

Martin Gallo

 **TROOPERS**

March 2016

Deep-dive into SAP archive file formats

Agenda

Introduction

Motivation

SAP compression algorithms

Archive file programs

SAP archive file formats

- CAR
- SAR v2.00
- SAR v2.01

Relative/absolute paths

TCPDB.DAT case

Archive file signatures

Attack surface

Attack vectors

Defense

Conclusions

Introduction

- SAP
- SAP systems
- SAP security
- Complexity
- Archive files
 - Software packaging
 - Software distribution
 - Transport files



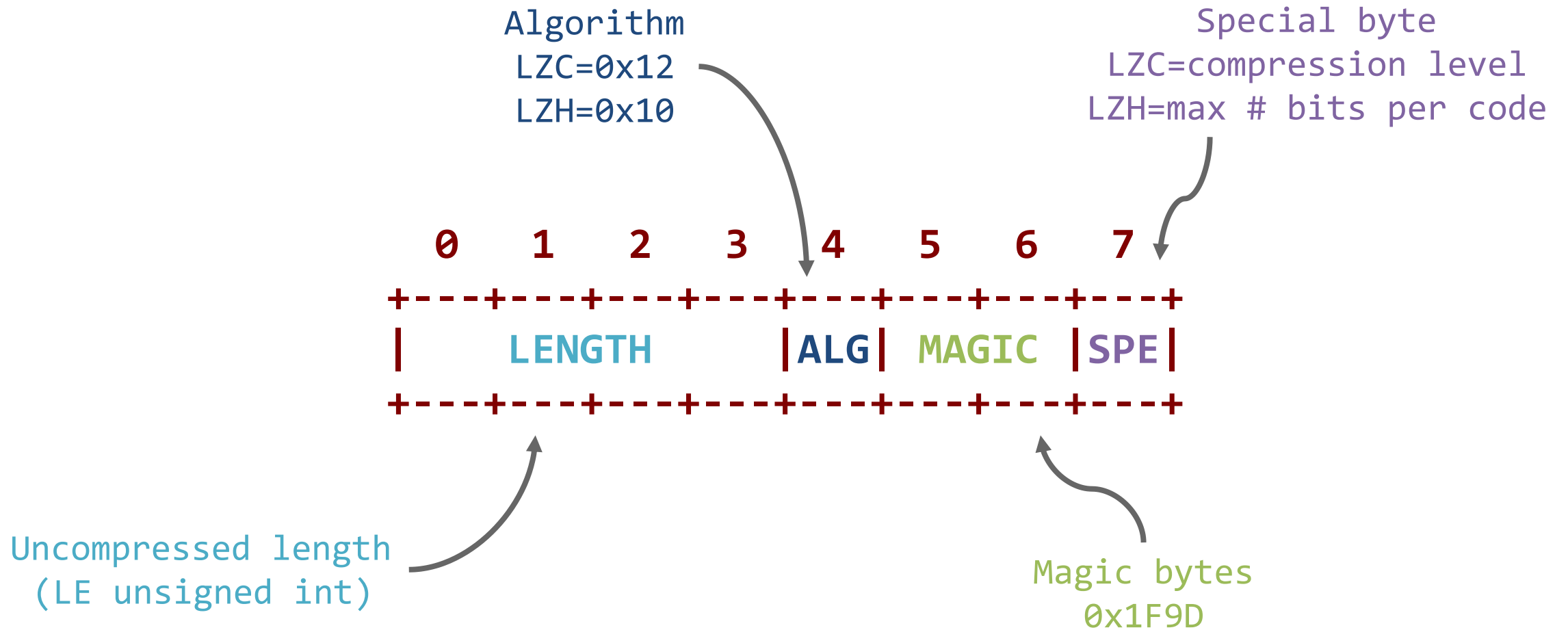
Motivation

- File formats are not known
- Lack of public documentation
- Lack of practical known attacks
- Went deep into the compression mechanisms
- A different attack vector
- Targets sysadmins, operators and **BASIS** admins
 - High privileged users

SAP Compression algorithms

- Based on Lempel-Ziv algorithm
- Adaptive dictionary compression
- Custom implementation
- Two variants
 - LZH (Lempel-Ziv-Huffman)
 - LZC (Lempel-Ziv-Welch-Thomas)

Compression header



Archive files programs

- SAPCAR program
 - Command-line
 - Available on multiple platforms
 - Allows listing, adding, extracting, verifying archive files
 - Works with CAR, SAR v2.00 and v2.01 files
 - Latest version release 721
 - > 16 March 2015

SAPCAR program

```
$ ./SAPCAR
```

```
usage:
```

```
create a new archive:
```

```
SAPCAR -c[vir][f archive] [-P]
      [-A filename] [-T filename]
      [-p value] [-V] file1 file2
```

```
list the contents of an archive
```

```
SAPCAR -t[vs][f archive] [file1
```

```
extract files from an archive:
```

```
SAPCAR -x[v][f archive] [-R dir
      [-V] [file1 file2....]
```

```
verify the archive:
```

```
SAPCAR -d[v][f archive] [-V] [f
[...]
```

```
[...]
```

```
append files to an archive:
```

```
SAPCAR -a[v][f archive] file1 [file2....]
```

```
merge two archives:
```

```
SAPCAR -m[v]f "source target"
```

```
check availability of files to be processed:
```

```
SAPCAR -l [-A filename][-X filename] [file1 file2...]
```

```
sign archive:
```

```
SAPCAR -S[v]f MY.SAR [-key keyname] [-H file hash]
```

```
verify the content of signed manifest:
```

```
SAPCAR -M[v][f manifest file] [-manifest file]
```

```
[...]
```

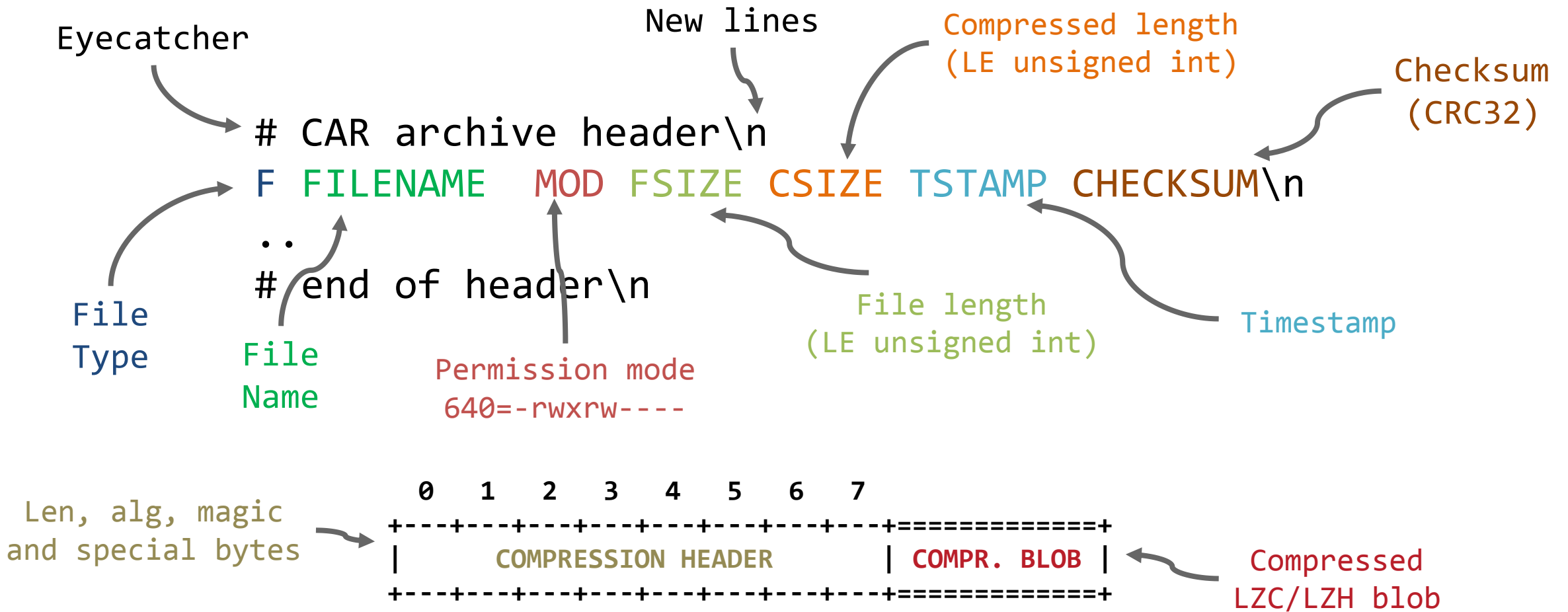

SAP archive file formats

- Software packaging/distribution
 - CAR
 - SAR v2.00
 - SAR v2.01
- Transport files
 - Transport files

CAR archive file format

- Old (first?) version of the archive file
- Text based archive header
- Blob content
- Still supported on SAPCAR for extracting
- Not supported for creating new archives

CAR archive header



CAR example

```
$ ./SAPCAR -xvf carcar_test_string.sar
processing archive carcar_test_string.sar...
x test_string.txt
```

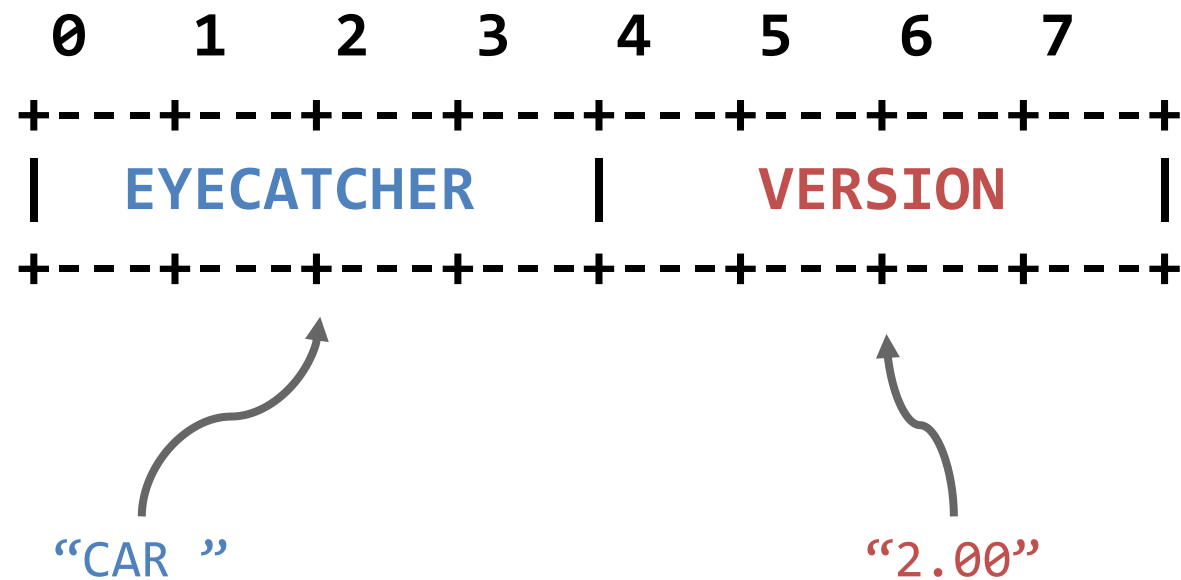
```
$ xxd carcar_test_string.sar
0000000: 2320 4341 5220 6172 6368 6976 6520 6865 # CAR archive he
0000010: 6164 6572 0a46 2074 6573 745f 7374 7269 ader.F test_stri
0000020: 6e67 2e74 7874 2020 2020 2020 2020 2020 ng.txt
0000030: 2034 3434 2020 2020 2020 2020 3433 2020 444 43
0000040: 2020 2020 2020 3533 2031 3434 3930 3130 53 1449010
0000050: 3132 3820 3331 3136 3736 3331 3434 0a23 128 3116763144.#
0000060: 2065 6e64 206f 6620 6865 6164 6572 0a2b end of header.+
0000070: 0000 0012 1f9d 027b 2119 a90a 85a5 99c9 .....{!.....
0000080: d90a 4945 f9e5 790a 69f9 150a 59a5 b905 ..IE..y.i...Y...
0000090: c50a f965 a945 0a25 40e9 9cc4 aa4a 8594 ...e.E.%@....J..
00000a0: fc74 0000 .t..
```

```
File type=file, Filename=test_string.txt, Perm mode=444, File length=43, Compressed
Length=53, Timestamp=01 Dec 2015 19:48, Checksum=0xb9c60808, Uncompressed Length=43,
Algorithm=LZC, Special byte=02
```

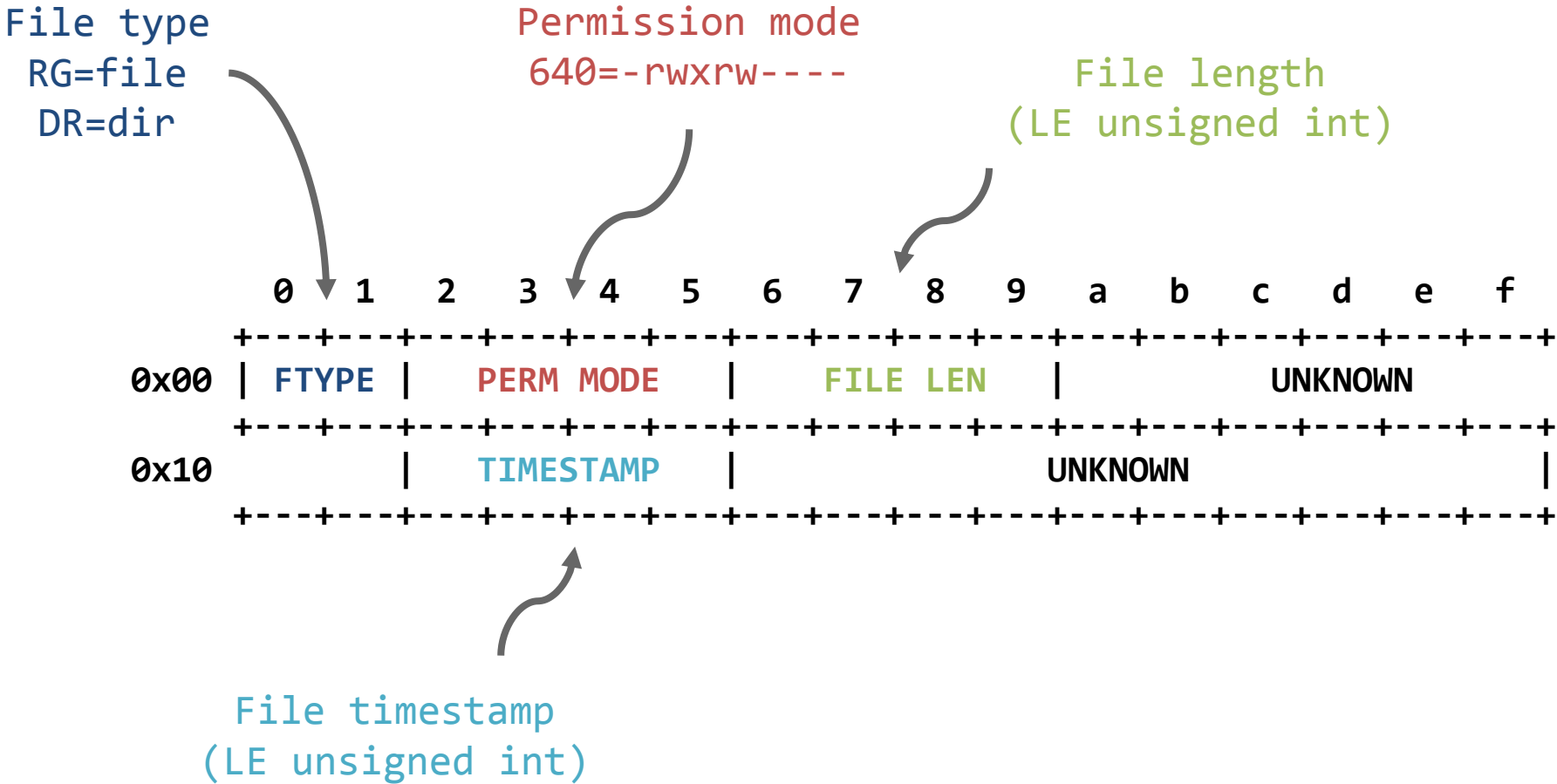
SAR v2.00 archive file format

- New version of the archive file (R/3 > 4.70)
- Binary based archive file header
- Still supported on SAPCAR for extracting
- Not supported for creating new archives

SAR v2.00 archive header

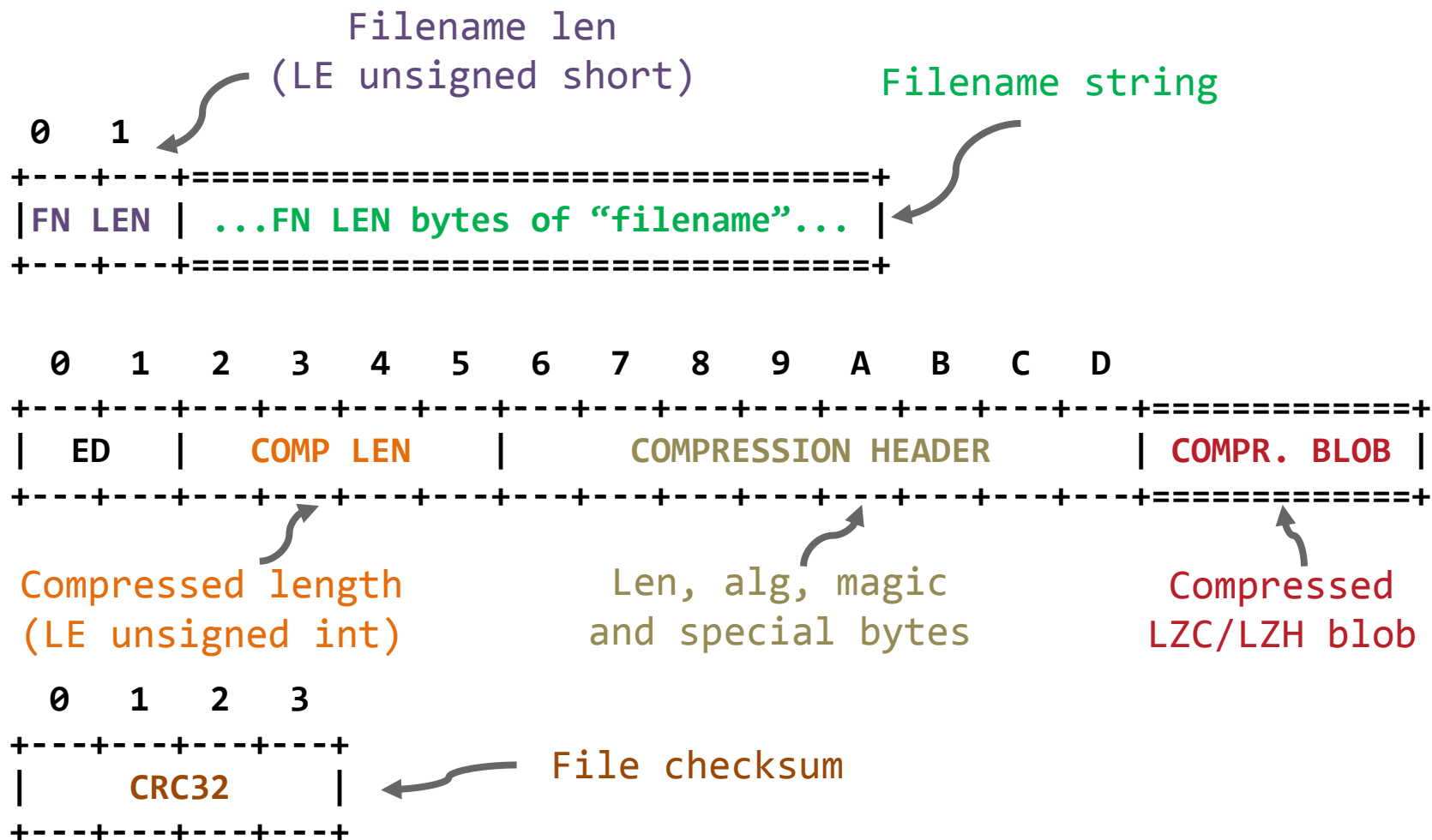


SAR v2.00 archive file header



SAR v2.00 archive file header

If it's a regular file, and file length > 0



SAR v2.00 example

```
$ ./SAPCAR -xvf car200_test_string.sar
SAPCAR: processing archive car200_test_string.sar (version 2.00)
x test_string.txt
SAPCAR: 1 file(s) extracted
```

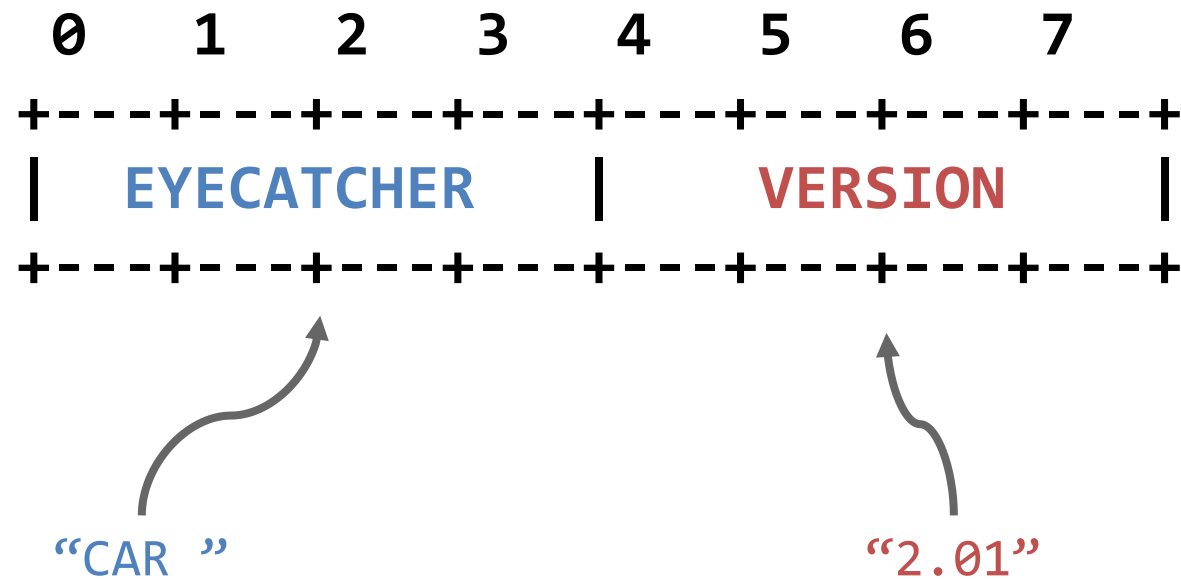
```
$ xxd car200_test_string.sar
00000000: 4341 5220 322e 3030 5247 b481 0000 2b00  CAR 2.00RG....+.
00000010: 0000 0000 0000 0000 0000 d023 5e56 0000  .....#^V..
00000020: 0000 0000 0000 0000 0f00 7465 7374 5f73  .....test_s
00000030: 7472 696e 672e 7478 7445 4435 0000 002b  tring.txtED5...+
00000040: 0000 0012 1f9d 027b 2119 a90a 85a5 99c9  .....{!.....
00000050: d90a 4945 f9e5 790a 69f9 150a 59a5 b905  ..IE..y.i...Y...
00000060: c50a f965 a945 0a25 40e9 9cc4 aa4a 8594  ...e.E.%@....J..
00000070: fc74 0000 0808 c6b9  .t.....
```

```
Version=2.00, File type=file, Perm mode=664, File length=43, Timestamp=01 Dec
2015 19:48, Filename length=15, Filename=test_string.txt, Compressed Length=53,
Uncompressed Length=43, Algorithm=LZC, Special byte=02, Checksum=-1178204152
```

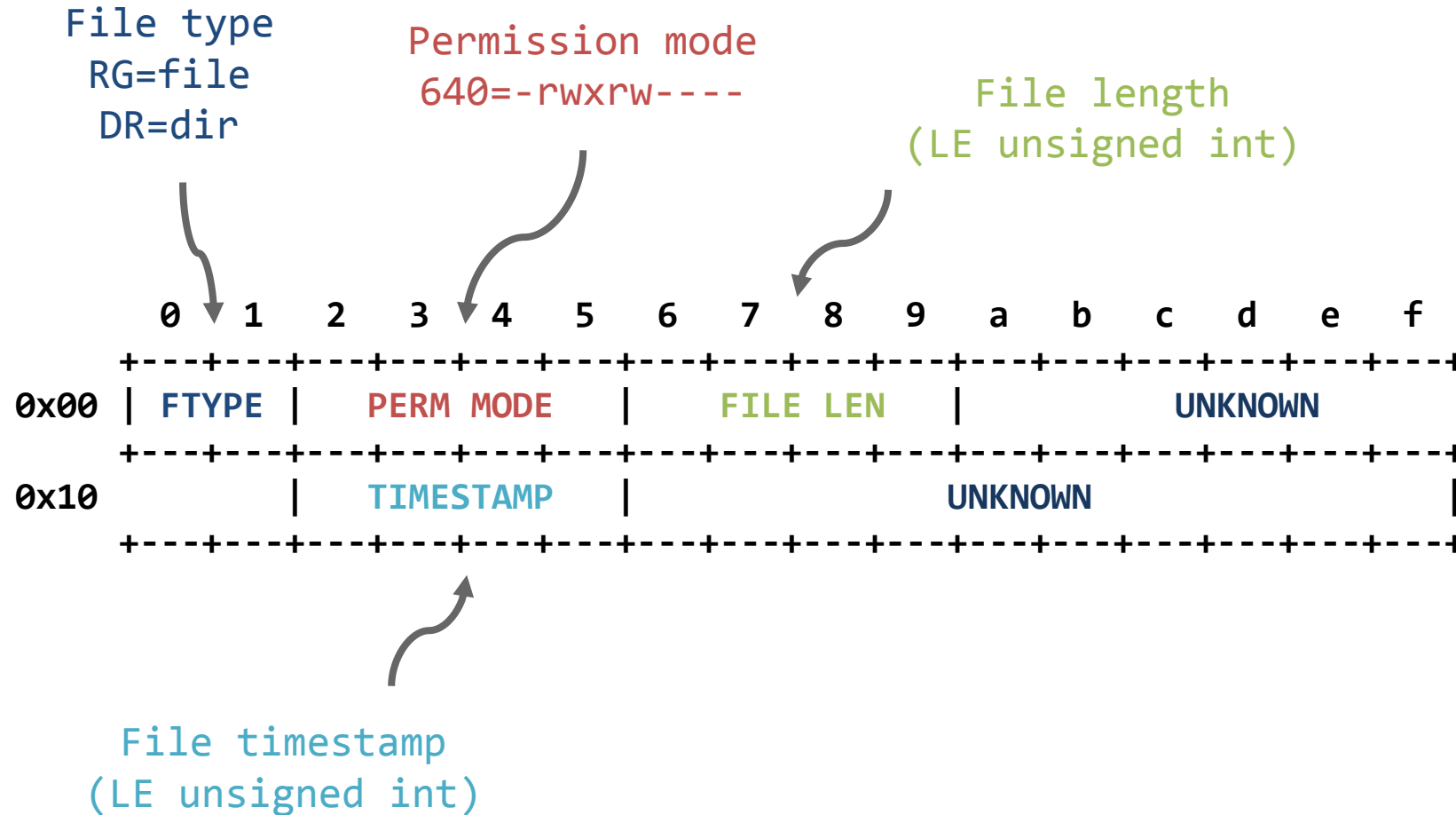
SAR v2.01 archive file format

- Newest version of the archive file
- Same structure as v2.00, except:
 - Handling of filename length
 - Filename is null-terminated
- Default version on SAPCAR

SAR v2.01 archive header



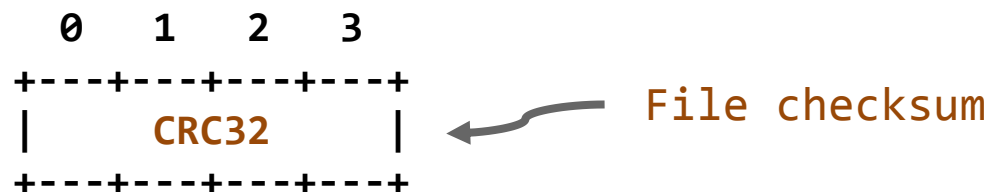
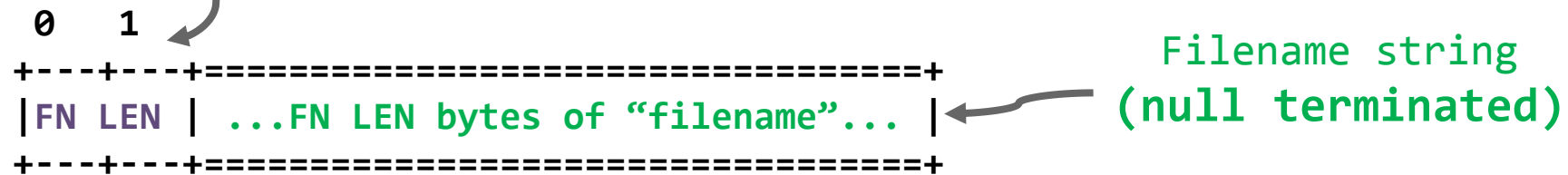
SAR v2.01 archive file header



SAR v2.01 archive file header

If it's a regular file, and file length > 0

Filename len
(LE unsigned short)



SAR v2.01 example

```
$ ./SAPCAR -xvf car201_test_string.sar
SAPCAR: processing archive car201_test_string.sar (version 2.01)
x test_string.txt
SAPCAR: 1 file(s) extracted
```

```
$ xxd car201_test_string.sar
00000000: 4341 5220 322e 3031 5247 b481 0000 2b00  CAR 2.01RG....+.
00000010: 0000 0000 0000 0000 0000 d023 5e56 0000  .....#^V..
00000020: 0000 0000 0000 0000 1000 7465 7374 5f73  .....test_s
00000030: 7472 696e 672e 7478 7400 4544 3500 0000  tring.txt.ED5...
00000040: 2b00 0000 121f 9d02 7b21 19a9 0a85 a599  +.....{!.....
00000050: c9d9 0a49 45f9 e579 0a69 f915 0a59 a5b9  ...IE..y.i...Y..
00000060: 05c5 0af9 65a9 450a 2540 e99c c4aa 4a85  ....e.E.%@....J.
00000070: 94fc 7400 0008 08c6 b9          ..t.....
```

```
Version=2.00, File type=file, Perm mode=664, File length=43, Timestamp=01 Dec
2015 19:48, Filename length=16, Filename=test_string.txt, Compressed Length=53,
Uncompressed Length=43, Algorithm=LZC, Special byte=02, Checksum=-1178204152
```

SAPCAR Relative/absolute paths

- Handling of absolute/relative paths
 - “/usr/var/some_file_name”
 - “../../some_file_name”

```
$ ./SAPCAR
```

```
usage:
```

```
[..]
```

```
using absolute pathnames:
```

```
If you create an archive with absolute pathnames the files will be  
extracted with exactly these pathnames! SAPCAR does not cut the first  
slash like the UNIX tool tar.
```

```
[..]
```

SAPCAR TCPDB.DAT case

- Ran SAPCAR with strace and found:

```
[..]  
open("TCPDB.DAT", O_RDONLY)          = -1 ENOENT (No such file or directory)  
[..]
```

```
$ ./SAPCAR  
TCPDB.DAT line 1: format error: "____".  
TCPDB.DAT line 2: format error: " | | _/".  
TCPDB.DAT line 3: format error: " | | \".  
TCPDB.DAT line 4: format error: "      ".
```


SAPCAR archive files signature

- Feature introduced some time ago
 - Release 720 patch level 2
 - SAP Note 1598550 - SAPCAR: Signed archive (2011)
 - SAP Note 1634894 - SAPCAR: Signed Archive (2012)
- Feature documentation
 - SAP Note 2178665 - Signature validation of archives with SAPCAR (2016)
- Cryptographic primitives provided by crypto library
 - SAPCRYPTOLIB
 - CommonCryptolib

SAPCAR archive files signature

- Manifest files
 - **MANIFEST.SMF** inside the archive file
 - Detached manifest file
 - Contains hashes and signature
- Defaults algorithms
 - SHA256 for hashes
 - PKCS7 for signature
- Embedded public keys in crypto library
- Support for CRL and certificate revocations

SAPCAR signature example

```
$ cat SIGNATURE.SMF
SAP-MANIFEST
Version: 1.0
Hash: SHA256
Signature: PKCS7-TSTAMP
Body: Digest | Name-Length | Name

f212e04bca96925a5cd424d9f8b2733533e1d800b38021bf332ea092d0be4c6f 000d Changelog.txt
9a07479720f9c40bd672e16cded8af322afd84eda40d18279d9d108bc0ed6163 0009 LEGAL.TXT
ae113a023d4b93ca1dfe1c9e0d1bdf4d37ac0a6cbc3fd6077f8cd36756dcdb08 000b LICENSE.TXT
[.]

-----BEGIN SIGNATURE-----
MIINTgYJKoZIhvcNAQcCoIINPzCCDTsCAQExDzANBgIghkgBZQMEAgEFADALBgkq
hkiG9w0BBwGgggQDMIID/zCCAuegAwIBAgIBCjANBgkqhkiG9w0BAQsFADBMMQsw
CQYDVQQGEwJERTEfMBOGA1UEChMWU0FQIFRydXN0IENvbW11bm10eSBJSSTEcMBoG
[.]
-----END SIGNATURE-----
```

SAPCAR signature example

```
$ ./SAPCAR -L ./libsapcrypto.so -tVvf SAPCAR_signed.SAR
SAPCAR: processing archive SAPCAR_signed.SAR (version 2.00)
srw-r--r--      46977      11 May 2015 13:43 Changelog.txt
srw-r--r--       891       03 Feb 2012 16:53 LEGAL.TXT
srw-r--r--      2708       03 Feb 2012 16:52 LICENSE.TXT
[..]
```

```
SAPCAR: Signature Subject >CN=CodeSigner006, OU=Code Signing, O=SAP Trust Community
II, C=DE<
```

```
SAPCAR: Signature Issuer  >CN=SAP Code Signing CA, O=SAP Trust Community II, C=DE<
```

```
SAPCAR: Signature Type   >SAP software<
```

SAPCAR signature example

```
$ ./SAPCAR -L ./libsapcrypto.so -tVvf SAPCAR_signed_tampered.SAR
SAPCAR: processing archive SAPCAR_signed_tampered.SAR (version 2.00)
srw-r--r--      46977      11 May 2015 13:43 Changelog.txt
srw-r--r--       891       03 Feb 2012 16:53 LEGAL.TXT
srw-r--r--      2708       03 Feb 2012 16:52 LICENSE.TXT
[.]
-rw-rw-r--       43       01 Dec 2015 19:48 test_string.txt
SAPCAR: at least one file was not signed (error 58).
Detail: File >test_string.txt< was not found in manifest
```

```
$ ./SAPCAR -L ./libsapcrypto.so -tVvf SAPCAR_invalid_signature.SAR
SAPCAR: processing archive SAPCAR_invalid_signature.SAR (version 2.01)
SAPCAR: SAPCAR_invalid_signature.SAR is not digitally signed (error 59). No such file or directory
```

Attack surface

- Compression algorithms
- Archive file handling
- Signature validation
- Download process

Archive file handling

- Fuzzing archive file parsing
 - File name handling
 - Metadata, archive header fields
- Old formats of special interest

```
$ ./SAPCAR -xvf SAPCAR_crash.SAR
SAPCAR: processing archive SAPCAR_crash.SAR (version 2.01)
x input-dir/in#t
Segmentation fault
```

Compression algorithms

- Complex piece of code
 - C/C++, 9 files, 2963 lines of code, 443 McCabe CC
- Source code available
 - Part of old MaxDB, open source version ([LZC](#)/[LZH](#))
- Very extended on SAP products and components
 - SAP Content server
 - ABAP code stored in DB
 - Diag and RFC protocols
 - HANA, MaxDB database engines
 - SAPCAR, SAP GUI, r3trans, r3load, Jco, RFC SDK

Compression algorithms - CVE-2015-2278

- LZH decompression out-of-bounds read
- Building Huffman tree requires doing lookups of previous values
- Indexing an uninitialized array
- Denial of service impact

```
[..]
int CsObjectInt::BuildHufTree (
    unsigned * b, /* code lengths in bits (all assumed <= BMAX) */
    unsigned  n, /* number of codes (assumed <= N_MAX) */
    unsigned  s, /* number of simple-valued codes (0..s-1) */
    int       * d, /* list of base values for non-simple codes */
    int       * e, /* list of extra bits for non-simple codes */
    HUFTREE **t, /* result: starting table */
    int       * m) /* maximum lookup bits, returns actual */

[..]
    if (p >= v + n)
    {
        r.e = INVALIDCODE; /* out of values--invalid code */
    }
    else if (*p < s)
    {
        /* 256 is end-of-block code */
        r.e = (unsigned char)(*p < 256 ? LITCODE : EOBCODE);
        r.v.n = (unsigned short) *p; /* simple code is just the value*/
        p++;
    }
    else
    {
        r.e = (unsigned char) e[*p - s]; /*non-simple,look up in lists*/
        r.v.n = (unsigned short) d[*p - s];
        p++;
    }
[..]
```

Compression algorithms - CVE-2015-2282

```
[..]
int CsObjectInt::CsDecomprLZC (SAP_BYTE * inbuf,
                               SAP_INT   inlen,
                               SAP_BYTE * outbuf,
                               SAP_INT   outlen,
                               SAP_INT   option,
                               SAP_INT *  bytes_read,
                               SAP_INT *  bytes_written)

[..]
/* Generate output characters in reverse order ...*/
while (code >= 256)
{
    *stackp++ = TAB_SUFFIXOF(code);
    OVERFLOW_CHECK
    code = TAB_PREFIXOF(code);
}
[..]
```

- LZC decompression stack-based overflow
- OVERFLOW_CHECK seems to be insufficient
- OVERFLOW_CHECK not enabled at compile time!
- Remote code execution impact is likely

Compression algorithms - CVE-2015-2278/2282

- High profile vulnerabilities
- Affected lot of components
- Require urgent patching
- Patches released on May 2015:
 - SAP Note 2125316 - Potential termination of running processes in SAPCAR
 - SAP Note 2121661 - Potential remote termination of running processes in ABAP & Java Server
 - SAP Note 2127995 - Potential remote termination of running processes in Content Server
 - SAP Note 2124806 - Potential remote termination of running processes in SAP GUI

Attack vectors

- Drive-by-download
- Signature validation
- Automated processing of files
- Download process
 - SAP Download Manager
- User's trust

User's trust

- User trust that file is authentic
- Signature validation is not enforced by default
- Absolute/relative paths are supported by default
- Installation/patch procedures recommends extracting SAR files directly on SAP's directory
- What could go wrong?

User's trust

SAP Note

Statistic 2009 | Printer-Friendly Version | PDF Version | Add to favorites | Subscribe | Quick link | Open SAPNote

6. Unpack the new kernel with the following commands:

```
<newkernel>/SAPCAR -xvf <newkernel>/SAPEXE.SAR
```

```
<newkernel>/SAPCAR -xvf <newkernel>/SAPEXEDB.SAR
```

7. Oracle only:

Unpack the DBATools with the following command:

```
<newkernel>/SAPCAR -xvf <newkernel>/DBATOOLS.SAR
```

Also unpack the Oracle instant client to the directory as explained in the note [819829](#).

8. If you use IGS, you must unpack the IGS archive using the following command:

```
<newkernel>/SAPCAR -xvf <newkernel>/igsexse.sar
```

User's trust

- Demo
 - Using pysap API for handling SAR files
 - Open an existing SAR file
 - Add a new file using an absolute path
- Let the user extract it!

Download process

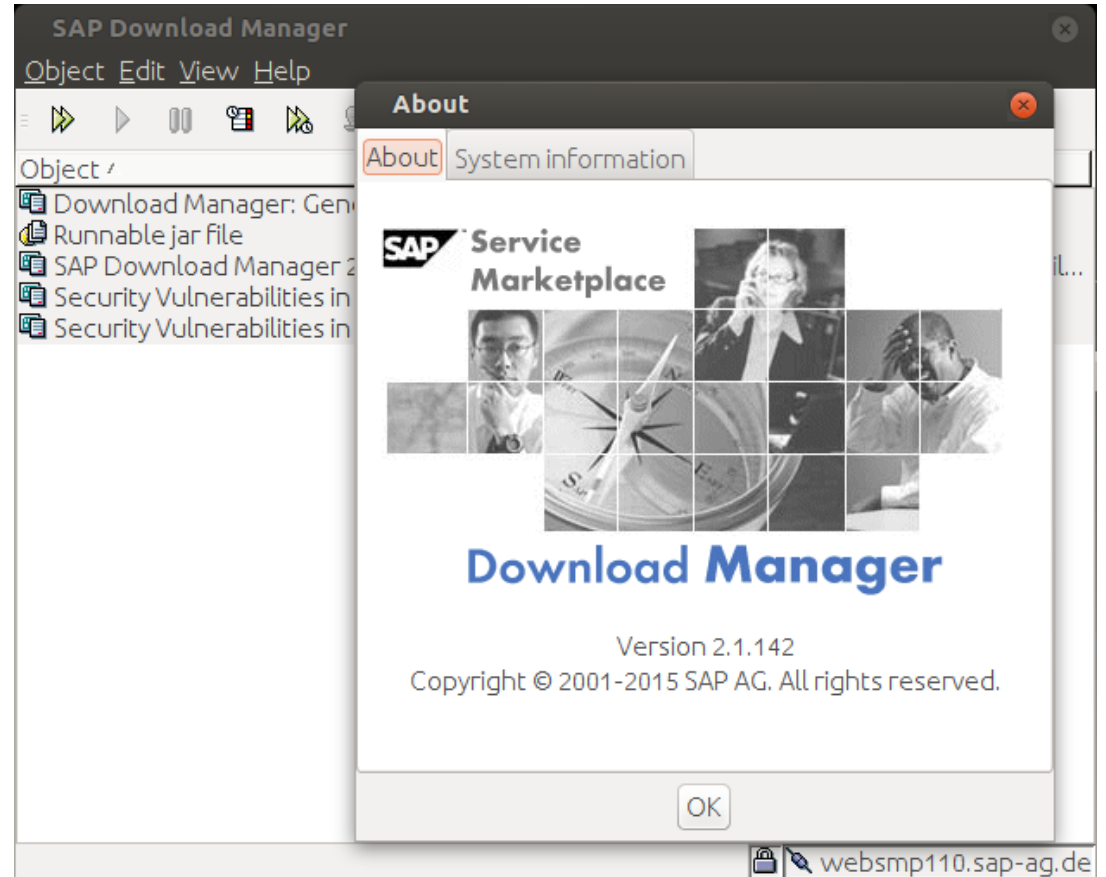
- Downloads from untrusted sources
- Unencrypted channels
- Share folders
- ..
- SAP Download Manager

Unencrypted channels

- User downloading SAP notes through HTTP
- MitM via SSLStrip
- Demo
 - Using mitmproxy to perform MitM + SSL Strip
 - Locate SAR file being downloaded
 - Using pysap to infect the SAR file

SAP Download Manager

- Small Java Application
- Allows downloading notes from their basket
- Latest version is 2.1.143 (March 2016)

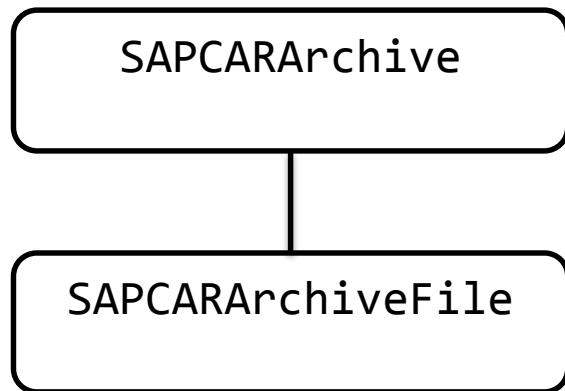


SAP Download Manager

- Version < 2.1.142 (Oct 2015) vulnerable to MitM
 - SAP Note 2233617 - Security Vulnerabilities in SAP Download Manager
 - SAP Note 2235412 - Security Vulnerabilities in SAP Download Manager
- Attack idea?
 - Using mitmproxy to perform MitM
 - Locate SAR file being downloaded inside a zip file
 - Using pysap to infect the SAR file

pysap's SAPCAR API

- Python library for SAP's network protocols
- Implemented SAR v2.00/v2.01 file formats
- High level abstraction of archive/files



<https://github.com/CoreSecurity/pysap>

```
In [2]: from pysap.SAPCAR import *
In [3]: ar = SAPCARArchive("car201_test_string.sar", "r")
In [4]: ar.files_names
Out[4]: ['test_string.txt']
In [5]: f = ar.files["test_string.txt"]
In [6]: f.size, f.permissions, f.timestamp
Out[6]: (43, '-rw-rw-r--', '01 Dec 2015 19:48')
In [7]: f.open().read()
Out[7]: 'The quick brown fox jumps over the lazy dog'
```

Defense

- Ensure running latest program versions
 - SAPCAR
 - CommonCryptoLib
 - SAP Download Manager
 - Etc.
- Download software packages from trusted sources
 - Ensure links are HTTPS
 - Protect shared folders, repositories

Defense

- Extract disabling relative paths
 - Create a new folder
 - Extract there using -flat option
- Validate archive signatures
 - Before extracting!
- Enable CLR checks
 - Download CRL list
 - Validate signature using -crl option

Conclusions

- BASIS life is hard
- New attack vectors unveiled
- More knowledge about archive files
- Plenty of further work
 - Fuzzing?
 - Signatures?

Q & A

Martin Gallo
@martingalloar
github.com/martingalloar
mgallo at coresecurity.com

Thank you.

Thanks to Troopers crew, Joris, Dana & Euge!