Detecting malicious documents with combined static and dynamic analysis

Virus Bulletin 2009, Geneva

Markus Engelberth, Carsten Willems, Thorsten Holz

Pi1 - Laboratory for Dependable Distributed Systems

UNIVERSITÄT MANNHEIM

Motivation

- Malware in the past: mostly **executable** files
- Targeted attacks use specially prepared **application data files**, e.g., .pdf, .doc, ...
- Example: attacks against European governments and U.S. defense organizations [1]

⇨ MalOffice

[1] NISCC Briefing 08/2005

MalOffice

- Analysis of various **application data** files
- Combination of
  - **Static** analysis
    - general and filetype-depending scanners
  - **Dynamic** analysis
    - CWSandbox
    - Testing analysis reports vs. policies

Outline

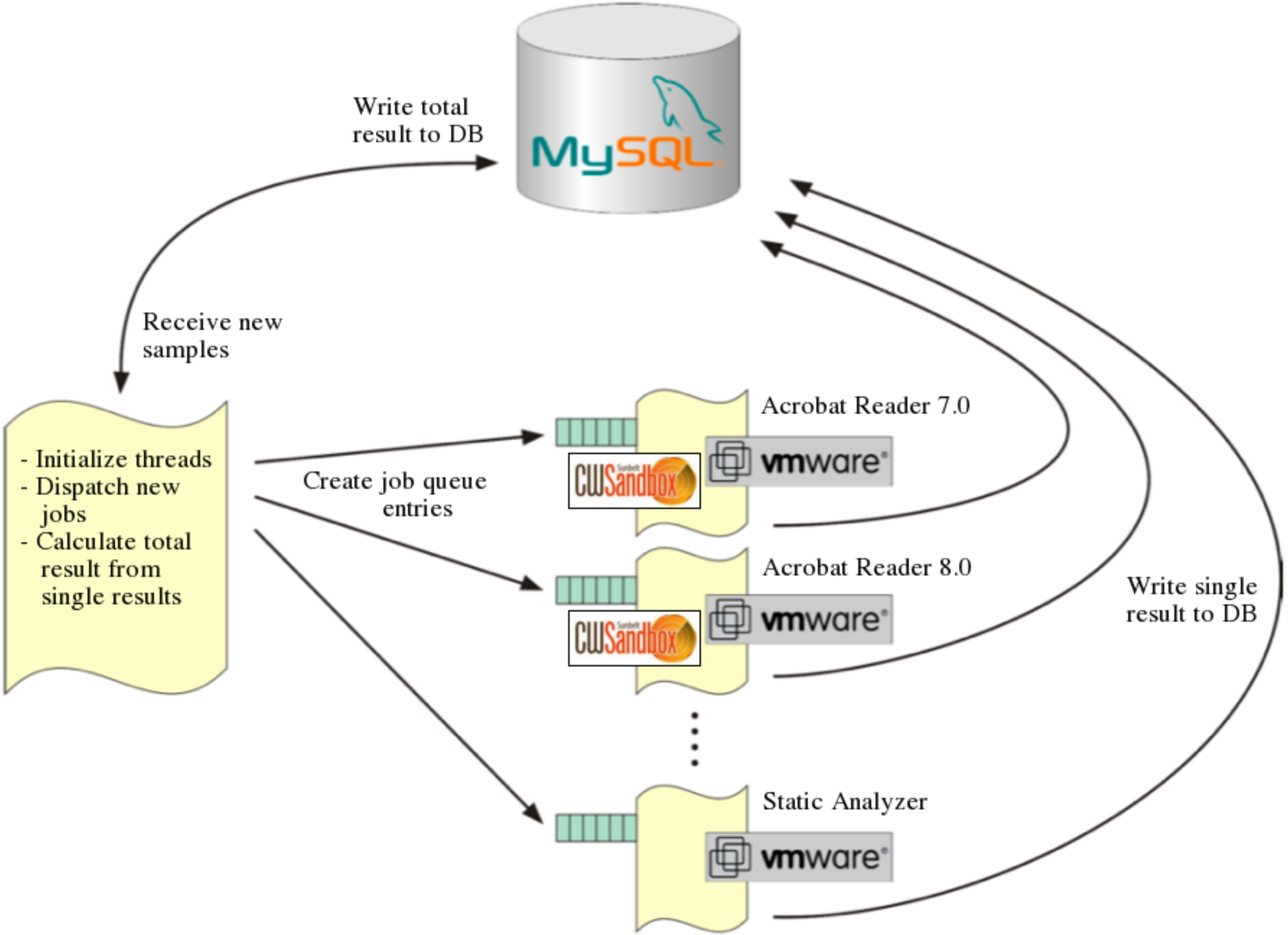
- System overview
- Analysis
  - static analysis / dynamic analysis
  - application policies
  - reaching a verdict
- Example
- Limitations, Future Work, ...

System Overview

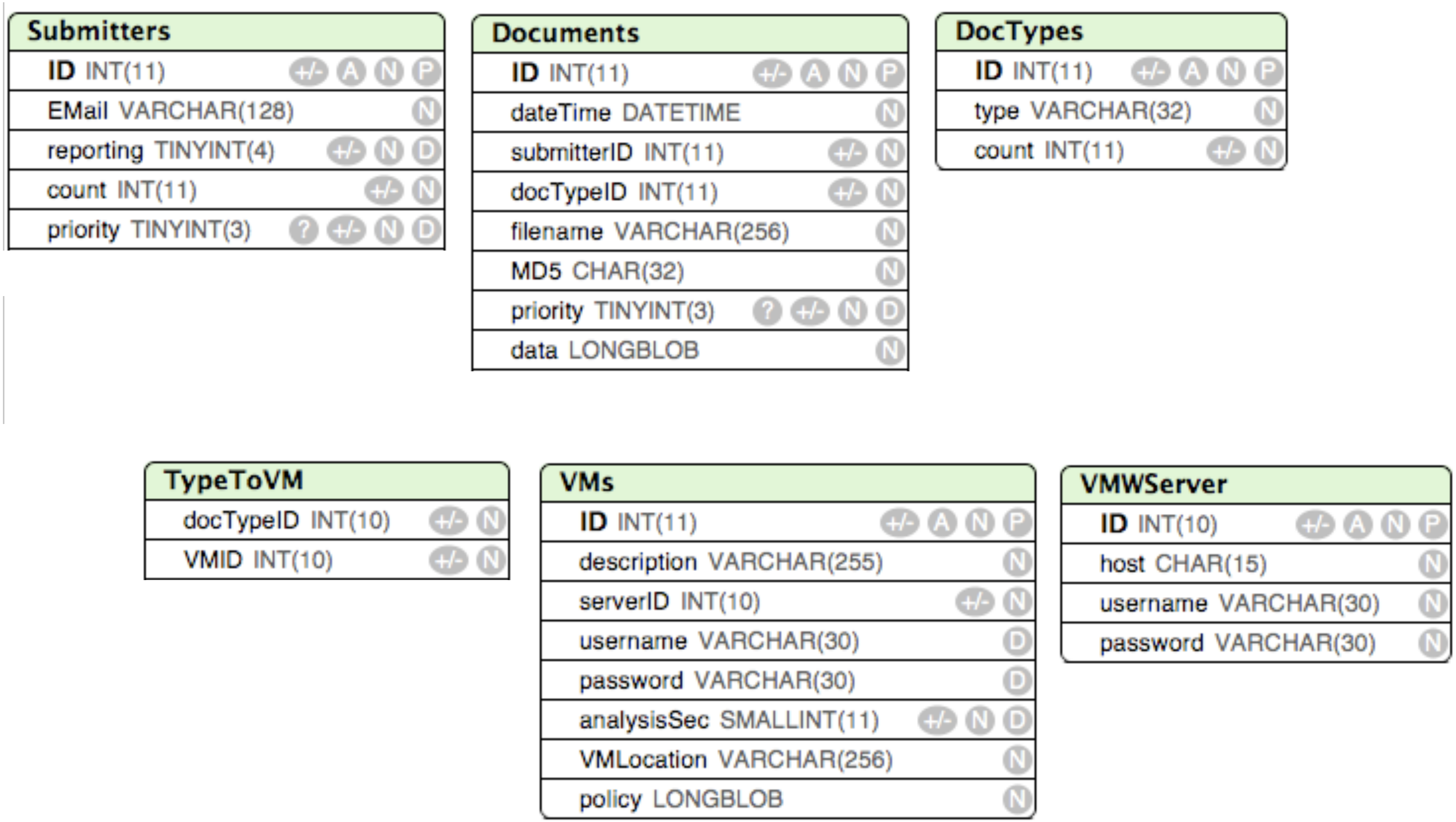
System Overview

- *Database*
  - samples (= to-be-analyzed data files)
  - analysis results
  - other findings
    - embedded PE-files, javascript code, ...
- *Virtual machines* perform analyses
- Python scripts manage and coordinate the whole process

System Overview



System Overview



Submitters		
ID	INT(11)	+/- A N P
EMail	VARCHAR(128)	N
reporting	TINYINT(4)	+/- N D
count	INT(11)	+/- N
priority	TINYINT(3)	? +/- N D

Documents		
ID	INT(11)	+/- A N P
dateTime	DATETIME	N
submitterID	INT(11)	+/- N
docTypeID	INT(11)	+/- N
filename	VARCHAR(256)	N
MD5	CHAR(32)	N
priority	TINYINT(3)	? +/- N D
data	LONGBLOB	N

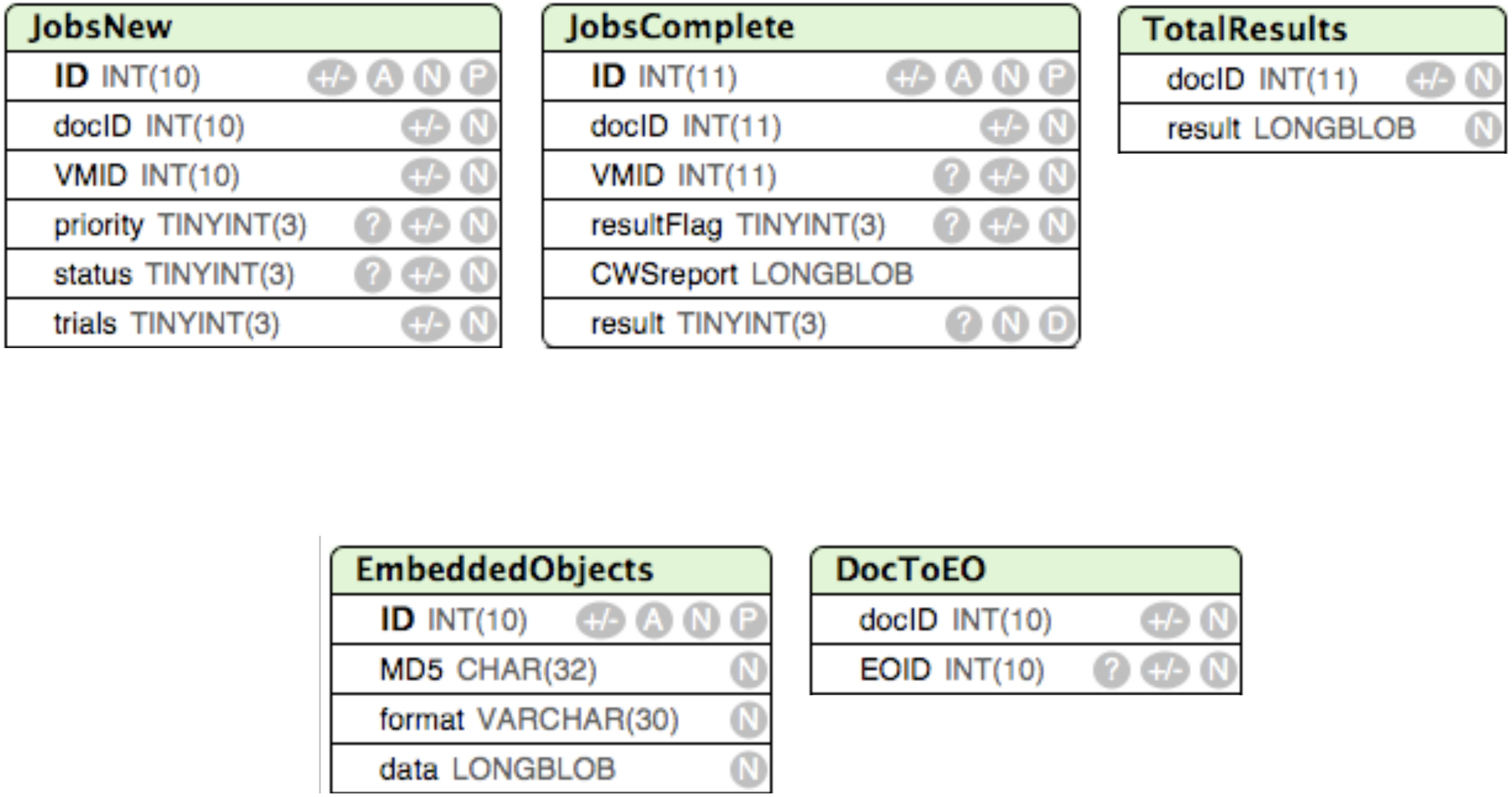
DocTypes		
ID	INT(11)	+/- A N P
type	VARCHAR(32)	N
count	INT(11)	+/- N

TypeToVM		
docTypeID	INT(10)	+/- N
VMID	INT(10)	+/- N

VMs		
ID	INT(11)	+/- A N P
description	VARCHAR(255)	N
serverID	INT(10)	+/- N
username	VARCHAR(30)	D
password	VARCHAR(30)	D
analysisSec	SMALLINT(11)	+/- N D
VMLocation	VARCHAR(256)	N
policy	LONGBLOB	N

VMWServer		
ID	INT(10)	+/- A N P
host	CHAR(15)	N
username	VARCHAR(30)	N
password	VARCHAR(30)	N

System Overview



JobsNew		
ID	INT(10)	+/- A N P
docID	INT(10)	+/- N
VMID	INT(10)	+/- N
priority	TINYINT(3)	? +/- N
status	TINYINT(3)	? +/- N
trials	TINYINT(3)	+/- N

JobsComplete		
ID	INT(11)	+/- A N P
docID	INT(11)	+/- N
VMID	INT(11)	? +/- N
resultFlag	TINYINT(3)	? +/- N
CWSreport	LONGBLOB	
result	TINYINT(3)	? N D

TotalResults		
docID	INT(11)	+/- N
result	LONGBLOB	N

EmbeddedObjects		
ID	INT(10)	+/- A N P
MD5	CHAR(32)	N
format	VARCHAR(30)	N
data	LONGBLOB	N

DocToEO		
docID	INT(10)	+/- N
EOID	INT(10)	? +/- N

Static Analysis

Static analyzers

- *General* scanners
  - AV Scanner
  - PE-detector (plain, XORed)
- *Specialized* scanner per filetype
  - detect embedded javascript in PDF
  - heuristics for malicious javascript
  - detect shellcode in Office documents

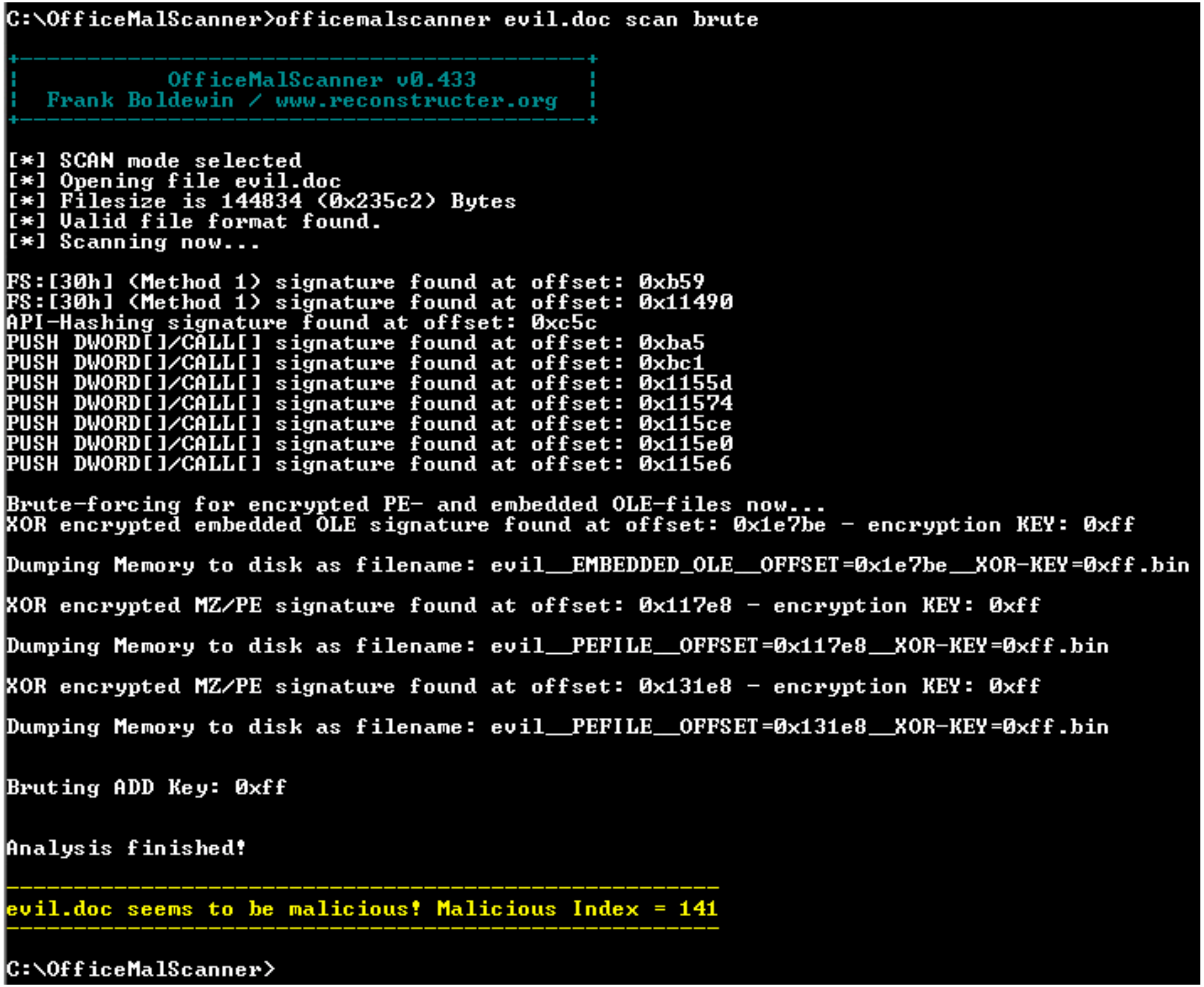
PDF scanner

- Specialized scanner for PDF files
  - decompose PDF stream into objects (pdftoolkit)
  - detect javascript objects
  - use heuristics to detect malicious javascript
    - variable names
    - code obfuscation
    - usage of known vulnerable functions

OfficeMalScanner

- Specialized scanner for MS Word files
  - uses OfficeMalScanner, by Frank Boldewin
  - http://www.reconstructer.org
  - forensic tool for Office documents
    - scans for shellcode pattern
    - dumps OLE structures and VB-macros
    - generates a *malicious index* value

OfficeMalScanner



```
C:\OfficeMalScanner>officemalscanner evil.doc scan brute

+------------------------------------------+
|           OfficeMalScanner v0.433        |
|  Frank Boldewin / www.reconstructer.org  |
+------------------------------------------+

[*] SCAN mode selected
[*] Opening file evil.doc
[*] Filesize is 144834 (0x235c2) Bytes
[*] Valid file format found.
[*] Scanning now...

FS:[30h] (Method 1) signature found at offset: 0xb59
FS:[30h] (Method 1) signature found at offset: 0x11490
API-Hashing signature found at offset: 0xc5c
PUSH DWORD[]/CALL[] signature found at offset: 0xba5
PUSH DWORD[]/CALL[] signature found at offset: 0xbc1
PUSH DWORD[]/CALL[] signature found at offset: 0x1155d
PUSH DWORD[]/CALL[] signature found at offset: 0x11574
PUSH DWORD[]/CALL[] signature found at offset: 0x115ce
PUSH DWORD[]/CALL[] signature found at offset: 0x115e0
PUSH DWORD[]/CALL[] signature found at offset: 0x115e6

Brute-forcing for encrypted PE- and embedded OLE-files now...
XOR encrypted embedded OLE signature found at offset: 0x1e7be - encryption KEY: 0xff

Dumping Memory to disk as filename: evil__EMBEDDED_OLE__OFFSET=0x1e7be__XOR-KEY=0xff.bin

XOR encrypted MZ/PE signature found at offset: 0x117e8 - encryption KEY: 0xff

Dumping Memory to disk as filename: evil__PEFILE__OFFSET=0x117e8__XOR-KEY=0xff.bin

XOR encrypted MZ/PE signature found at offset: 0x131e8 - encryption KEY: 0xff

Dumping Memory to disk as filename: evil__PEFILE__OFFSET=0x131e8__XOR-KEY=0xff.bin


Bruting ADD Key: 0xff


Analysis finished!

-----------------------------------------------------
evil.doc seems to be malicious! Malicious Index = 141
-----------------------------------------------------

C:\OfficeMalScanner>
```

Dynamic Analysis

Dynamic Analysis

CWSandbox Sunbelt

- Tool for automated behavior analysis
- PE-executables or **arbitrary data files**
- Creates XML analysis report: operations executed by the monitored process(es)
  - *filesystem, registry, network, user management, services, protected storage, ...*

Host applications

- Each file type has associated **host application**
  e.g., *Acrobat Reader, Foxit Reader, MS Word, ...*
- Some exploits only trigger in specific app versions
  - use all available host application **versions**
    e.g., *Acrobat Reader* 8.0, 8.1.0, 8.1.1, 9.0, ..
  - *one* sample =>
    multiple host application (versions) =>
    multiple analyses / analysis results

Policies

- Task: decide from analysis report, if executed data file is malicious => **Policies**
  - consist of *white- and blacklisted operations*
  - created in a semi-automated way
- One policy per host application version
  - *what operations are usually perfomed when running this application with a (benign) data file?*

Policy

```
[FILE_DELETE]
+C:\WINDOWS\TEMP\**

[FILE_OPEN]
+$ANALYSIS_TARGET$
+\\.\Ip
+\Device\Tcp
+C:\WINDOWS\TEMP\**
+C:\WINDOWS\System32\spool\DRIVERS\COLOR\sRGB Color Space Profile.icm
+C:\Programme\Gemeinsame Dateien\Adobe\TypeSpt\Unicode\ICU\*
+C:\Programme\Adobe\Acrobat 7.0\Reader\AcroRd32.dll

[FILE_CREATE]
+C:\WINDOWS\TEMP\*
+C:\D&E\Adobe7\Anwendungsdaten\Microsoft\Crypto\RSA\**
+C:\D&E\Adobe7\Anwendungsdaten\Adobe\Acrobat\7.0\Security\CRLCache\*

[REG_CREATE]
+HKEY_CURRENT_USER\SW\Adobe\AR\7.0\Security\cASPKI\cASPKI\cCustomCertPrefs\**
+HKEY_CURRENT_USER\SW\Adobe\AR\7.0\Security\cASPKI\cASPKI\cCustomCertPrefs
+HKEY_CURRENT_USER\SW\Adobe\AR\7.0\Security\cASPKI

[SERVICES]

[PROC_CREATE]
```

Whitelist

- **Whitelist** generation process:

  1) Analyse corpus of known *benign* documents in CWSandbox

  2) Extract and group actions from XML analysis reports

  3) Generalize results with * and **

Blacklist

- **Blacklist** generation process:

  1) Analyze corpus of known *malicious* documents in CWSandbox

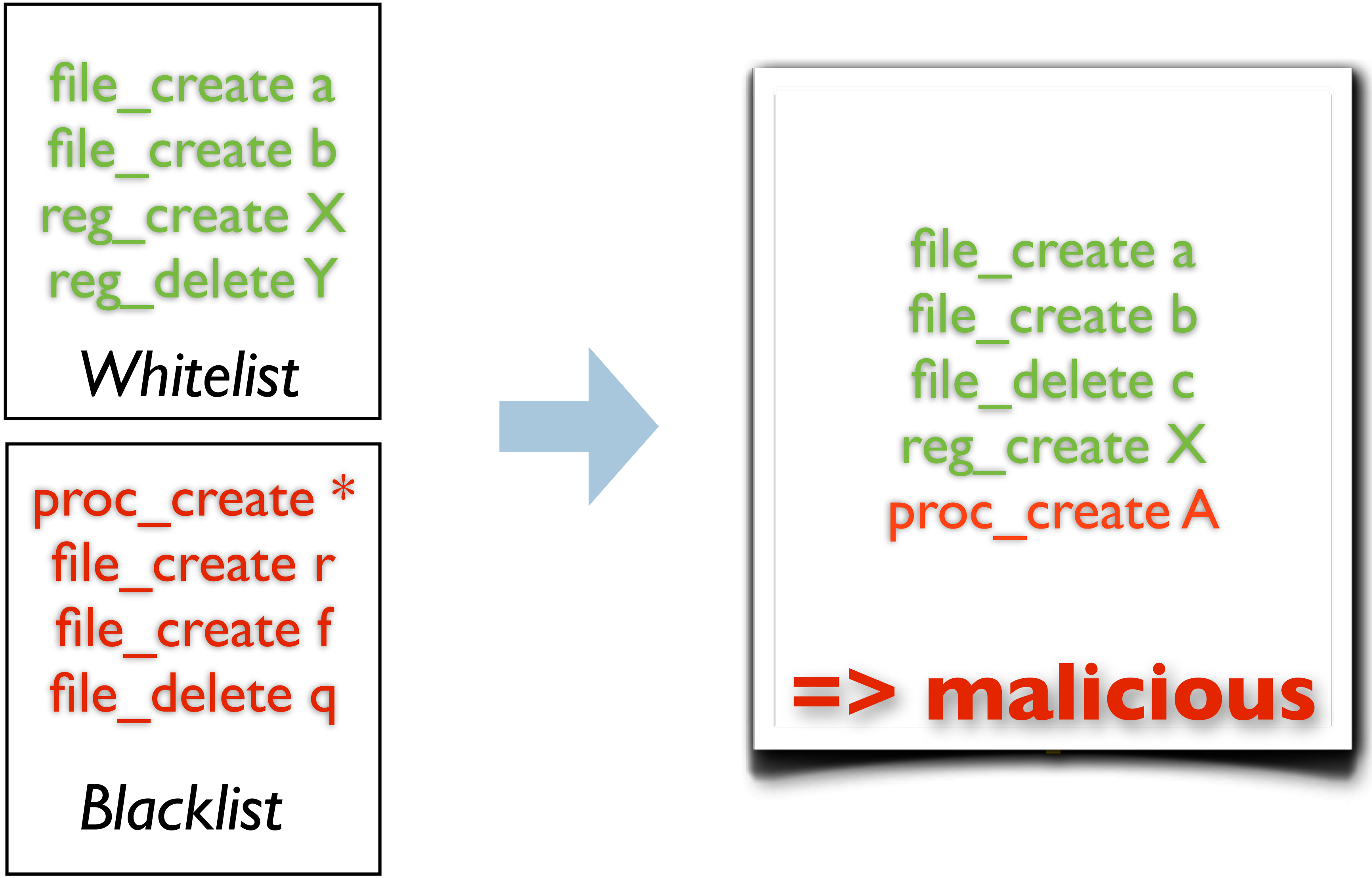
  2) Extract and group actions from XML analysis reports

  3) Remove benign actions (gained by whitelist)

  4) Generalize results with * and **

Policy test

- Test analysis report vs. policy
  - *benign*
    - all operations are whitelisted
  - *malicious*
    - at least one blacklisted operation
  - *suspicious*
    - at least one operation, that is neither whitelisted nor blacklisted



file_create a
file_create b
reg_create X
reg_delete Y

Whitelist

proc_create *
file_create r
file_create f
file_delete q

Blacklist

file_create a
file_create b
file_delete c
reg_create X
proc_create A

=> malicious

Combining Results

Reaching a Verdict

- For each data file => multiple analyses/results
  - (static) general scanner analysis
  - (static) specialized filetype scanner analysis
  - (dynamic) multiple CWSandbox analyses
    - one per host application version
- Need to combine multiple sub-results into one **total** result

Reaching a Verdict

- Numeric values for results:
  - `0.0`: *benign*
  - `0.5`: *suspicious*
  - `1.0`: *malicious*
- Total result =
  - `1.0`, if one single result is `1.0`
  - ∅ single results otherwise (`0.0` ... `0.5`)

Example

Evaluation

- Only a small corpus of malicious documents
  - no real evaluation possible
  - demonstration by an example
    - *addresses_of_TSGS_in_Italy.pdf*
    - *Collab.collectEmailInfo* (CVE-2007-5659)

Static Analysis Result

Extracted Javacript:

```
function start() {
  sc = unescape(\"%u9090%u9090%u9090%u9090%uEB90%u5E1a...");
  if (app.viewerVersion >= 7.0) {
    plin = re(1124,unescape(\"%u0b0b%u0028%u06eb%u06eb\")) +
           unescape(\"%u0b0b%u0028%u0aeb%u0aeb\") + unescape(\"%u9090%u9090\") +
           re(122,unescape(\"%u0b0b%u0028%u06eb%u06eb\")) + sc +
           re(1256,unescape(\"%u4141%u4141\"));
  }
  else {
    ef6 = unescape(\"%uf6eb%uf6eb\") + unescape(\"%u0b0b%u0019\");
    plin = re(80,unescape(\"%u9090%u9090\")) + sc +
           re(80,unescape(\"%u9090%u9090\"))+ ...
    while ((plin.length % 8) != 0)
      plin = unescape(\"%u4141\") + plin;
    plin += re(2626,ef6);
  }
  if (app.viewerVersion >= 6.0) {
    this.collabStore = Collab.collectEmailInfo({subj: \"\",msg: plin});
  }
}
```

=> suspicious

Dynamic Analysis Result

Violations of Policy “Adobe Reader 7.0”:

```
FILE_DELETE     c:\a.exe
FILE_OPEN       c:\a.exe
FILE_OPEN       C:\WINDOWS\system32\hal.dll
FILE_OPEN       C:\WINDOWS\system32\sys.exe
PROC_KILL       kill_process
FILE_CREATE     c:\a.exe
FILE_CREATE     C:\WINDOWS\system32\sys.exe
FILE_CREATE     C:\WINDOWS\TEMP\winsxvs.exe
FILE_CREATE     C:\WINDOWS\TEMP\audel.bat
PROC_CREATE     c:\a.exe
PROC_CREATE     C:\WINDOWS\TEMP\winsxvs.exe
PROC_CREATE     C:\WINDOWS\TEMP\audel.bat
PROC_CREATE     C:\Programme\Internet Explorer\IEXPLORE.EXE -nohome
```

=> malicious

Combined Result

Static	General: ClamAV	0
Static	General: PE-Detect	0
Static	Specialized: PDF-Files	0,5
Dynamic	Acrobat Reader 7.0	1,0
Dynamic	Acrobat Reader 8.1.2	0
Dynamic	Acrobat Reader 9.0	0

=> 1.0 (malicious)

Outro

Related Work

- *SPARSE* by Li and Stolfo
  - focussed only on Word documents
- *OfficeCat* by Sourcefire
  - static scanner for Office documents
- *OfficeMalScanner* by Frank Boldewin
- *Wepawet* by UCSB
  - powerful tool to analyze PDF and Flash files

Limitations

- Static analyis can be *circumvented* by attacker
  - different kinds of obfuscation are possible
  - general drawback of static malware analysis
- No user-interaction yet
  - exploit might trigger only on certain events
- Exploit might require specific version
  - partly addressed by multiple versions of each tool

Future Work

- More file types
- Polished static analysis
- Webinterface
- Stability and performance improvements

Conclusion

- MalOffice: approach to combine both static and dynamic analysis
  - use static signatures and heuristics for detecting exploits
  - combined with powerful dynamic analysis
- Can be used to examine arbitrary data files
  - PDF, Microsoft Office, Flash, ...
- Results look promising, more tests needed

Carsten Willems

http://pi1.informatik.uni-mannheim.de/
cwillems@cwse.de

Thanks for your attention!
Any questions?

Pi1 - Laboratory for Dependable Distributed Systems

UNIVERSITÄT
MANNHEIM