Smart Unpacking

Understanding Files Through Patterns

Benjamin Long, AAFS 2009

NIST United States Department of Commerce
National Institute of Standards and Technology

Disclaimer

Disclaimer
Trade names and company products are mentioned in the text or identified. In no case does such identification imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the products are necessarily the best available for the purpose.

Statement of Disclosure
This research was funded by the National Institute of Standards and Technology Office of Law Enforcement Standards, the Department of Justice National Institute of Justice, the Federal Bureau of Investigation and the National Archives and Records Administration.

Outline

- NSRL Project Description
- Smart Unpacking Description
- Smart Unpacking use
- Smart Unpacking overview

Introduction to the NSRL

- Is a computer forensics project at NIST
- Is composed of 3 things:
  1. **library:** physical collection of software media and packages
  2. **database:** logical collection of collected file information
     - file name, size, type, package, MD5, SHA1, etc.
  3. **RDS:** (Reference Data Set) and other products
     - Produced quarterly
     - Metadata db – close to 75M files; publicly available data + db schema

NSRL Use (Data Reduction)

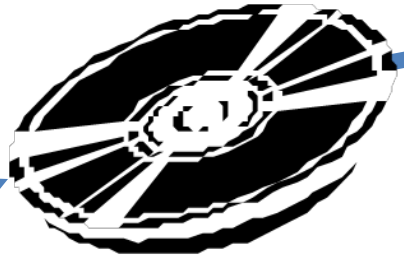
Disk Drive

FILES

Analysis Program

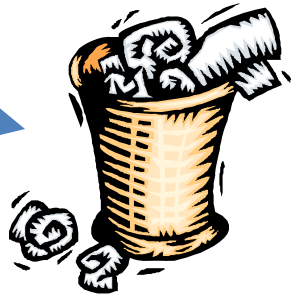
UNKNOWN FILES

RDS



File info

KNOWN FILES



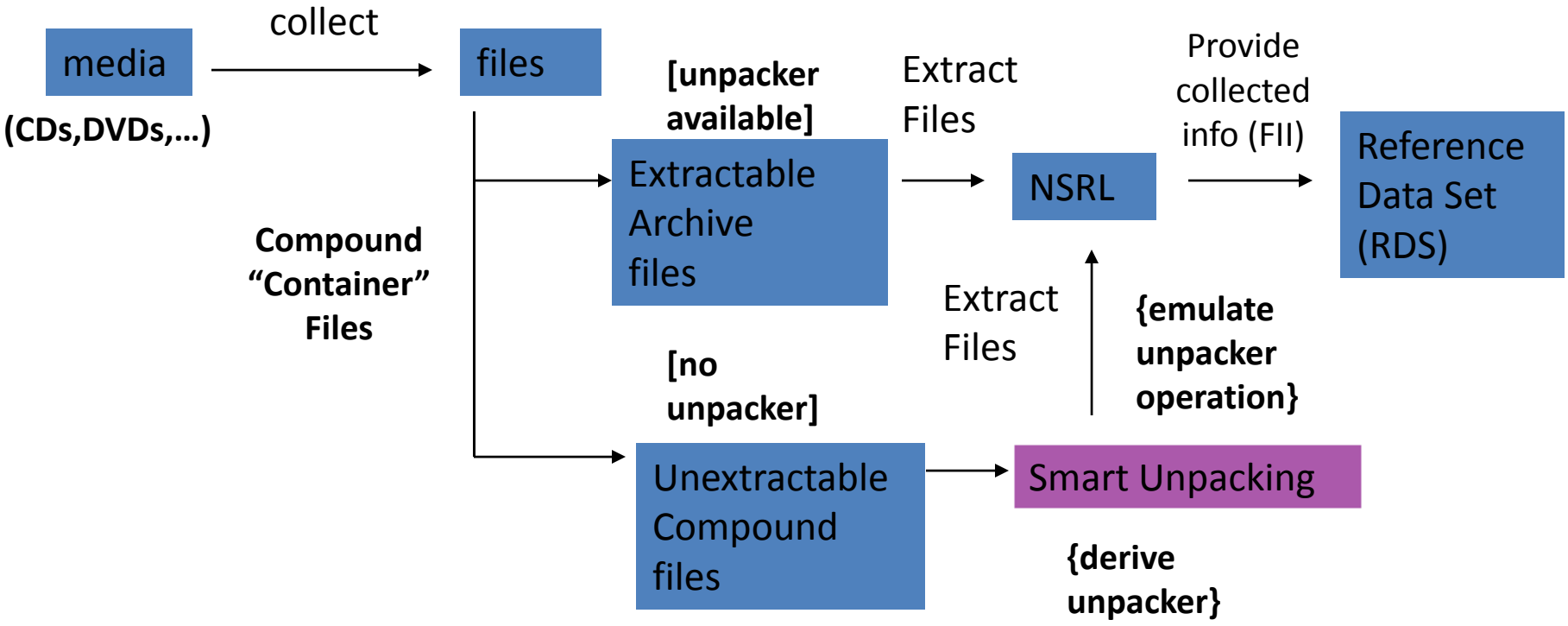
NSRL

- Unpack archives
- Provide File info

NSRL Use

- Application of NSRL unique identification
  - Law Enforcement
    - Automated elimination of “known” app files from an investigation
    - Positive ID of “interesting” files
    - Forensic tools use various metadata
    - NIST/NSRL – provides – unbiased, court-admissible data to NIJ, FBI, DoD
  - NARA Presidential search
  - Voting – baseline maintenance; auditing
  - Ongoing research – online archive of data, file-type identification
  - Used by all major tools

NSRL File Extraction



The Need

NSRL wants more files …

- Two primary ways to get more
  1. Get more media (containing files)
  2. Unpack more compound files (i.e., archives)
- Fundamental challenge
  - get files out of archives
- Why is it hard?
  - different versions, different formats
  - corruption
  - unpacker not always available or current

This need will always exist

Smart Unpacking Approach

Addresses NSRL Need by creating a *process, tools, and techniques* for …

1. Getting files out
2. Doing this in a general, reliable, testable manner
3. Producing unpacker programs and measures of unpacking results

Smart Unpacking - How it works

3 major steps

Representation → Inference → Unpacking

1. **Representation**
   - choose primitives
   - encode data to differentiate key patterns
2. **Inference**
   - perform pattern-theoretic and other analyses on representations
   - locate files and meta-data
   - derive measures
   - derive + test unpacker
3. **Unpacking**
   - extract files
   - measure results

Example Smart Unpacking Experiment

Unextractable Archive

acpid*.deb

Archive NSRL cannot unpack:
total files extracted = 0

Chaos

```
213c617263683e0a6465626961 6e2d62
696e617279202020313039393333 03833
383720203020202020203020202020 20
313030363434202034202020202020 20
2020600a322e300a636f6e74726f6c2e
7461722e677a2020313039393333 03833
383720203020202020203020202020 20
313030363434202034313832202020 20
2020600a1f8b0800000000000003ed5b
ff731cc595f7af9abfe2c54e619b9256
df25220e5f8ced234ea0f0c59854ee2e
45f5eef4eeb677667a6ea667e555f1c7
d830fbcb9914a94b38305c4c52a46c13
```

```
dmaaaamlaaaaaadaaaaaasssn
nnnnsssssssdlbbbzzzzzzbbkf
```

```
dmamladasnsnsnsnsnsnsdl
adbabdbabkbdbdbabadba
```

Representation & transformation

```
dma+mla+da+s+n+s+ns+ns+n
pb+kbabab+alb+kbadnb+dab
```

```
a = alpha = upper or lower
b = binary
d = delimiters = punctuati
e = email = at symbol = @
f = 0xff = all one bits
k = brackets, parentheses
l = newline
m = markup = html or xml =
n = numeric = 0..9
p = path = slash (fwd or b
s = space
t = dot = period
w = white space = tab
z = 0x00 = the zero byte =
```

Order

```
0x00000000      dm
0x00000002      aml
0x00000008      ad
0x0000000f      asnsnsnsnsnsdlntnl
0x00000048      at
0x00000050      at
0x00000054      asnsnsnsnsnsdlbzbkf
0x00000091      ab
0x0000009b      ab
0x0000009f      abdbdbd
0x000000a8      ab
0x000000ad      abt
0x000000b0      ab
0x000000b6      ab
0x000000bb      ab
```

```
Files Unpacked

File: ../../../data/acpid_1.0.4-1_i386.deb
- file header (bytes read = 60) relative t
-- saving file to filename=[debian-binary]
```

SU process derives unpacker + extracts files:
total files extracted = 89

An Example Usage of Smart Unpacking

1. File ACPID*.deb was **1** archive in NSRL
   - Applied SU process, tools, techniques
2. Now it is **89** files
   - Derived unpacker program, executed it, extracted files
3. Verification
   - Found alternate unpacker program and verified SU result
   - Original archive had corrupted data, causing format variances from standard format

Technical /Scientific Basis and Methods

Methods used to Discern Structure and Metadata
-Modeling
-Uniqueness features in – syntax, content, abstract pattern representations

Pattern Theory

$$G = \{g_0, g_1, ..., g_i, g_i^0\} = generator\_space$$
$$S : G \longleftrightarrow G, s \in S = similarity\_group$$
$$G = \bigcup_{\alpha \in A} G^{\alpha} = partition\_of\_generators$$
$$b_1, b_2, ..., b_{\omega} = bond\_values$$
$$B_s(g) = \{b_j; j = 1,2,3,..., w(g)\} = bond\_structures$$
$$B_v(g) = \{\beta j = 1,2,..., w(g)\}$$
...

Formal Language Theory

Context-free grammars

$$G = (V, \Sigma, R, S) = grammar$$
$$V = nonter \min als$$
$$\Sigma = ter \min als$$
$$S = start\_rule$$
$$R = rules$$
...

Mathematical Modeling based on:

- Information Theory
- Statistics
- Probability
- Universal Algebra
- Topology
- Graph Theory, and more …

Parser Theory

- Variable lookahead mechanisms
- Multi-channel token processors
- Augmentation with syntactic and semantic predicates for context-sensitivity

Measurements and Measurability

- Derived from mathematical models

Next Steps

- Production-quality
- Validation + Testing
- Integration into NSRL ops

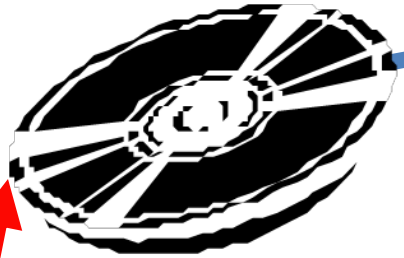
NSRL + SU Use (Data Reduction)

Disk Drive

FILES

Analysis Program

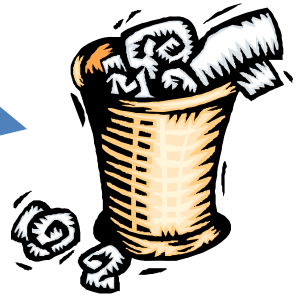
UNKNOWN FILES



RDS

MORE File info

KNOWN FILES



NSRL

- Unpack archives
- Provide FII

Smart Unpacking → {unextractable archives}

***NSRL Unpackers* → {extractable archives}**

SU Useful for Other Things

- SU can add to current NSRL/CF efforts
  - Block hashing
    - locate, optimize blocks to hash
    - Associate parse-rule for fixed-sections with variable content to BH algorithm
      - NOW:
        - » hash according to parse rule rather than fixed content
  - File Carving/Content Analysis
    - Through pattern experiments, build up corpus of useful alphabets, primitives, and rules
    - Make higher-level reductions of content
      - NOW:
        - » Use existing techniques (suffix arrays, bioinformatics) on these abstract, compressed representations rather than just raw data

Contact

Benjamin Long
Software and Systems Division
www.nsrl.nist.gov, NSRL Project
blong@nist.gov

Barbara Guttman
Software and Systems Division
barbara.guttman@nist.gov

Douglas White, NSRL Project Lead
Software and Systems Division
dwhite@nist.gov

Sue Ballou, Office of Law Enforcement Standards
Rep. For State/Local Law Enforcement
susan.ballou@nist.gov