PackTest 2004-07

An anti-malware product test conducted by the antiVirusTestCenter, University of Hamburg

Content

1. Introduction

1. a) About us

This test was conducted by students at the antiVirusTestCenter, University of Hamburg:

Heiko Fangmeier (5fangmei@informatik.uni-hamburg.de)	➔ Product testing ➔ Test report
Michel Messerschmidt (uni@michel-messerschmidt.de)	➔ Concept ➔ Testbed creation ➔ Evaluation ➔ QA ➔ Test report
Fabian Müller (9fmuelle@informatik.uni-hamburg.de)	➔ Product testing ➔ Test report ➔ Graphical presentation
Jan Seedorf (seedorf@informatik.uni-hamburg.de)	➔ Concept ➔ Product testing ➔ Test report

1. b) Background of this test

This test targets the detection of malware in compressed files.

Nearly all anti-malware product tests focus on the detection rate as the most important fact. However, the question whether anti-malware products can detect and protect from malware in compressed formats is basically unanswered. The antiVirusTestCenter (aVTC) at the university of Hamburg has developed a sound methodology for measuring the detection quality of anti-malware software. To measure the detection quality for compressed malware, this methodology was adapted to answer the following questions more thoroughly than before:

- Which product supports which compression format?

  The test shall deliver results that show which format is/ is not supported by a given product at all.

- What is the detection quality of the supported formats?

  The test should reveal the quality of the decompression engine[1] of each anti-malware product. The anti-malware software should have the same detection rate on compressed malware compared to the same uncompressed malware.

- Does the anti-malware software support every version of compression/archive formats?

  Some compression formats differ from previous versions (for example RAR). This test challenges each anti-malware product with each version of a compression format. If some versions of a compression format are not supported, a user could get a false sense of security. Additionally, some archive formats support different modes of compression (e. g. self-extracting archives, solid archives, etc.). These are also included in this test, as anti-malware products should support them.

- How do the tested anti-malware programs handle problems with compressed files?

  If an anti-malware product is not able to decompress or scan a compressed file, the user should be informed to prevent a malicious file to enter the system.

- Do the tested anti-malware programs detect malware in recursive compressed archives?

  A strong protection through anti-malware software will discover malware even in recursive compressed archives (archives within archives).

Our methodology does not include a systematic test of the availability of anti-malware products. However, specific archives may threaten the availability of the anti-malware programs through denial-of-service attacks (see http://www.aerasec.de/security/advisories/txt/bzip2bomb-antivirusengines.txt for an example) and will be mentioned when occurred during the test.

2. Test methodology

The anti Virus Test Center (aVTC) of the university of Hamburg tests anti-malware software in a closed environment and based on ethical standards (see ftp://agn-www.informatik.uni-hamburg.de/pub/CodeConduct/CoC-016.txt). For most of our tests, each product is tested in on-demand mode and scans the malware in a testbed, stored on a Windows NT 4.0 file server. The detection quality (detected malware in the testbed), reliable identification (equal identification of different samples of the same variant) and detection reliability (reliable detection of all infected samples of a variant) are evaluated through parsing the log files of each product.

The comparability of the products is given through a submission day for the product versions and their virus definitions. All test reports of the aVTC are published including detailed software configuration, hardware, test environment and test methodology to deliver reproducable scientific results[2].

The directory structure of the aVTC testbeds is hierarchical:

[1] Anti-malware products usually will decompress an archive and then compare the temporarily extracted files with their virus patterns. However, some products have different approaches, for example separate patterns for each combination of malware and compression format.

[2] A detailed description of the test methodology can be found at ftp://agn-www.informatik.uni-hamburg.de/pub/texts/tests/pc-av/2003-04/

```
<testbed>\<platform>\<malwarefamily>\<variant>\<sample>
```

A perl script (see http://www.michel-messerschmidt/en/avtctest.html) is used to parse and evaluate the log files. The log file is parsed in several steps:

- The log file is changed to a common format (separation of the log in path of the tested object, message of test product, description of found malware)
- The log file is split in reported infected files, not infected files and else lines.
- The detection quality and other criteria is calculated.
- Manual quality assurance of the evaluation.

In case of inconsistencies during the evaluation of the logfiles or if a product did not report all objects of the testbed, the missing objects are repeatedly scanned and evaluated up to two times. This repeated testing enhances the chances for the products to scan all objects and to minimize the sources for other mistakes.

2. a) Test specific adaptions

Due to the special research topic of compressed malware, this test methodology had to be adapted. This test evaluates the quality of support for the specific compression formats. Therefore the test data is scanned uncompressed (reference testbed) as well as in each specific compression format. The difference of the detection quality of the compressed to the uncompressed testbed gives the quality of support for the specific compression format. The loss of the detection quality (in percent points) per product and compression format can be calculated as follows:

$$\text{loss of detection quality}_{\text{product, format}}$$
$$= \text{detection quality of reference testbed}_{\text{product}} - \text{detection quality}_{\text{product, format}}$$

The compression formats are used in the following modes (if possible), to detect less obvious vulnerabilities of anti-malware software products:

- standard compression[3]
- complete reference testbed (incl. directory structure) in one archive
- the archive samples are renamed (generic name without file extension)
- recursive archives: each compression is done 2x or 9x times
- creation of self-extracting archives
- creation of password-protected archives

Different format versions and special modes of single compression formats[4] are described in this test as compression formats.

3. Description of the test environment and testbeds

Because this test was done in parallel to a regular test, only one computer (P4, 1.8 GHz, 512 MB Ram, 80 GB HD) could be used. Therefore the test was conducted on an isolated Windows 2000 system with testbeds stored locally on a separate write-protected NTFS partition. The hypothesis that the results are transferable to other Win32-operating systems (for example Windows XP) is not discussed in this report.

According to the test methodology of the aVTC each product was installed on a clean operating

[3] Our "standard" compressed testbed consists of a compressed archive per subdirectory and compression format. Therefore all samples of a malware variant are contained in the same archive, archives contain only files but no directory structures and the testbed contains many archives for each format. All archives are created with that formats' standard compression options.

[4] e.g. rar: solid archive

system and for scanning a specific testbed a disk image was restored.

All together 26 anti-malware products have been tested on the detection of 32 compression- and archive-formats (incl. different format-versions) on 8 testbeds (7 different compression modes plus the reference testbed). These are listed below.

For this test the reference testbed constituted of „in-the-wild“ file viruses[5] , which are listed in the „Wildlist“ for October 2001 (the list can be found at http://www.wildlist.org/WildList/200110.htm). These relatively old viruses are assumed to be well known by most anti-malware products. This leads to the assumption that almost every anti-malware product is able to detect all (or nearly all) samples in the reference testbed, so that the results on the compressed testbeds will be comparable.

Testbeds			
Code	Testbed type	Archives	Compression Formats
FI	File in-the-wild (reference testbed)	-	-
P	"Standard" archives	1600	32
P2	Obfuscated archives	1600	32
Q	Archives containing the complete testbed	28	28
R	Recursive compressed archives (2 levels)	1600	32
R2	Recursive compressed archives (9 levels)	1600	32
S	Self-extracting archives	700	14
E	Encrypted archives	800	18

The reference testbed ("FI") contains a total of 442 different samples (files) from 50 different "File in-the-wild" viruses or virus variants (with each virus in a different subdirectory).

For all supported compression formats each compressed testbed contains all files from the reference testbed in the archives of this format.

Our "standard" compressed testbed ("P") consists of a compressed archive per subdirectory and compression format. Therefore all samples of a malware variant are contained in the same archive, archives contain only files but no directory structures and the testbed contains many archives for each format. All archives are created with that formats' standard compression options. All archives have a common filename with the default extension for this compression format (e. g. "ZIP.ZIP" for ZIP archives,"ZIB.ZIP" for ZIB archives, etc). For compression formats that support only single files (for example Gzip, Bzip2, Base64, UUEncode) each sample is compressed separately.

The "obfuscated" testbed ("P2") has exactly the same contents, but the archives have generic filenames without filename extension (e.g. "VTC27VTC" instead of "ZIP.ZIP").

The archives in the "complete" testbed ("Q") contain all directory structures from the reference testbed. For compression formats that don't support structured archives themselves (for example Gzip), a TAR archive containing the reference testbed is created first, on which the compression format is applied.

For the "recursive" testbeds ("R" and "R2") archives are compressed with our "standard" method but several times (with the same compression format).

Archives in the "encrypted" ("E") and "self-extracting" ("S") testbeds are created with the "standard"

[5] The complete testbed of the aVTC is divided into platform specific parts. Thus, there are macro, script, file and boot viruses as testbeds. In this test of compressed malware the „file-in-the-wild“-viruses are used as reference testbed. The hypothesis is, that the tested products will first decompress the malicious software and the decompression routines will work in the same way for all other testbeds (e.g. script viruses).

options but with additional options to password-protect the archive (with the password "packtest") or to create a self-extracting archive.

Tested anti-malware products		*Compression formats*	
Code	*Product name*	*Code*	*Format name*
ANT	H+B EDV Antivir	7Z_	7-Zip
AVA	Alwill Avast!	ACE	Ace v1
AVG	Grisoft Antivirus System	AC2	Ace v2
AVK	GData AntiVirenKit	ARC	Arc
AVP	Kaspersky Antivirus	ARJ	Arj
BDF	BitDefender	B64	MIME Base64
CMD	Command Antivirus	BH_	Black Hole
DRW	Dr. Web	BZ2	Bzip2
FIR	Fire Anti-Virus Kit	CAB	MS Cabinet File
FPR	F-Prot for Windows	CMS	MS Compress
FSE	F-Secure Anti Virus	GZ_	Gzip
GLA	Gladiator Antivirus	HA_	Ha
IKA	Ikarus Virus Utilities	JAR	Jar
INO	eTrust Antivirus	JAV	Java Archive
NAV	Symantec Antivirus	LHA	Lha
NVC	Norman Virus Control	PAK	Pak
PAV	GData PowerAntivirus	RA1	Rar v1
PER	Per Antivirus	RA2	Rar v2
PRO	Protector	RA3	Rar v3
QHL	Quickheal	RAR	Rar v3 (solid compression)
RAV	RAV Antivirus	SHA	Shell Archive (shar)
SCN	McAfee ViruScan	SQZ	Squeeze It
SWP	Sophos Anti Virus	TAR	Tape Archive
VBR	VirusBuster	UC2	Ultra Compressor 2
VSP	VirScanPlus	UUE	UUEncode
		ZIP	InfoZip 2.3
		ZI2	PkZip 6.0 (zip2.04 compatible)
		ZI6	PkZip 6.0 (default compression)
		ZIB	PkZip 6.0 (bzip2 compression)
		ZID	PkZip 6.0 (DCLimplode compression)
		ZIE	PkZip 6.0 (Deflate64 compression)
		ZOO	Zoo

4. Test results

4. a) Selected observations and problems

On testing compressed files without file extension, some products did not detect any viruses (AVG all supported formats, CMD and FPR in „LHA"-archives).

When testing the password-protected archives the detection rate was at 0% (as expected). However, many tested anti-malware products reported these files as „not infected" or „OK" or not at all (ANT, AVG, GLA, IKA, INO, PER, PRO, RAV, SCN, VBR, VSP). To evaluate the risk of these files, the anti-malware software should at least report that such archives could not be scanned (additionally a reason could be helpful for the user, e.g. „password-protected file").

It should be mentioned, that no product in this test fully supports all modes of the ZIP format. All products had difficulties in the decompression of two modes of this format (ZIB, ZID). Although these modes are not widely in use[6], they are completely valid ZIP archives that should be supported by any product claiming "full ZIP support".

Even more alarming is the fact that many anti-malware products don't support "Java archives" (JAV[7]), since this format is a plain ZIP archive (pkzip 2.04 compatible) with some additional semantics regarding the contents. Additionally this format is used by many applications to transfer executable code. It is even supported internally by some browsers (e.g. mozilla). Given that most products are technically able to scan the JAV format (as they support the ZIP format), it seems to be a vendor decision not to scan this archives (or make this even configurable).

In addition some technical problems occurred while testing. Many anti-malware products could not fully report the content of the archives (ANT, BDF, FPR, GLA, INO, NAV, QHL, SCN). This weakness occurred especially in archives with directory structures and multi compressed archives. With some products this behaviour could be found as a general problem for the complete test (ANT, BDF, GLA, NAV).

In testing the recursive compressed archives software stability problems have been observed (FSE, NVC, PRO, SCN crashed repeatedly). Some of the products supporting „HA"-format needed several hours for scanning a single „HA"-archive (RAV, AVK[8]).

4. b) Result matrices

This section gives an overview over the support of compression formats for each tested anti-malware product. The results for each testbed are presented as a colored matrix with the rows representing the compression formats and the columns as the anti-malware products.

A dark green cell means that no loss of detection (0%) in comparison to the reference testbed occurred, thus this compression format is fully supported. This does **not** mean that the detection rate is 100%, but that the detection rate on the compressed testbed is equal to the detection rate on the reference testbed. One product (AVA) even manages to achieve higher detection rates for compressed testbeds than for the reference testbed. These results are also treated as "fully supported" and thus as dark green cells but with a negative percentage value in the the cell to denote the raised detection rate in relation to the reference testbed.

The opposite is a dark red cell. In this case the loss of detection is 100% (which means that the detection rate is reduced to 0%), stating that this compression format is not supported by the anti-malware product at all.

Light green and orange cells denote a loss of detection between 0% and 100% (with the exact value given in the cell). For the light green cells (0.1% - 20% loss) this could be an indication that these formats are supported in general, but support is only partial or not free of errors. This could point to an implementation problem of the compression engines and lead to a vulnerability for compressed malware. For the orange cells (20.1% - 99.9% loss) it seems reasonable to assume at least a severe

[6] Only commercial ZIP compression tools support these modes for now, most shareware compression tools (and therefore most zip archives) today are "pkzip 2.04 compatible"

[7] These are the archives with the default filename extension .jar first used by the JavaVM. Not to be confused with the compression format JAR from ARJsoft with has the default filename extension .j

[8] As AVK internally uses the RAV scan engine , this problem could be a single weakness of this scan engine.

problem with these compression formats.

Finally, the matrices contain an additional row labeled "_A_". The results in this row show the detection loss for all **files** in the testbed, i.e. the archives themselves (for all formats) are counted here but not the archive contents, while the results in all other rows represent only the archive contents for a specific compression format. We decided to add this additional evaluation data because some products (for example ANT) never report archive contents but only the archives files. Therefore these products couldn't achieve any positive detection rate in our traditional content-centered evaluation routines. The real detection loss will be somewhere between the "_A_" value and the format specific value but can't be obtained exactly with our current test method.

The absolute detection rates for the reference testbed are listed at the end of this section to show the validity of the assumption from section 3 (about the comparability of the results).

Result matrix for the "standard compressed" testbed ("P")

	ANT	AVA	AVG	AVK	AVP	BDF	CMD	DRW	FIR	FPR	FSE	GLA	IKA	INO	NAV	NVC	PAV	PER	PRO	QHL	RAV	SCN	SWP	VBR	VSP
_A_	26,9	🟩	23,6	🟩	🟩	🟩	🟩	🟩	🟥	🟩	🟩	🟩	87,2	🟩	🟩	🟩	🟩	53,3	25,6	🟩	🟩	🟩	🟩	🟥	0,0
7Z_	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
AC2	🟥	🟥	🟥	🟩	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟩	🟥	🟥	0,9	92,1	🟥	🟥	🟥	🟥
ACE	🟥	🟥	🟥	🟩	🟩	🟩	🟩	🟥	🟥	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟩	🟥	🟥	0,9	6,6	🟩	🟥	🟥	🟥
ARC	🟥	🟥	🟥	30,3	🟥	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟩	🟥	🟥	🟥	🟥	32,8	🟩	🟥	🟥	🟥
ARJ	🟥	-0,2	🟩	🟩	🟩	🟩	🟩	🟩	🟥	🟩	🟥	🟥	🟥	🟩	🟥	🟩	🟩	🟥	🟩	0,9	🟩	🟩	🟩	🟥	🟥
B64	🟥	0,2	🟥	🟩	🟩	🟥	🟥	🟩	🟥	🟥	🟩	🟥	34,0	🟥	🟩	🟥	🟩	🟥	97,7	🟥	🟩	🟩	🟥	🟥	🟥
BH_	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
BZ2	🟥	🟥	🟥	🟩	🟩	🟩	🟥	🟥	🟥	🟥	🟩	🟥	🟥	🟥	🟥	🟥	🟩	🟥	🟥	🟥	2,9	🟩	🟥	🟥	🟥
CAB	🟥	-0,2	🟥	🟩	🟩	🟩	🟩	🟩	🟥	🟩	🟩	🟥	🟥	🟩	🟩	🟥	🟩	🟥	🟩	0,9	🟩	🟩	🟥	🟥	🟥
CMS	🟥	🟥	🟥	🟩	🟩	🟩	0,9	🟥	🟥	0,9	🟥	🟥	🟥	🟥	🟩	🟥	🟩	🟥	🟥	🟥	🟩	🟩	🟩	🟥	🟥
GZ_	99,8	-0,2	🟥	🟩	🟩	🟩	🟩	🟩	🟥	🟩	🟩	🟥	🟥	🟩	🟩	🟩	🟩	🟥	🟥	🟥	🟩	🟩	🟩	🟥	0,0
HA_	🟥	🟥	🟥	🟩	🟩	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟩	🟥	🟥	🟥	69,2	🟥	🟥	🟥	🟥
JAR	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
JAV	🟥	-0,2	🟥	🟥	🟥	🟩	🟥	🟥	🟥	🟥	🟩	🟩	🟥	🟩	60,0	🟩	🟥	🟥	🟩	0,9	🟥	🟥	🟥	🟥	🟥
LHA	🟥	🟥	🟥	🟩	🟩	🟩	🟩	🟥	🟥	🟩	🟩	🟥	🟥	6,3	0,7	🟥	0,5	🟥	🟥	🟥	🟩	🟩	🟩	🟥	🟥
PAK	🟥	🟥	🟥	🟥	🟥	97,9	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
RA1	🟥	🟩	🟩	🟩	🟩	🟩	95,0	🟩	🟥	95,0	🟩	🟥	🟥	99,5	🟥	🟥	🟩	🟥	🟥	0,9	🟩	🟩	🟩	🟥	🟥
RA2	🟥	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟥	🟩	🟩	🟥	🟥	🟩	🟥	🟩	🟩	🟥	🟥	0,9	🟩	🟩	🟩	🟥	🟥
RA3	🟥	-0,2	🟥	🟩	🟩	🟩	🟥	🟩	🟥	🟥	🟩	🟥	🟥	🟥	🟥	🟥	🟩	🟥	🟥	0,9	🟩	🟥	🟥	🟥	🟥
RAR	🟥	-0,2	🟥	🟩	🟩	🟩	🟥	🟩	🟥	🟥	🟩	🟥	🟥	🟥	🟥	🟥	🟩	🟥	🟥	0,9	🟩	🟥	🟥	🟥	🟥
SHA	🟥	🟥	🟥	🟥	55,2	🟥	89,3	🟥	🟥	89,1	🟥	🟥	🟥	55,7	🟥	56,2	55,2	🟥	🟥	🟥	🟥	🟥	55,3	🟥	🟥
SQZ	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
TAR	🟥	0,5	🟥	🟩	🟩	8,9	🟩	🟩	🟥	🟩	🟩	🟥	🟥	🟥	🟩	🟩	🟩	🟥	🟥	🟥	🟩	🟩	2,7	🟥	🟥
UC2	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
UUE	🟥	🟥	🟥	🟩	🟩	🟩	🟩	🟩	🟥	🟩	🟩	🟥	🟥	0,7	🟩	🟥	🟩	🟥	🟥	🟥	🟩	🟩	🟩	🟥	🟥
ZI2	🟥	-0,2	1,4	🟩	🟩	🟩	🟩	🟩	🟥	🟩	🟩	🟩	🟥	🟩	🟩	🟩	🟩	🟩	🟩	0,9	🟩	🟩	🟩	🟥	🟥
ZI6	🟥	🟩	1,4	🟩	🟩	🟩	🟩	🟩	🟥	🟩	🟩	🟩	🟥	🟩	🟩	🟩	🟩	🟩	🟩	0,9	🟩	🟩	🟩	🟥	🟥
ZIB	🟥	97,5	97,5	97,5	🟥	97,4	🟥	🟥	🟥	🟥	🟥	72,2	🟥	🟥	97,5	🟥	🟥	99,3	97,5	🟥	97,5	97,5	97,5	🟥	🟥
ZID	🟥	82,0	82,0	81,7	90,5	81,1	90,9	90,5	🟥	90,5	90,5	54,4	🟥	90,5	81,7	82,3	90,5	86,0	82,2	91,3	81,7	81,7	82,3	🟥	🟥
ZIE	🟥	🟩	99,5	99,5	🟥	🟩	🟥	🟥	🟥	🟥	🟥	🟩	🟥	🟥	80,1	🟩	🟥	🟥	99,5	🟥	99,5	99,5	99,5	🟥	🟥
ZIP	🟥	🟩	2,3	🟩	🟩	🟩	🟩	🟩	🟥	🟩	🟩	🟩	🟥	🟩	22,2	🟩	🟩	🟩	🟩	0,9	🟩	🟩	🟩	🟥	🟥
ZOO	🟥	🟥	🟥	🟩	🟥	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟩	🟥	🟥	🟥	🟥

Loss of detection (in percent): 🟩 0,0% 🟦 0,1% - 20,0% 🟧 20,1% – 99,9% 🟥 100,0%

Result matrix for the "obfuscated" testbed ("P2")

	ANT	AVA	AVG	AVK	AVP	BDF	CMD	DRW	FIR	FPR	FSE	GLA	IKA	INO	NAV	NVC	PAV	PER	PRO	QHL	RAV	SCN	SWP	VBR	VSP
_A_	26,9	🟩	95,5	🟩	🟩	🟩	🟩	🟩	🟥	🟩	🟩	🟩	87,2	🟩	🟩	🟩	🟩	53,3	25,6	🟩	🟩	🟩	🟩	🟥	98,3
7Z_	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
AC2	🟥	🟥	🟥	🟩	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟩	🟥	🟥	🟥	92,1	🟥	🟥	🟥	🟥
ACE	🟥	🟥	🟥	🟩	🟩	🟩	🟩	🟥	🟥	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟩	🟥	🟥	🟥	6,6	🟩	🟥	🟥	🟥
ARC	🟥	🟥	🟥	32,8	🟥	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟩	🟥	🟥	🟥	🟥	32,8	🟩	🟥	🟥	🟥
ARJ	🟥	-0,2	🟥	🟩	🟩	🟩	🟩	🟩	🟥	🟩	🟥	🟥	🟥	🟩	🟥	🟩	🟩	🟥	🟩	0,9	🟩	🟩	🟩	🟥	🟥
B64	🟥	0,2	🟥	🟩	🟩	🟥	🟥	🟩	🟥	🟥	57,5	🟥	34,0	🟥	🟩	🟥	🟩	🟥	97,7	🟥	🟩	🟩	🟥	🟥	🟥
BH_	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
BZ2	🟥	🟥	🟥	🟩	🟩	🟩	🟥	🟥	🟥	🟥	57,5	🟥	🟥	🟥	🟥	🟥	🟩	🟥	🟥	🟥	2,9	🟩	🟥	🟥	🟥
CAB	🟥	-0,2	🟥	🟩	🟩	🟩	🟩	🟩	🟥	🟩	62,4	🟥	🟥	🟩	🟩	🟥	🟩	🟥	🟩	0,9	🟩	🟩	🟥	🟥	🟥
CMS	🟥	🟥	🟥	🟩	🟩	🟩	0,9	🟥	🟥	0,9	🟥	🟥	🟥	🟥	🟩	🟥	🟩	🟥	🟥	🟥	🟩	🟩	🟩	🟥	🟥
GZ_	99,8	🟩	🟥	🟩	🟩	🟩	🟩	🟩	🟥	🟩	57,5	🟥	🟥	🟩	🟩	🟩	🟩	🟥	🟥	🟥	🟩	🟩	🟩	🟥	99,2
HA_	🟥	🟥	🟥	🟩	🟩	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟩	🟥	🟥	🟥	69,2	🟥	🟥	🟥	🟥
JAR	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
JAV	🟥	-0,2	🟥	🟥	🟥	🟩	🟥	🟥	🟥	🟥	62,4	🟩	🟥	🟩	60,0	🟩	🟥	🟥	🟩	0,9	🟥	🟥	🟥	🟥	🟥
LHA	🟥	🟥	🟥	🟩	🟩	0,2	🟥	🟥	🟥	🟥	62,4	🟥	🟥	6,3	0,7	🟥	0,5	🟥	🟥	🟥	2,3	🟩	🟩	🟥	🟥
PAK	🟥	🟥	🟥	🟥	🟥	97,9	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
RA1	🟥	-0,2	🟥	🟩	🟩	🟩	95,0	🟩	🟥	95,0	62,4	🟥	🟥	99,5	🟥	🟥	🟩	🟥	🟥	0,9	🟩	🟩	🟩	🟥	🟥
RA2	🟥	🟩	🟥	🟩	🟩	🟩	🟩	🟩	🟥	🟩	62,4	🟥	🟥	🟩	🟥	🟩	🟩	🟥	🟥	0,9	🟩	🟩	🟩	🟥	🟥
RA3	🟥	-0,2	🟥	🟩	🟩	🟩	🟥	🟩	🟥	🟥	62,4	🟥	🟥	🟥	🟥	🟥	🟩	🟥	🟥	0,9	🟩	🟥	🟥	🟥	🟥
RAR	🟥	🟩	🟥	🟩	🟩	🟩	🟥	🟩	🟥	🟥	62,4	🟥	🟥	🟥	🟥	🟥	🟩	🟥	🟥	0,9	🟩	🟥	🟥	🟥	🟥
SHA	🟥	🟥	🟥	🟥	55,2	🟥	89,3	🟥	🟥	89,1	🟥	🟥	🟥	55,7	🟥	56,2	55,2	🟥	🟥	🟥	🟥	🟥	55,3	🟥	🟥
SQZ	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
TAR	🟥	0,5	🟥	🟩	🟩	8,9	🟩	🟩	🟥	🟩	62,4	🟥	🟥	🟥	🟩	🟩	🟩	🟥	🟥	🟥	🟩	🟩	2,7	🟥	🟥
UC2	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
UUE	🟥	🟥	🟥	🟩	🟩	🟩	🟩	🟩	🟥	🟩	57,5	🟥	🟥	0,7	🟩	🟥	🟩	🟥	🟥	🟥	🟩	🟩	🟩	🟥	🟥
ZI2	🟥	🟩	🟥	🟩	🟩	🟩	🟩	🟩	🟥	🟩	62,4	🟩	🟥	🟩	🟩	🟩	🟩	🟩	🟩	0,9	🟩	🟩	🟩	🟥	🟥
ZI6	🟥	-0,2	🟥	🟩	🟩	🟩	🟩	🟩	🟥	🟩	62,4	🟩	🟥	🟩	🟩	🟩	🟩	🟩	🟩	0,9	🟩	🟩	🟩	🟥	🟥
ZIB	🟥	97,5	🟥	97,5	🟥	97,4	🟥	🟥	🟥	🟥	🟥	72,2	🟥	🟥	97,5	🟥	🟥	99,3	97,5	🟥	97,5	97,5	97,5	🟥	🟥
ZID	🟥	82,0	🟥	81,7	90,5	81,1	90,9	90,5	🟥	90,5	92,1	54,4	🟥	90,5	81,7	82,3	90,5	86,0	82,2	91,3	81,7	81,7	82,3	🟥	🟥
ZIE	🟥	-0,2	🟥	99,5	🟥	🟩	🟥	🟥	🟥	🟥	🟥	🟩	🟥	🟥	80,1	🟩	🟥	🟥	99,5	🟥	99,5	99,5	99,5	🟥	🟥
ZIP	🟥	🟩	🟥	🟩	🟩	🟩	🟩	🟩	🟥	🟩	62,4	🟩	🟥	🟩	22,2	🟩	🟩	🟩	🟩	0,9	🟩	🟩	🟩	🟥	🟥
ZOO	🟥	🟥	🟥	🟩	🟥	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟩	🟥	🟥	🟥	🟥

Loss of detection (in percent):	🟩	0,0%	🟦	0,1% - 20,0%	🟧	20,1% – 99,9%	🟥	100,0%

Result matrix for the "complete" testbed ("Q")

	ANT	AVA	AVG	AVK	AVP	BDF	CMD	DRW	FIR	FPR	FSE	GLA	IKA	NAV	NVC	PAV	PER	PRO	RAV	SCN	SWP	VBR	VSP
_A_	97,2	96,1	98,0	95,0	95,7	94,9	97,0	97,1	🟥	97,1	97,1	95,0	🟥	96,8	97,7	95,7	98,2	98,2	95,2	95,9	96,8	🟥	🟥
7Z_	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
AC2	🟥	🟥	🟥	🟩	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟥
ACE	🟥	🟥	🟥	🟩	🟩	🟩	60,6	🟥	🟥	60,4	🟥	🟥	🟥	🟥	🟥	🟩	🟥	🟥	23,8	🟥	🟥	🟥	🟥
ARJ	🟥	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟥	🟩	🟥	🟥	🟥	🟥	🟩	🟩	🟥	🟩	🟩	🟥	🟩	🟥	🟥
B64	🟥	🟩	🟥	🟩	🟩	🟥	🟥	🟩	🟥	🟥	🟥	🟥	🟥	22,2	🟥	🟩	🟥	🟥	🟩	🟥	🟥	🟥	🟥
BH_	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
BZ2	🟥	🟥	🟥	🟩	🟩	47,8	🟥	🟥	🟥	🟥	🟩	🟥	🟥	🟥	🟥	🟩	🟥	🟥	0,5	🟥	🟥	🟥	🟥
CAB	🟥	0,2	🟥	🟩	🟩	🟩	🟩	🟩	🟥	🟩	🟩	🟥	🟥	22,2	🟥	🟩	🟥	🟥	🟩	🟥	🟥	🟥	🟥
CMS	🟥	🟥	🟥	🟩	4,5	47,8	🟩	🟥	🟥	🟩	🟥	🟥	🟥	22,2	🟥	5,4	🟥	🟥	🟩	🟥	46,5	🟥	🟥
GZ_	🟥	🟩	🟥	🟩	🟩	47,8	🟩	🟩	🟥	🟩	🟩	🟥	🟥	22,2	🟩	🟩	🟥	🟥	🟩	🟥	46,5	🟥	🟥
HA_	🟥	🟥	🟥	🟩	🟩	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟩	🟥	🟥	97,1	🟥	🟥	🟥	🟥
JAR	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
JAV	🟥	🟩	🟥	🟥	🟥	🟩	🟥	🟥	🟥	🟥	🟩	🟩	🟥	22,2	🟩	🟥	🟥	🟩	🟥	🟥	🟥	🟥	🟥
LHA	🟥	🟥	🟥	🟥	🟥	0,5	🟩	🟥	🟥	🟩	🟩	🟥	🟥	22,9	🟥	🟥	🟥	🟥	🟩	🟥	🟥	🟥	🟥
PAK	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
RA1	🟥	🟩	🟩	🟩	🟩	🟩	99,8	🟩	🟥	99,8	🟩	🟥	🟥	🟥	🟥	🟩	🟥	🟥	🟩	🟥	🟩	🟥	🟥
RA2	🟥	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟥	🟩	🟩	🟥	🟥	🟥	🟩	🟩	🟥	🟥	🟩	🟥	🟩	🟥	🟥
RA3	🟥	🟩	🟥	🟩	🟩	🟩	🟥	🟩	🟥	🟥	🟩	🟥	🟥	🟥	🟥	🟩	🟥	🟥	🟩	🟥	🟥	🟥	🟥
RAR	🟥	-0,2	🟥	🟩	🟩	🟩	🟥	🟩	🟥	🟥	🟩	🟥	🟥	🟥	🟥	🟩	🟥	🟥	🟩	🟥	🟥	🟥	🟥
TAR	🟥	🟩	🟥	🟩	🟩	47,8	🟩	🟩	🟥	🟩	🟩	🟥	🟥	22,2	🟩	🟩	🟥	🟥	🟩	🟥	46,5	🟥	🟥
UC2	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
UUE	🟥	🟥	🟥	🟩	🟩	47,8	🟩	🟩	🟥	🟩	🟥	🟥	🟥	22,2	🟥	🟩	🟥	🟥	🟩	🟥	46,5	🟥	🟥
ZI2	🟥	🟩	1,4	🟩	🟩	🟩	🟩	🟩	🟥	🟩	🟩	🟩	🟥	22,2	🟩	🟩	🟩	🟩	🟩	🟥	🟩	🟥	🟥
ZI6	🟥	🟩	1,4	🟩	🟩	🟩	🟩	🟩	🟥	🟩	🟩	🟩	🟥	22,2	🟩	🟩	🟩	🟩	🟩	🟥	🟩	🟥	🟥
ZIB	🟥	97,5	97,5	97,7	🟥	97,4	🟥	🟥	🟥	🟥	🟥	70,0	🟥	97,5	🟥	🟥	99,3	97,5	97,5	🟥	97,5	🟥	🟥
ZID	🟥	82,0	82,0	81,7	🟥	81,1	🟥	🟥	🟥	🟥	🟥	20,0	🟥	91,9	82,3	🟥	86,0	82,2	81,7	🟥	81,9	🟥	🟥
ZIE	🟥	🟩	99,5	99,5	🟥	🟩	🟥	🟥	🟥	🟥	🟥	🟩	🟥	88,7	🟩	🟥	🟥	99,5	99,5	🟥	99,5	🟥	🟥
ZIP	🟥	🟩	2,3	🟩	🟩	🟩	🟩	🟩	🟥	🟩	🟩	🟩	🟥	22,2	🟩	🟩	🟩	🟩	🟩	🟥	🟩	🟥	🟥

Loss of detection (in percent):	🟩	0,0%	🟦	0,1% - 20,0%	🟧	20,1% – 99,9%	🟥	100,0%

See problem list for `INO` and `QHL`.

Result matrix for the "recursive" testbed ("R")

	ANT	AVA	AVG	AVK	AVP	BDF	CMD	DRW	FIR	FPR	FSE	GLA	IKA	INO	NAV	NVC	PAV	PER	PRO	QHL	RAV	SCN	SWP	VBR	VSP
_A_	29,5	🟩	25,7	🟩	🟩	🟩	🟩	🟩	🟥	🟩	🟩	67,2	88,1	🟩	🟩	🟩	🟩	53,7	26,8	🟩	🟩	🟩	🟩	🟥	0,0
7Z_	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
AC2	🟥	🟥	🟥	🟩	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟩	🟥	🟥	🟩	95,5	🟥	🟥	🟥	🟥
ACE	🟥	🟥	🟥	🟩	🟩	🟩	🟩	🟥	🟥	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟩	🟥	🟥	🟩	36,0	🟩	🟥	🟥	🟥
ARC	🟥	🟥	🟥	32,4	🟥	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟩	🟥	🟥	🟥	🟥	33,0	🟩	🟥	🟥	🟥
ARJ	🟥	0,2	🟩	🟩	🟩	🟩	🟩	🟩	🟥	🟩	🟥	🟥	🟥	🟩	🟥	🟩	🟩	🟥	🟩	🟩	41,0	🟩	🟩	🟥	🟥
B64	🟥	🟩	🟥	🟩	🟩	🟥	🟥	🟩	🟥	🟥	57,7	🟥	34,0	🟥	🟩	🟥	🟥	🟥	🟥	🟥	🟩	🟩	🟥	🟥	🟥
BH_	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
BZ2	🟥	🟥	🟥	🟩	🟩	🟩	🟥	🟥	🟥	🟥	57,9	🟥	🟥	🟥	🟥	🟥	🟩	🟥	🟥	🟥	2,9	🟩	🟥	🟥	🟥
CAB	🟥	🟥	🟥	🟩	🟩	🟩	🟩	🟩	🟥	🟩	63,3	🟥	🟥	🟩	🟩	🟥	🟩	🟥	🟩	🟩	🟩	🟩	🟥	🟥	🟥
CMS	🟥	🟥	🟥	🟩	🟩	🟩	1,1	🟥	🟥	1,6	🟥	🟥	🟥	🟥	🟩	🟥	🟩	🟥	🟥	99,8	🟩	🟥	🟩	🟥	🟥
GZ_	99,8	🟩	🟥	🟩	🟩	🟩	🟩	🟩	🟥	🟩	57,9	🟥	🟥	4,8	🟩	🟩	🟩	🟥	🟥	🟥	🟩	🟩	🟩	🟥	0,0
HA_	🟥	🟥	🟥	🟩	🟩	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟩	🟥	🟥	🟥	69,2	🟥	🟥	🟥	🟥
JAR	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
JAV	🟥	🟩	🟥	🟥	🟥	🟩	🟥	🟥	🟥	🟥	63,3	🟥	🟥	🟩	60,0	🟩	🟥	🟥	🟩	🟩	🟥	🟥	🟥	🟥	🟥
LHA	🟥	🟥	🟥	🟩	🟩	🟩	🟩	🟥	🟥	🟩	63,3	🟥	🟥	16,3	6,1	🟥	18,6	🟥	🟥	🟥	3,6	🟩	🟩	🟥	🟥
PAK	🟥	🟥	🟥	🟥	🟥	97,9	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
RA1	🟥	🟩	🟩	🟩	🟩	🟩	97,3	🟩	🟥	97,5	63,6	🟥	🟥	🟥	🟥	🟥	🟩	🟥	🟥	🟩	🟩	🟩	🟩	🟥	🟥
RA2	🟥	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟥	🟩	63,6	🟥	🟥	🟩	🟥	🟩	🟩	🟥	🟥	🟩	🟩	🟩	🟩	🟥	🟥
RA3	🟥	🟩	🟥	🟩	🟩	🟩	🟥	🟩	🟥	🟥	63,6	🟥	🟥	🟥	🟥	🟥	🟩	🟥	🟥	🟩	🟩	🟥	🟥	🟥	🟥
RAR	🟥	🟩	🟥	🟩	🟩	🟩	🟥	🟩	🟥	🟥	63,6	🟥	🟥	🟥	🟥	🟥	🟩	🟥	🟥	🟩	🟩	🟥	🟥	🟥	🟥
SHA	🟥	🟥	🟥	🟥	55,2	🟥	89,5	🟥	🟥	89,4	🟥	🟥	🟥	47,7	🟥	56,2	🟥	🟥	🟥	🟥	🟥	🟥	47,8	🟥	🟥
SQZ	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
TAR	🟥	🟩	🟥	🟩	🟩	12,4	🟩	🟩	🟥	🟩	63,6	🟥	🟥	🟥	🟩	🟩	🟩	🟥	🟥	🟥	🟩	🟩	2,7	🟥	🟥
UC2	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
UUE	🟥	🟥	🟥	🟩	🟩	🟩	🟩	🟩	🟥	🟩	57,7	🟥	🟥	5,2	🟩	🟥	🟩	🟥	🟥	🟥	🟩	🟩	🟩	🟥	🟥
ZI2	🟥	-0,2	2,3	🟩	🟩	🟩	🟩	🟩	🟥	🟩	63,3	🟥	🟥	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟥	🟥
ZI6	🟥	0,5	5,5	🟩	🟩	🟩	🟩	🟩	🟥	🟩	63,3	🟥	🟥	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟥	🟥
ZIB	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
ZID	🟥	82,0	82,0	81,7	90,5	81,1	90,9	90,5	🟥	90,5	92,1	🟥	🟥	90,5	81,7	82,3	90,5	86,0	82,2	90,4	81,7	81,7	82,3	🟥	🟥
ZIE	🟥	0,5	🟥	🟥	🟥	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	98,0	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
ZIP	🟥	🟩	2,3	🟩	🟩	🟩	🟩	🟩	🟥	🟩	63,6	🟥	🟥	🟩	22,2	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟥	🟥
ZOO	🟥	🟥	🟥	🟩	🟥	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟩	🟥	🟥	🟥	🟥

Loss of detection (in percent):	🟩	0,0%	🟦	0,1% - 20,0%	🟧	20,1% – 99,9%	🟥	100,0%

Result matrix for the "deep recursive" testbed ("R2")

	ANT	AVA	AVG	AVK	AVP	BDF	CMD	DRW	FIR	FPR	FSE	GLA	IKA	INO	NAV	NVC	PAV	PER	PRO	QHL	RAV	SCN	SWP	VBR	VSP
_A_	29,7	🟩	29,3	🟩	🟩	🟩	🟩	🟩	🟥	🟩	69,2	67,2	98,8	🟩	🟩	🟩	🟩	53,7	98,6	🟥	🟩	🟩	🟩	🟥	0,0
7Z_	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
AC2	🟥	🟥	🟥	🟩	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
ACE	🟥	🟥	🟥	🟩	🟩	🟩	🟩	🟥	🟥	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟩	🟥	🟥	🟥	🟥	🟩	🟥	🟥	🟥
ARC	🟥	🟥	🟥	32,8	🟥	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟩	🟥	🟥	🟥	🟥	33,0	🟩	🟥	🟥	🟥
ARJ	🟥	0,2	🟩	🟩	🟩	🟩	🟩	🟩	🟥	🟩	🟥	🟥	🟥	🟩	🟥	🟩	🟩	🟥	94,7	🟥	🟥	🟩	🟩	🟥	🟥
B64	🟥	🟩	🟥	🟩	🟩	🟥	🟥	🟩	🟥	🟥	32,1	🟥	🟥	🟥	🟩	🟥	🟥	🟥	🟥	🟥	🟩	🟩	🟥	🟥	🟥
BH_	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
BZ2	🟥	🟥	🟥	🟩	🟩	🟩	🟥	🟥	🟥	🟥	🟥	🟥	99,8	🟥	🟥	🟥	🟩	🟥	🟥	🟥	2,9	0,5	🟥	🟥	🟥
CAB	🟥	🟥	🟥	🟩	🟩	🟩	🟩	🟩	🟥	🟩	🟥	🟥	🟥	🟩	🟩	🟥	🟩	🟥	🟥	🟥	🟩	🟩	🟥	🟥	🟥
CMS	🟥	🟥	🟥	🟩	🟩	🟩	4,6	🟥	🟥	4,3	🟥	🟥	🟥	🟥	🟩	🟥	🟩	🟥	🟥	🟥	0,5	🟥	🟩	🟥	🟥
GZ_	99,8	🟩	🟥	🟩	🟩	🟩	🟩	🟩	🟥	🟩	🟥	🟥	🟥	4,8	🟩	16,8	🟩	🟥	🟥	🟥	🟩	🟩	🟩	🟥	🟥
HA_	🟥	🟥	🟥	🟩	🟩	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟩	🟥	🟥	🟥	69,2	🟥	🟥	🟥	🟥
JAR	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
JAV	🟥	0,2	🟥	🟥	🟥	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟩	60,0	18,2	🟥	🟥	97,0	🟥	🟥	🟥	🟥	🟥	🟥
LHA	🟥	🟥	🟥	🟩	🟩	5,8	🟩	🟥	🟥	🟩	🟥	🟥	🟥	16,3	6,1	🟥	18,6	🟥	🟥	🟥	4,8	🟩	🟩	🟥	🟥
PAK	🟥	🟥	🟥	🟥	🟥	97,9	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
RA1	🟥	🟩	🟩	🟩	🟩	🟩	97,7	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟩	🟥	🟥	🟥	🟩	🟩	🟩	🟥	🟥
RA2	🟥	3,0	🟩	🟩	🟩	🟩	🟩	🟩	🟥	🟥	🟥	🟥	🟥	🟩	🟥	🟩	🟩	🟥	🟥	🟥	🟩	🟩	🟩	🟥	🟥
RA3	🟥	-0,2	🟥	🟩	🟩	🟩	🟥	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟩	🟥	🟥	🟥	🟩	🟥	🟥	🟥	🟥
RAR	🟥	-0,2	🟥	🟩	🟩	🟩	🟥	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟩	🟥	🟥	🟥	🟩	🟥	🟥	🟥	🟥
SHA	🟥	🟥	🟥	🟥	55,2	🟥	91,1	🟥	🟥	🟥	🟥	🟥	🟥	47,7	🟥	56,2	🟥	🟥	🟥	🟥	🟥	🟥	47,8	🟥	🟥
SQZ	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
TAR	🟥	🟩	🟥	🟩	🟩	12,4	🟩	🟩	🟥	🟩	🟥	🟥	🟥	🟥	🟩	🟩	🟥	🟥	🟥	🟥	🟩	🟩	2,7	🟥	🟥
UC2	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
UUE	🟥	🟥	🟥	🟩	🟩	🟩	🟩	🟩	🟥	🟩	32,1	🟥	🟥	5,4	🟩	🟥	🟥	🟥	🟥	🟥	🟩	🟩	🟩	🟥	🟥
ZI2	🟥	-0,2	8,4	🟩	🟩	🟩	🟩	🟩	🟥	🟥	🟥	🟥	🟥	🟩	🟩	4,6	🟩	🟩	97,0	🟥	🟩	🟩	🟩	🟥	🟥
ZI6	🟥	0,5	4,5	🟩	🟩	🟩	🟩	🟩	🟥	🟥	🟥	🟥	🟥	🟩	🟩	4,6	🟩	🟩	🟥	🟥	🟩	🟩	🟩	🟥	🟥
ZIB	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
ZID	🟥	82,0	82,0	81,7	90,5	81,1	90,9	90,5	🟥	🟥	🟥	🟥	🟥	90,5	81,7	82,5	90,5	86,0	🟥	🟥	81,7	81,7	82,3	🟥	🟥
ZIE	🟥	🟩	🟥	🟥	🟥	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	98,0	4,6	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
ZIP	🟥	🟩	2,3	🟩	🟩	🟩	🟩	🟩	🟥	🟩	🟥	🟥	🟥	🟩	22,2	🟩	🟩	🟩	97,0	🟥	🟩	🟩	🟩	🟥	🟥
ZOO	🟥	🟥	🟥	🟩	🟥	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟩	🟥	🟥	🟥	🟥

Loss of detection (in percent):	🟩	0,0%	🟦	0,1% - 20,0%	🟧	20,1% – 99,9%	🟥	100,0%

Result matrix for the "self-extracting" testbed ("S")

	ANT	AVA	AVG	AVK	AVP	BDF	CMD	DRW	FIR	FPR	FSE	GLA	IKA	INO	NAV	NVC	PAV	PER	PRO	QHL	RAV	SCN	SWP	VBR	VS D
_A_	70,3	39,3	49,5	18,1	18,1	🟩	74,3	29,4	🟥	74,4	47,5	91,1	92,8	98,4	67,6	88,5	18,1	🟥	🟥	76,9	77,4	38,0	65,3	🟥	🟥
7Z_	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
AC2	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	0,9	🟥	🟥	🟥	🟥	🟥
ACE	🟥	🟥	🟥	🟥	🟥	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	0,9	🟥	🟩	🟥	🟥	🟥
ARJ	🟥	-0,2	🟥	🟩	🟩	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟩	🟩	🟥	🟥	🟥	🟥	🟥	🟩	🟥	🟥
PAK	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
RA1	🟥	-0,2	🟩	🟩	🟩	🟩	🟥	🟩	🟥	🟥	59,7	🟥	🟥	🟥	🟥	🟥	🟩	🟥	🟥	🟥	🟥	🟩	🟩	🟥	🟥
RA2	🟥	0,2	🟩	🟩	🟩	🟩	🟥	🟩	🟥	🟥	59,7	🟥	🟥	🟥	🟥	🟥	🟩	🟥	🟥	🟥	🟥	🟩	🟩	🟥	🟥
RA3	🟥	🟩	🟥	🟩	🟩	🟩	🟥	🟩	🟥	🟥	59,7	🟥	🟥	🟥	🟥	🟥	🟩	🟥	🟥	🟥	🟩	🟥	🟥	🟥	🟥
RAR	🟥	-0,2	🟥	🟩	🟩	🟩	🟥	🟩	🟥	🟥	59,7	🟥	🟥	🟥	🟥	🟥	🟩	🟥	🟥	🟥	🟩	🟥	🟥	🟥	🟥
ZI2	🟥	🟥	1,4	🟩	🟩	🟩	🟩	🟩	🟥	🟩	55,9	🟥	🟥	🟥	🟩	🟥	🟩	🟥	🟥	🟥	🟥	🟩	🟥	🟥	🟥
ZI6	🟥	🟥	1,4	🟩	🟩	🟩	🟩	🟩	🟥	🟩	55,9	🟥	🟥	🟥	🟩	🟥	🟩	🟥	🟥	🟥	🟥	🟩	🟥	🟥	🟥
ZIB	🟥	🟥	97,5	🟥	🟥	97,4	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	97,5	🟥	🟥	🟥	🟥	🟥	🟥	97,5	🟥	🟥	🟥
ZID	🟥	🟥	82,0	🟥	🟥	81,1	90,7	90,5	🟥	90,3	90,5	🟥	🟥	🟥	81,7	🟥	🟥	🟥	🟥	🟥	🟥	81,7	🟥	🟥	🟥
ZIE	🟥	🟥	99,5	🟥	🟥	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	80,1	🟥	🟥	🟥	🟥	🟥	🟥	99,5	🟥	🟥	🟥

Loss of detection (in percent): 🟩 0,0% 🟦 0,1% - 20,0% 🟧 20,1% – 99,9% 🟥 100,0%

Result matrix for the "encrypted" testbed ("E")

	ANT	AVA	AVG	AVK	AVP	BDF	CMD	DRW	FIR	FPR	FSE	GLA	IKA	INO	NAV	NVC	PAV	PER	PRO	QHL	RAV	SCN	SWP	VBR	VS P
_A_	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
7Z_	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
AC2	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
ACE	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
ARC	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
ARJ	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
BH_	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
JAR	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
PAK	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
RA1	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
RA2	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
RA3	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
RAR	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
ZI2	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
ZI6	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
ZIB	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
ZID	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
ZIE	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
ZIP	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥

Loss of detection (in percent):	🟩	0,0%	🟦	0,1% - 20,0%	🟧	20,1% – 99,9%	🟥	100,0%

Detection rates for the reference testbed

Product	detected Viruses		unreliable identification		unreliable detection		detected Files	
ANT	50	100,0%	11	22,0%	5	10,0%	431	97,5%
AVA	50	100,0%	8	16,0%	3	6,0%	438	99,1%
AVG	50	100,0%	12	24,0%	1	2,0%	440	99,5%
AVK	50	100,0%	6	12,0%	0	0,0%	442	100,0%
AVP	50	100,0%	6	12,0%	0	0,0%	442	100,0%
BDF	50	100,0%	7	14,0%	2	4,0%	429	97,1%
CMD	50	100,0%	4	8,0%	3	6,0%	439	99,3%
DRW	50	100,0%	4	8,0%	0	0,0%	442	100,0%
FIR	48	96,0%	1	2,0%	13	26,0%	349	79,0%
FPR	50	100,0%	4	8,0%	0	0,0%	442	100,0%
FSE	50	100,0%	7	14,0%	0	0,0%	442	100,0%
GLA	33	66,0%	1	2,0%	7	14,0%	180	40,7%
IKA	50	100,0%	5	10,0%	6	12,0%	429	97,1%
INO	50	100,0%	6	12,0%	0	0,0%	442	100,0%
MR2	-[*]	-[*]	-[*]	-[*]	-[*]	-[*]	-[*]	-[*]
NAV	50	100,0%	11	22,0%	0	0,0%	442	100,0%
NVC	50	100,0%	6	12,0%	3	6,0%	434	98,2%
PAV	50	100,0%	8	16,0%	0	0,0%	442	100,0%
PER	39	78,0%	2	4,0%	9	18,0%	285	64,5%
PRO	50	100,0%	6	12,0%	3	6,0%	437	98,9%
QHL	49	98,0%	2	4,0%	6	12,0%	425	96,2%
RAV	50	100,0%	7	14,0%	0	0,0%	442	100,0%
SCN	50	100,0%	5	10,0%	0	0,0%	442	100,0%
SWP	50	100,0%	5	10,0%	1	2,0%	441	99,8%
VBR	43	86,0%	5	10,0%	13	26,0%	352	79,6%
VSP	5	10,0%	1	2,0%	1	2,0%	120	27,1%

These results show clearly that most products have similar detection rates for the reference testbed. Therefore the results on the compressed testbeds will be comparable for all major anti-malware products without significant dependencies on the selected malware set.

Only some of the not well-known anti-malware products (FIR, GLA, MR2, PER, VBR, VSP) have detection rates below 95% while the majority of products (14) achieves more than 99% detection rate (and therefore differs by less than 1%).

[*] No results - see problem list

Appendix

A Problems observed during the test

A.1 List of postscans

In several cases the tested products did not access and/or scan all files in the testbeds. This is possibly due to the "FF/FN anomaly" (as reported in previous tests) or due to crashes or other product misbehaviour (as reported in the problem list below). In such cases, up to 2 "postscans" were started (wherever possible on the remainder of the related testbed), and the test results are computed from the union of these scan attempts.

The following list summarizes those products where at least 1 postscan was initialized on a specific testbed:

Testbed	*Products with postscans*
FI (reference)	FIR,GLA,PER,PRO,QHL,VBR
P (standard)	AVA,AVG,AVK,CMD,FIR,FPR,FSE(2x), GLA,INO,NAV,PAV,PRO,QHL,RAV
P2 (obfuscated)	AVA,AVK,FIR,FPR,FSE(2x),GLA,INO,NAV,NVC,PRO,QHL,RAV
Q (complete)	AVG,AVK,AVP(2x),BDF,FIR,FPR,FSE(2x),NAV,PAV, PRO(2x),RAV
R (2x recursive)	AVA,AVG,AVK,BDF,DRW,FIR,FPR,FSE(2x)GLA,INO,NAV,PAV, PRO,QHL,RAV
R2 (9x recursive)	AVA,AVG,AVK,BDF,DRW,FIR,FPR,FSE(2x),GLA,INO,NAV, NVC(2x),PAV,PRO(2x),QHL,RAV,SCN(2x)
S (self-extracting)	AVG,AVK,FIR,FSE(2x),GLA,NAV,QHL
E (encrypted)	AVK,BDF,FIR,FPR,GLA,NAV,QHL,SCN,FSE(2x)

A.2 List of product specific problems

All product specific problems observed during the test are documented here.

ANT	- this product failed to report any archive contents making it impossible to produce any detailed results for single compression formats
AVK	- scanning of HA archives is very slow
AVP	- scanning of the MS Compress (CP_) archive on the "complete compressed" testbed (P:\CP_.CP_) aborted with "I/O Error"
BDF	- filenames in SHA archives are always truncated making it impossible to evaluate them (we counted only the archives themselves)
CMD	- some archive types were never reported or scanned
FPR	- very long archive contents (as in the "deep recursive compressed" testbed) were not completely reported, but cut to a maximum length leaving only the beginning of the path and the last filename - some archive types are never reported or scanned

FSE This product had severe problems to execute properly:

- the scanner process hangs after the scan seems to be finished if using the /REPORT option
- logfiles created by shell direction seems to be incomplete
- on all compressed testbeds the scanner reported:
  `"Scanning of ... was aborted [F-Secure F-PROT]"`
  for some files. All remaining files in the testbed were not scanned but instead lots of
  `"Cannot open file"` errors were reported.
- scanning of the "deep recursive compressed" testbed ended with:
  `"Error: Unknown error"`

GLA

- some archive contents were reported only with a temporary filename, which (besides making no sense at all) makes it impossible to count these samples. Example:
  `"UPX Runtime packed: E:\Program Files\Gladiator Scanner\TEMP\EXA_004_.EXE"`

INO

- this product fails to report paths inside archives making it impossible to produce any results for the "complete compressed" testbed

MR2

- scanning of "File-ITW" and "self-extracting archive" testbeds failed with the error message:

```
     -=[ Fatal Error ]=-
GetSigVirusName: Signature file not found!
VBS/Signature File = virscan.trj
Version needed     = Version 1.20
ERROR: Can not write to LOG file: mr2ssub2.rep
!!! Fatal: Mr2S I/O-Error !!!
```

  This made it impossible to get any detection loss values.

NAV

- this product doesn't report all scanned files, so we can't ensure that all files were really scanned
- some files were detected as infected but could not be counted as `NAV` failed to report the complete path.
  Although there is some evidence that `NAV` in fact detected all samples (100%) in the archives B64, CAB, CP_, GZ_, JAR, LZH, TAR, UUE, ZI2, ZI6, ZIP this can't be confirmed due to missing filenames in the logfile.

NVC

- aborted scanning of the "deep recursive compressed" testbed with: `"Internal error"`

PER

- this product doesn't report all scanned files, so we can't ensure that all files were really scanned

PRO

- this product doesn't report all scanned files, so we can't ensure that all files were really scanned
- crashed three times on the "deep recursive compressed" testbed, probably due to a memory leak. Error message: `"The system is low on virtual memory"`

QHL

- this product doesn't report all scanned files, so we can't ensure that all files were really scanned
- this product fails to report paths inside archives making it impossible to produce any results for the 'Complete Packed' testbed

RAV

- scanning of HA archives is very slow

SCN

- this product fails to report paths inside archives making it impossible to produce detailed results for single compression formats on the "complete compressed" testbed
- crashed two times on the file:
  `R:\MALW\MROW\MRONON\E\PIZEROLP.XE\MROW\594021\BZ2.BZ2\EXA_000_.EXE`

VBR

- this product doesn't report all scanned files, so we can't ensure that all files were really scanned

B Additional details

Due to the huge amount of data, the following test details are not included in this report but can be retrieved from our ftp server.

B.1 Product / Vendor Details

All product version and configuration details as well as vendor contact information are collected in the file: ftp://agn-www.informatik.uni-hamburg.de/pub/texts/tests/pc-av/packtest/a2scanls.txt

B.2 All detection rate result tables

The result matrices presented in chapter 4 were obtained from many single detection results that are accessible at:
ftp://agn-www.informatik.uni-hamburg.de/pub/texts/tests/pc-av/packtest/result/

B.3 All product logfiles

All logfiles produced by products during this test can be obtained from:
ftp://agn-www.informatik.uni-hamburg.de/pub/texts/tests/pc-av/packtest/logs/
These logfiles are the complete base for all test results.