# BENCHMARK REPORT

CA & HPA LAB

October 17, 2015

CA&HPA

# Introduction

Benchmark is an open-source, little-size test set to measure the effectiveness of antivirus softwares, which is provided to be downloaded and tested by any computer users. Benchmark test set usually includes hundreds of malwares which were selected from different clusters of a clustering model. This model is built as three layer: behavior extraction, similarity measurement and clustering. Benchmark is also a project which involves multi-domain theories, such as software's run-time analysis, graphical theory and data mining.

The Benchmark we show in this report is one Benchmark whose objects are antivirus softwares in Win32. All samples in this test set are executable files. We separate three layers in the clustering model into five steps as follow:

(1) Prepare: material collection and the build of database.
(2) Behavior analysis: an automatic extraction system of malware's behaviors.
(3) Similarity measurement: one method to get a value of the behaviors' similarity between two different malwares.
(4) Clustering and selection: a new designed clustering algorithm based on graph theory.
(5) Modifying and enlarging: modify the algorithms and enlarge the original set.

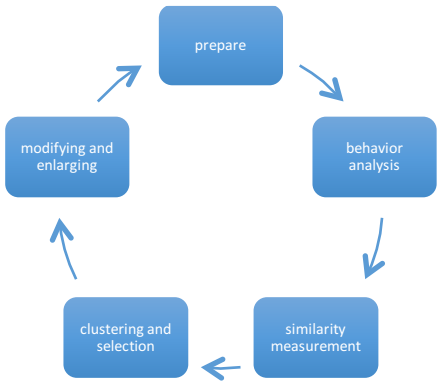



figure 1 map for the process of Benchmark's generation

# Architecture

## Behavior analysis

This layer is required to build a automatic extraction system to analyze thousands of malwares. The automatic extraction system works on virtual machine technology and Dr. Memory tool set. Dr. Memory

is an open-source tool set to monitor software's run-time memory, but what we focus on is one tool called drstrace.exe of Dr. Memory. Drstrace.exe can trace one software run-time behaviors then record its system calls into a log file. The flow of the system is as figure 2.

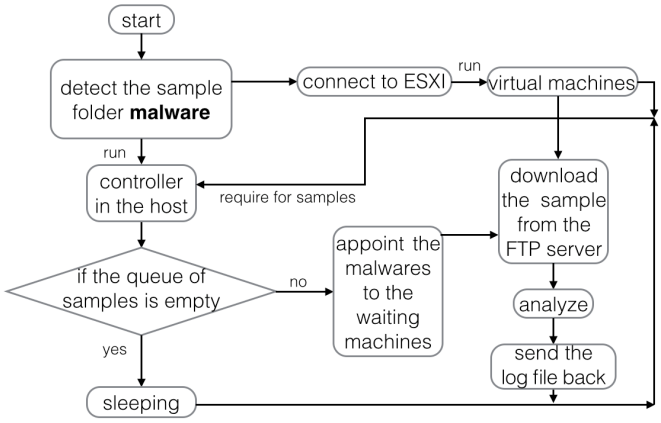



figure 2 the automatic extraction system's workflow map

When the system works, virtual machines will be turned on first and then ask the monitor for samples. The monitor program will accept the require then send a sample from SFTP server to the corresponding virtual machine. After extraction the virtual machines will send the log files back to SFTP server and tell the monitor. At this time, the monitor will revert the snapshot of the corresponding virtual machine.

Then we need to analyze the log files and change these text information into sequences, which can be operated and calculated by computer. We divided all system calls into 23 groups as *table 1* then encode every system call. The kilobit and the hundreds show the type, while the decade and the unit show the number of this system call in this type. For example, *NtCreateNamedPipeFile* is the second one in type *Processes and Threads* numbered 11, so the number of *NtCreateNamedPipeFile* is 1102.

table 1 Windows system call type list

| Number | Type | Number | Type |
|---|---|---|---|
| 0 | Special Files | 13 | Atoms |
| 1 | Drivers | 14 | Error Handling |
| 2 | Processor and Bus | 15 | Execution Environment |
| 3 | Debuging and Profiling | 16 | Timer and System Time |
| 4 | Channels | 17 | Synchronization |
| 5 | Power | 18 | Memory |
| 6 | Plug and Play | 19 | File and General I/O |
| 7 | Objects | 20 | Miscellaneous |
| 8 | Registry | 21 | Jobs |
| 9 | LocalProcedure Call | 22 | IA64 Boot |
| 10 | Security | 23 | EFI Drivers |
| 11 | Processes and Threads | | |

By this way, we can clearly see what one malware does in a line chart as figure 3.

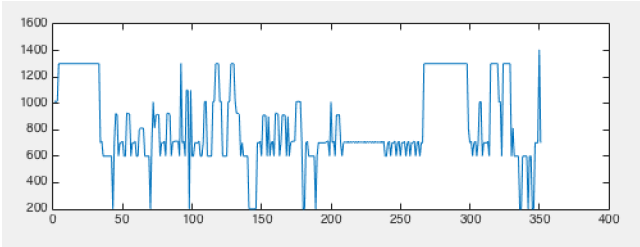



figure 3 the behaviors' line chart of Sample *Trojan.Win32.Aditer.b.exe*

## Similarity Measurement

After encoding the system call, we can describe one malware by a long sequence. This layer is to measure the similarity between two different malwares' sequence. In biological, text-analyzing and other applications, we generally want to compare two different sequences. We formalize this problem as the longest-common-subsequence problem. Especially in biological applications, scientists measure the similarity of the DNA of two different organisms with the length of their longest common subsequence. Malwares' system-call sequences can be see as their DNA learning from biology.

Using dynamic programming is a effect method to solve LCS problem. Because of the optimal substructure of LCS problem, we can give the equation as:

$$c[i,j]=\begin{cases}0, if(i=0) or (j=0)\\ c[i-1,j-1]+1, if(i,j>0) and (x_i=y_j)\\ \max(c[i,j-1],c[i-1,j]), if(i,j>0) and (x_i\neq y_j)\end{cases}$$

Given the sequences X and Y, and their length is m and n. The LCS problem using dynamic programming has m*n subproblems and each subproblem is computed in O(1) time. So the time complexity is O(m*n) at all.

Now we use SVD algorithm in calculating the matrix of malwares' distance. Following matrix is the similarity matrix of the set, which contains n malware samples:

$$\begin{pmatrix} \cdots & i_1 & \cdots & j_1 & \cdots \\ \cdots & i_2 & \cdots & j_1 & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & i_{n-1} & \cdots & j_{n-1} & \cdots \\ \cdots & i_n & \cdots & j_n & \cdots \end{pmatrix}$$

The value in this matrix, like c[x,y], shows the similarity of malware x and malware y.

So we can list the distance expression of two given malwares, I and J:

$$d_{ij} = |i_1 - j_1| + |i_2 - j_2| + \ldots + |i_{n-1} - j_{n-1}| + |i_n - j_n|$$

$d_{ij}$ is one element of the distance matrix. In the topological graph, $d_{ij}$ is the value of the edge which connects to sample I and sample J.

## Clustering

Clustering algorithm is one important part of data-mining application. Different from classification, clustering work without machine learning and cluster the similar samples from the whole set together into one subset.

In this paper we would show you a new-designed clustering algorithm based on graph which is easy to be implemented and fast to compute. It is a multi-level clustering algorithm, from bottom to top, as well. This algorithm can be separated into five steps:

(1) Construct a graph which contains all samples in the set. Every point in this graph implies one malware. The values of edges has been in the distance matrix.

(2) Input a value K as an expectation and a value M as a maximum threshold value. K implies the number of clusters we expect to, and M implies the maximum distance between every two points in one cluster. While clustering, it's not allowed if a point want to join one cluster but there exists a distance, which is larger than M, between this point and another point in that cluster.

(3) Ordering all edges by values. It's similar to Kruskal algorithm but we don't aim at generate a tree but K clusters or sets.

(4) Connect the edges ordered by their values. Make sure that every point has a distance, shorter than M, to any other point in the same cluster. Repeated this step until there does not exist any edge can be allowed to connected.

(5) After the loop of Step 4 has been break, there may exists two situations: the first one is that the number of the clusters generated is K and the algorithm is accomplished. The second situation is that the number of the clusters is larger than K. When in Situation 2, we can make the maximum threshold value M larger in an acceptable range and continue the loop of Step 4.

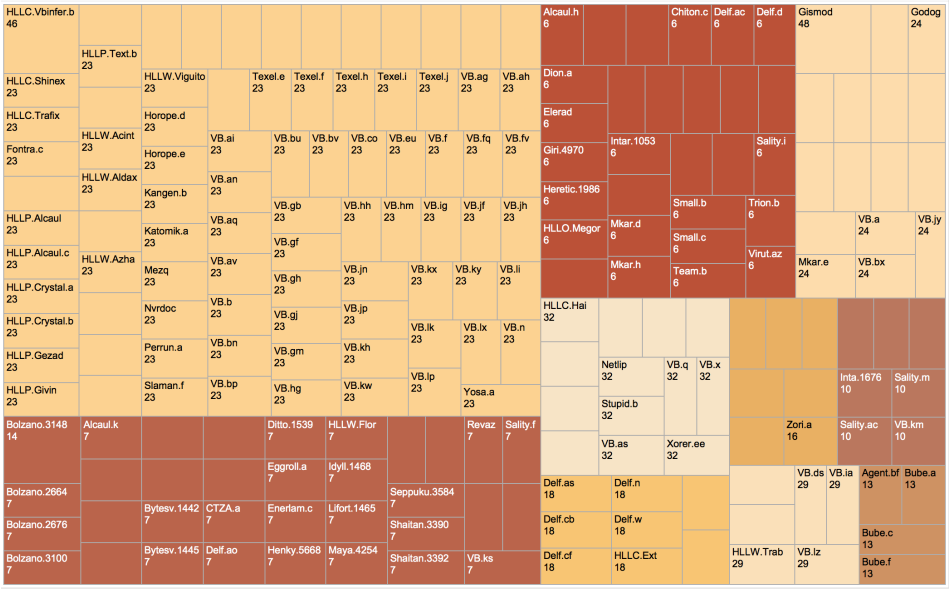



figure 5 the map of the clustering result

# Application And Test

Date: May 27th, 2015

Configuration:

table 2 test host configuration information

| Configuration type | Hardware information |
|---|---|
| Virtual machine software | VMware 11 |
| CPU | Inter(R) Core(TM) i5-4288U @ 2.6GHz |
| Memory | 512MB |
| Operating system | Mircosoft Windows XP Professional Service Pack 3 (2002) |

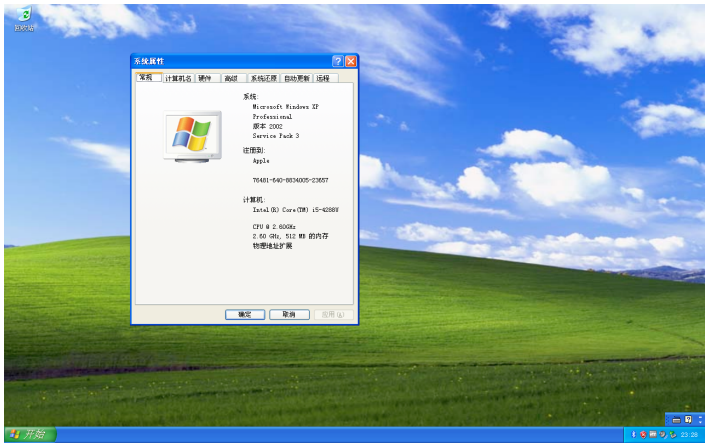
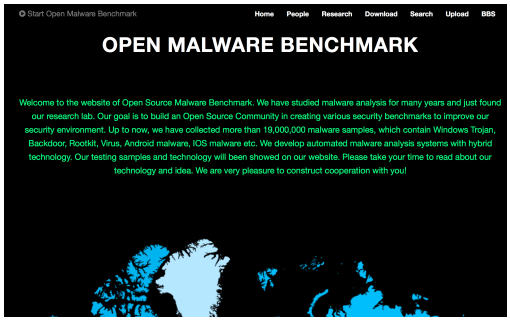


figure 6 the environment of the test is pure Windows XP SP3



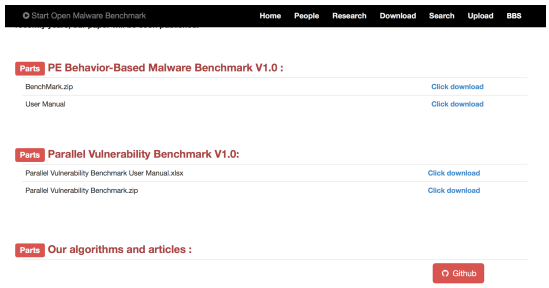




figure 7 download Benchmark test set from the website *malwarebenchmark.org*

Antivirus Software information:

table 3 nine antivirus softwares' version information

| Antivirus Software Name | Version Number |
|---|---|
| Qihoo 360 | 5.0.0.5104 |
| Baidu | 4.1.0.7423 |
| Jinsha | SP1.8.052500 |
| Kaspersky LAB | KIS15.0.0.463zh-Hans-CN |
| Rising Internet Security | 24.00.42.31 |
| Tencent QQ PC Manager | Qqpcmgr_v10.9.16338.220_1328_Normal.1431075803 |
| Avira | rj_vr8559 |
| ESET | 8.0.312.3 |
| AVG | 2015.0.5961 |

（notes: every software settings are default）

Index of the test: killing rate and killing time

Notes:

Except for the same operating system environment, we take the following measures to guarantee the fairness:

(1) We make a snapshot after a new and clean XP operating system is logined. Before we start to a new test, we will begin at this snapshot.

(2) After open the snapshot we made before, we will copy a package of the tested antivirus software, then copy the package of Benchmark as well. We won't download or install any other software except the tested software.

(3) Before we begin the test, we will guarantee that the network flow.

(4) We will guarantee that every tested antivirus software is updated to the newest version as well as the database.

(5) Every software's settings are default because Benchmark is face to all the computer user and most of them won't change the settings.

(6) When we begin, we won't do anything more until the test is over.

## Result:

We list some of antivirus softwares test:

## Qihoo 360

We update the 360 antivirus software and its database to the latest version.

The main program version is 5.0.0.5104.

figure 8 update Qihoo 360's product to the latest version

figure 9 right click Benchmark and scan it by the antivirus software

Just as figure 9 shows, what the user should do is download and unzip Benchmark, then right click Benchmark and choose to scan. The result will be showed by the antivirus software itself.

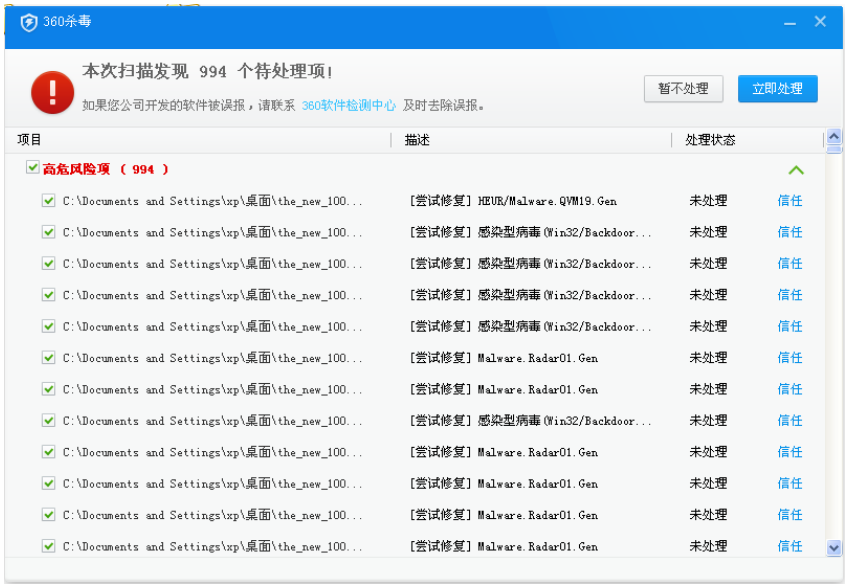



figure 10 the test result of Qihoo 360's product

The killing rate of Qihoo 360's product is 99.4%.

The killing time of Qihoo 360's product is 151 seconds.

Qihoo 360 do the best in killing rate in this serial test. And the time it costs is acceptable.

## Rising Internet Security

Download, install and update Rising antivirus software.

figure 11 update Rising's product to the latest version

Right click Benchmark and choose to scan Benchmark by Rising antivirus software.

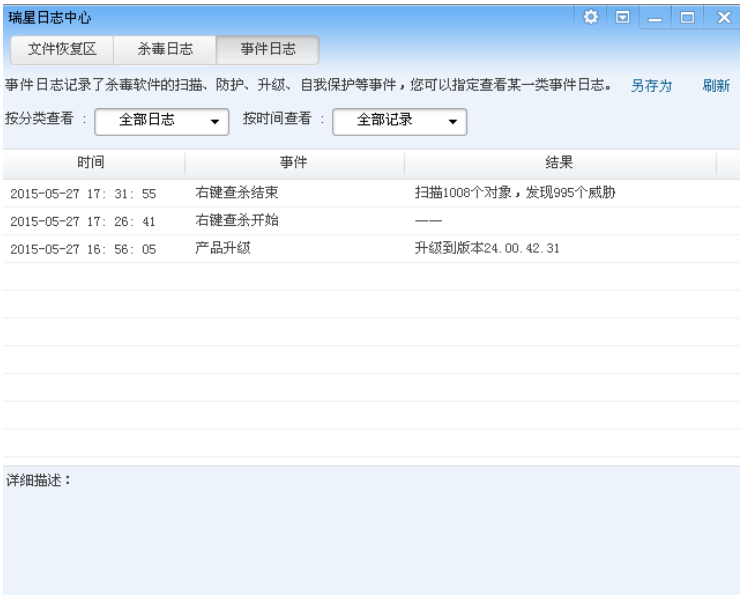



figure 12 the test result of Rising's product

Rising's product rank at second position in killing rate test. Because different principles of different softwares, the sum of files in Benchmark folder is showed differently. We use the the sum that the software showing itself as the real sum when recording.

Rising's product killing rate is 98.7%.
Its killing time is 240 seconds, which is a little slower than other products.

## AVG

We install AVG Internet Security version as the test object.

figure 13 update AVG from free version to Internet Security version

figure 14 corresponding information of AVG Internet Security product

we can find in figure 14 that AVG security Toolbar can't be used in this version of operating system.

figure 15 the test result of AVG Internet Security

AVG is at the third place in killing rate test. Its killing rate is 96.1%.

AVG's killing time is 73 seconds, which is at the second position in killing time test.

## Avira Antivirus

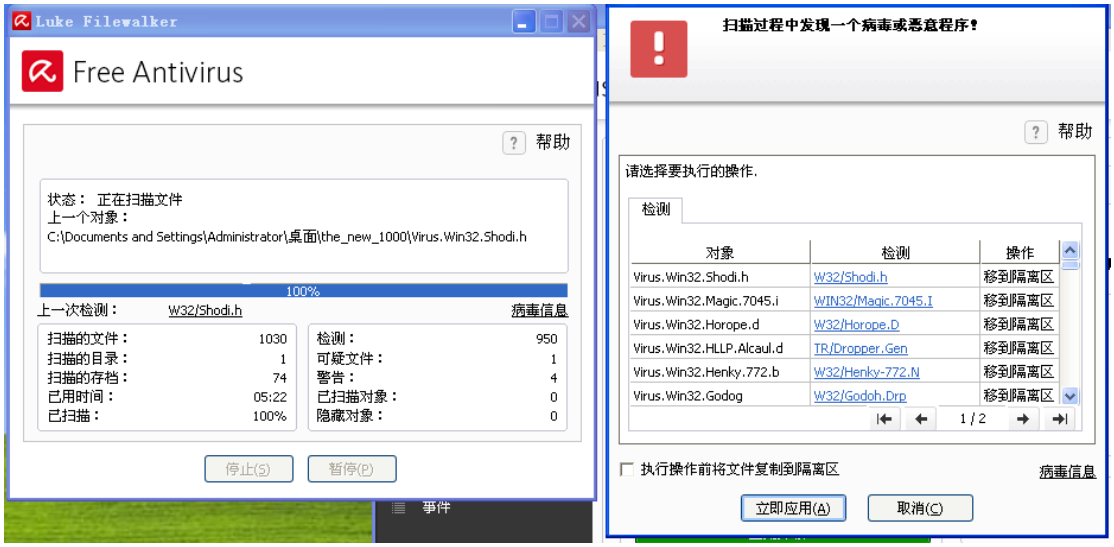



figure 16 Avira Antivirus test result

Avira's product do not bad in killing rate test as 92.2%.

It cost 322 seconds to finish killing time test, which is too long.

## Baidu

Install and update Baidu Antivirus Software. Version of the software is 4.1.0.7423.

figure 17 version information of Baidu Antivirus Software

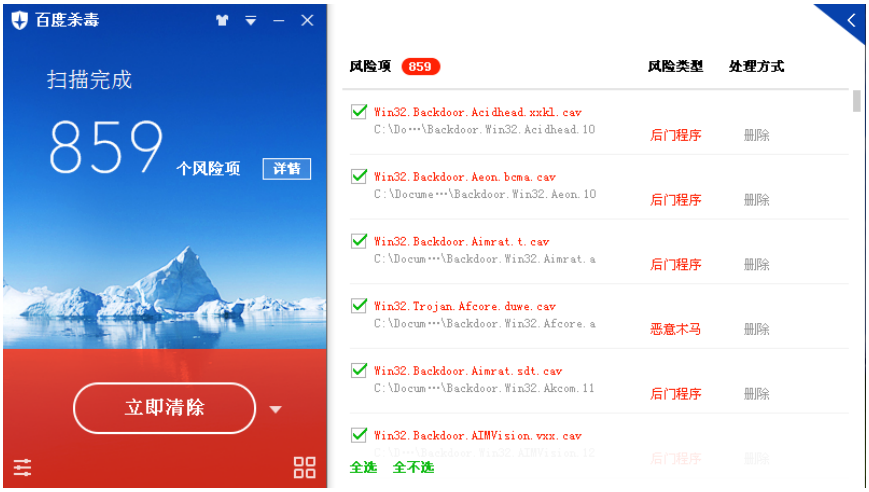



figure 18 the test result of Baidu

Baidu's product gets the rate as 85.9% in killing rate test.

But it get the first place in the killing time test as 70 seconds.

## Jinshan

We update the version of Jinshan to SP1.8.052500.

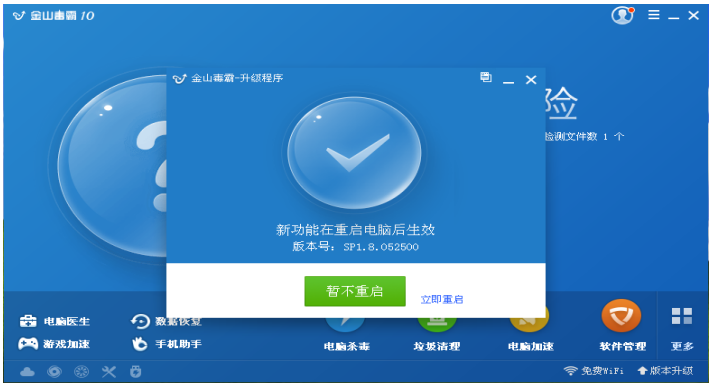



figure 19 version information of Jinshan

figure 20 the test result of Jinshan

Jinshan kills 80.1% samples in Benchmark while it costs 419 seconds to kill them.

## Tencent QQ PC Manager

Tencent QQ PC Manager is another popular antivirus softwares in China region.

figure 21 GUI of QQ PC Manager

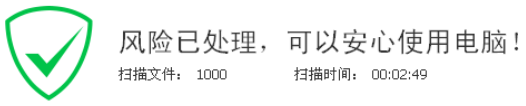



figure 22 the test result of QQ PC Manager

The killing rate of QQ PC Manager is 93.9%, while the killing time is 169 seconds.

## Kaspersky LAB

Kaspersky is a antivirus software from Russia. It works on statistic analysis.

figure 23 update Karspersky LAB antivirus software

figure 24 the test result of Karspersky

In the result Karspersky shows that there is 1396 files in Benchmark and its killing rate is 71.1%. When we sets Karspersky in default setting, it uses a long time as 992 seconds to kill all the malwares it finds.

## ESET

ESET smart security is installed and updated.

figure 25 update the ESET smart security

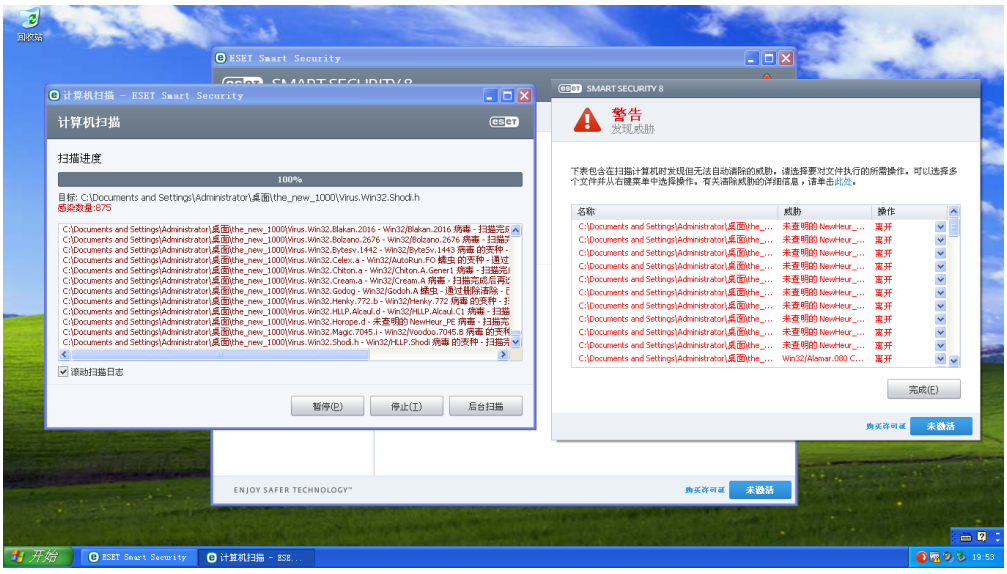



figure 26 the test result of ESET smart security

ESET killing rate is 87.7%.

ESET killing time is 151 seconds.

## AVAST!

We set AVAST! at the last place because we find something wrong and special of it.

First, install and update as usual.

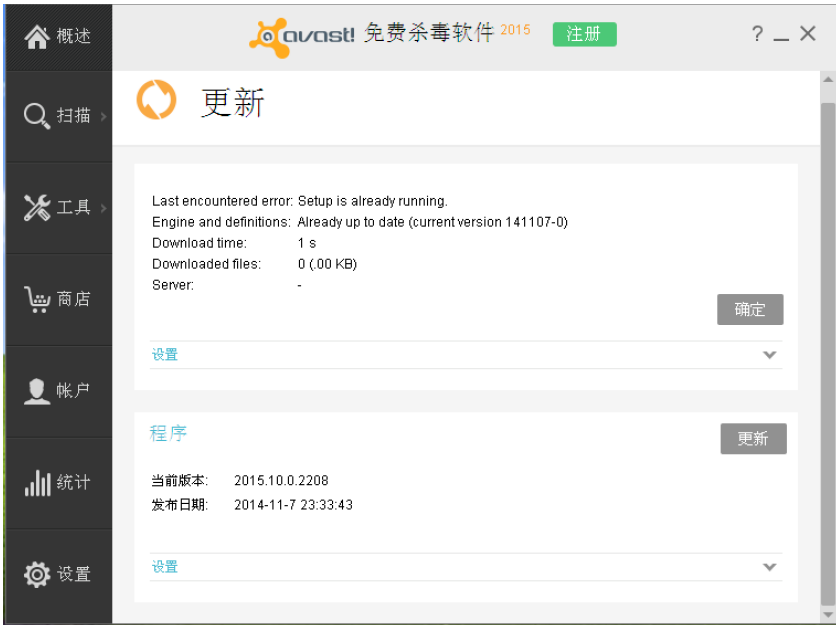



figure 27 the update GUI of AVAST!

We find that AVAST! can't update its malware database in time and some of its functions can't work on this platform.

AVAST! is a famous antivirus software form Czekh. But if you test in China region, the path to update is forbidden. So AVAST! do so bad in the test.

figure 28 the test result of AVAST!

AVAST!'s killing rate is 30.9% only.

Its killing time is 22 seconds.

## Conclusion

The killing rate table is following:

table 4 ranking of the killing rate

| Antivirus Software Name | Killing Rate |
|---|---|
| Qihoo 360 | 0.994 |
| Rising Internet Security | 0.987103175 |
| AVG | 0.960707269 |
| Tencent QQ PC Manager | 0.939 |
| Avira | 0.92233009 |
| ESET | 0.876753507 |
| Baidu | 0.859 |
| Jinshan | 0.801356589 |
| Karspersky LAB | 0.71058315 |
| AVAST! | 0.3099851 |

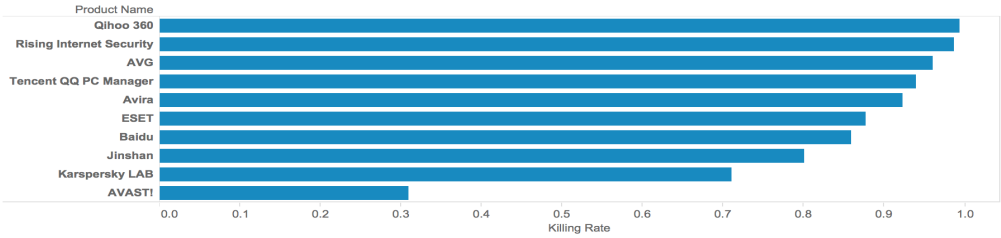



Figure 29 bar chart of the killing rate of antivirus softwares

We record how many seconds one antivirus software costs to kill the Benchmark test set. The time we record in the following table is the total of the detection time and the remove time.

table 5 ranking of the killing time

| Antivirus Software Name | Killing Time(s) |
|---|---|
| AVAST! | 22 |
| Baidu | 70 |
| AVG | 73 |
| Qihoo 360 | 151 |
| ESET | 151 |
| Tencent QQ PC Manager | 169 |
| Rising Internet Security | 240 |
| Avira | 322 |
| Jinshan | 419 |
| Karspersky LAB | 992 |

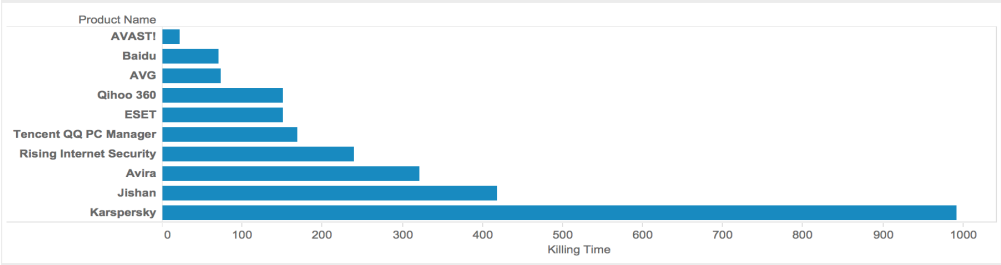



figure 30 bar chart of the killing time of antivirus softwares

Remarks:

In this test, Qihu360's product does the best in killing rate, which is 99.4%, while Rising's product get the second position and AVG is the third.

In killing time test, Baidu's product costs only 70 seconds to kill the malwares it has detected. AVG costs a little more and 360 is on the third position. (ps. Because there are something mistake with AVAST!, we don't take it in the rank)

In addition, we do this test in China so that some antivirus softwares is not as good as they are in the test of AV-C or AV-TEST because of limiting conditions. So we can find in this test that there are many unpredictable reasons that cause the antivirus softwares can't work as well as before. And the result from the traditional testing organization may mislead the users.

# Conclusions

The original intention of Benchmark security test set is to create a new and open produce method of security test set instead of traditional ways, whose validity is doubt today. The original security test system can't meet the needs of security software company and users any more. So we aim to make Benchmark as an open test set and the test can be repeated as well.

About the produce method, our work widely use the data mining technique, including LCS, SVD, and the graph-based clustering algorithm, etc. to analyze malwares' behaviors. High autonomy is another character of this work as well, with the full system of capture, analysis and clustering.

About the test result, both the clustering test and the killing rate and time test, are successful. It proves that this method is effective and practical. Benchmark can be as a security test faced to all users.

About the application prospect, Benchmark can be improved into a full security test set with multiple versions on different platforms and for different level of security. Except for these, Benchmark can also

be improved as an early warning system for malwares on the Internet.

# Reference


[1]D.Ye, M.Moffie, D.Kaeli. A Benchmark Suite for Behavior-Based Security Mechanisms[C]. Maryland: In Proceedings of the Workshop on Software Security Assurance Tools, 2005.

[2]James, Poe, Tao, Li. BASS: a benchmark suite for evaluating architectural security systems[J]. Computer Architecture News, 2006, 34(4): 26-33

[3]Georg Wicherski. pehash: A novel approach to fast malware clustering. In 2nd Usenix Workshop on Large-Scale Exploits and Emergent Threats (LEET'09), 2009.

[4]Md Enamul Karim, Andrew Walenstein, Arun Lakhotia, and Laxmi Parida. Malware phylogeny generation using permutations of code. JOURNAL IN COMPUTER VIROLOGY, 1:13–23, 2005

[5]BAYER U, COMPARETTI P M, HLAUSCHEK C, et al. Scalable,behavior-based malware clustering[A]. Network and Distributed System Security Symposium (NDSS)[C]. 2009.8-11.

[6]Konrad Rieck, Thorsten Holz, Carsten Willems, Patrick D¨ussel, and Pavel Laskov. Learning and classification of malware behavior. In Proceedings of the 5th international conference on Detection of Intrusions and Malware, and Vulnerability Assessment, DIMVA '08, pages 108–125, Berlin, Heidelberg, 2008. Springer-Verlag.

[7]Michael Bailey, Jon Andersen, Z. Morley mao, and Farnam Jahanian. Automated classification and analysis of internet malware. Technical report, In Proceedings of Recent Advances in Intrusion Detection (RAID07, 2007)

[8]Abraham Silberschatz, Peter Baer Galvin Greg Gagne. OPERATING SYSTEM CONCEPTS(Seventh Edition), Published by Higher Education Press. 2012-3.

[9]Thomas H.Cormen, Charles E. Leiserson, Ronald L.Rivest, Clifford Stein. Introduction to Algorithms(Third Edition), Published by China Machine Press. 2013-8.

[10]Michael Sikorski, Andrew Honig, Practical Malware Analysis. 2012-8.