (MCF)
How Does A Fax?
How Does A FAX?

Phase A
Phase B
Phase C
Phase D

TIFF Header
TIFF Body

G.3/G.4

T.30

HDLC

Tunnel
How Does a Fax?
How Does A Fax?

PHASE A
PHASE B
PHASE C
PHASE D

JPEG Header and Body

T.30
Color Extension

HDLC

Tunnel
- 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.
<table>
<thead>
<tr>
<th>FF</th>
<th>D8</th>
<th>FF</th>
<th>E0</th>
<th>00</th>
<th>10</th>
<th>4A</th>
<th>46</th>
<th>49</th>
<th>46</th>
<th>00</th>
<th>01</th>
<th>02</th>
<th>00</th>
<th>00</th>
<th>64</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>64</td>
<td>00</td>
<td>00</td>
<td>FF</td>
<td>C4</td>
<td>0A</td>
<td>02</td>
<td>34</td>
<td>D3</td>
<td>2A</td>
<td>78</td>
<td>80</td>
<td>42</td>
<td>6D</td>
<td>2B</td>
</tr>
<tr>
<td>FF</td>
<td>DA</td>
<td>12</td>
<td>28</td>
<td>2A</td>
<td>6F</td>
<td>2B</td>
<td>81</td>
<td>6A</td>
<td>16</td>
<td>0F</td>
<td>C8</td>
<td>9A</td>
<td>13</td>
<td>FF</td>
<td>D9</td>
</tr>
</tbody>
</table>

**SOI - Start Of Image**
**APP0 - Application Specific**
  Size
  Data
**DHT - Define Huffman Table**
  Size
  Data
**SOS - Start Of Scan**
  Data
**EOI - End Of Image**
• Define Huffman Table

• Defines 4X4 comparison matrix for the JPEG Image
• 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
<table>
<thead>
<tr>
<th>FF</th>
<th>C4</th>
<th>20</th>
<th>00</th>
<th>01</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>02</th>
<th>00</th>
<th>01</th>
<th>02</th>
<th>00</th>
<th>00</th>
<th>00</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>00</td>
<td>00</td>
<td>FF</td>
<td>FF</td>
<td>C4</td>
<td>0A</td>
<td>02</td>
<td>34</td>
<td>D3</td>
<td>2A</td>
<td>78</td>
<td>80</td>
<td>42</td>
<td>6D</td>
<td>2B</td>
</tr>
</tbody>
</table>

---

**Stack**

---

256
Can You Spot It?

```c
v2 = EI_jpg_stream_read_byte_from_file();
v3 = v2 >> 4;
v4 = v2 & 0xFF;
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;
vl_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

Overflow!!

Stack

256

4000
Exploit Chain

- 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
Little Help From My Friends

Lior Oppenheim
@oppenheim1

Yannay Livneh
@Yannayli

Tamir Bahar
@tmr232

Yoav Alon
@yoavalon
fin.

ynvb
Eyalltkin

Check Point
SOFTWARE TECHNOLOGIES LTD.