Internal distribution code:

(A) [-] Publication in OJ
(B) [-] To Chairmen and Members
(C) [-] To Chairmen
(D) [X] No distribution

Datasheet for the decision of 20 October 2016

Case Number: T 1575/11 - 3.5.06

Application Number: 04014752.2

Publication Number: 1505499

IPC: G06F9/445

Language of the proceedings: EN

Title of invention:
Automatic detection and patching of vulnerable files

Applicant:
Microsoft Technology Licensing, LLC

Headword:
Vulnerability detection/MICROSOFT

Relevant legal provisions:
EPC 1973 Art. 56

Keyword:
Inventive step - (no)

Decisions cited:

Catchword:

Beschwerdekammern
Boards of Appeal
Chambres de recours

European Patent Office
D-80298 MUNICH
GERMANY
Tel. +49 (0) 89 2399-0
Fax +49 (0) 89 2399-4465

Case Number: T 1575/11 - 3.5.06

D E C I S I O N
of Technical Board of Appeal 3.5.06
of 20 October 2016

Appellant: (Applicant)	Microsoft Technology Licensing, LLC One Microsoft Way Redmond, WA 98052 (US)
Representative:	Grünecker Patent- und Rechtsanwälte PartG mbB Leopoldstraße 4 80802 München (DE)
Decision under appeal:	**Decision of the Examining Division of the European Patent Office posted on 18 March 2011 refusing European patent application No. 04014752.2 pursuant to Article 97(2) EPC.**

Composition of the Board:

Chairman	W. Sekretaruk
Members:	S. Krischer
	M. Müller

Summary of Facts and Submissions

I. The appeal is directed against the decision of the examining division, dated 18 March 2011, to refuse the application 04014752 for lack of inventive step (main request) and lack of clarity (auxiliary request).

The following documents have been cited (in the order of relevance):

D2 US 2003/126472 A1.

D4 S. Kumar et al.: "A Generic Virus Scanner for C++", Computer Security Applications Conference, 30 November 1992, pages 210-219, XP10030999, San Antonio, Texas/USA, ISBN: 978-0-8186-3115-3.

D1 C. Liu et al.: "Automated security checking and patching using TestTalk", Automated Software Engineering Conference, 11-15 September 2000 , IEEE, pages 261-264, XP10513979, ISBN: 0-7695-0710-7.

II. A notice of appeal was received on 12 May 2011. The appeal fee was received the same day. A statement of grounds of appeal was received on 27 June 2011. Oral proceedings were requested.

III. In its summons to oral proceedings, the board gave reasons for its preliminary opinion that claim 1 of the two requests lacked an inventive step over D2 and that claim 1 of the auxiliary request was clear.

IV. In a letter dated 1 June 2016, the appellant submitted arguments as to why the claims were inventive.

V. Oral proceedings were held on 20 October 2016. At their end, the chairman announced the board's decision.

VI. The appellant requests that the decision be set aside and a patent be granted based on a main request, filed with the notice of appeal (identical to refused main request), or an auxiliary request, filed with the grounds of appeal (identical to the refused auxiliary request), and the description pages and drawings on file, i.e.: pages 1-3, 5-22 as originally filed, page 4 as filed on 10 July 2006; drawing sheets 1-5 as originally filed.

VII. Claim 1 of the main request reads as follows:

"1. A processor-readable medium comprising processor-executable instructions configured for:

receiving a binary signature of a vulnerable section in a binary file at a server, the binary signature being a bit pattern that is associated with a security vulnerability in a binary file on a client computer;

receiving a security patch at the server;

identifying, by the server, a vulnerable binary file on the client computer based on the binary signature, wherein the identifying a vulnerable binary file includes scanning, by the server, binary files located on the client computer and comparing, by the server, the bit pattern of the binary signature against the binary files located on the client computer; and

updating, by the server, the vulnerable binary file on the client computer with the security patch."

VIII. Claim 1 of the first auxiliary request reads as follows:

"1. A processor-readable medium comprising processor-executable instructions configured for:

receiving a binary signature at a server, the binary signature being an exact bit pattern of a vulnerable function within a binary file, the bit pattern being associated with a security vulnerability in the binary file on a client computer;

receiving a security patch at the server;

identifying, by the server, a vulnerable binary file on the client computer based on the binary signature, wherein the identifying a vulnerable binary file includes scanning, by the server, binary files located on the client computer and comparing, by the server, the bit pattern of the binary signature against the binary files located on the client computer; and

updating, by the server, the vulnerable binary file on the client computer with the security patch."

Reasons for the Decision

1. *Overview of the invention*

The application relates to scanning a client computer for so-called vulnerable files and fixing them with security patches. The files can be of any type (e.g. executable files or "other data useful for client computer 108", see A1 publication, paragraph [39], column 11, lines 6-11). The scanning is either performed by a server computer ([35], column 9, line 56 to column 10, line 1; [15], fifth sentence; [26]; claim 1) or by the client computer ([35], column 10, lines 2-5; [39], column 11, lines 2-6; [15], lines 41-45; [24]; [30]; claim 5). For doing so, binary signatures identifying vulnerable files ([21], lines 17-23) are searched in the files of the client computer (by the server: [26] and figure 2; by the client: [30] and figure 3). If a vulnerable file has been found, it is updated with a security patch from the server.

2. *Inventiveness*

2.1 In the following, the board only discusses in detail the auxiliary request. Since claim 1 of the auxiliary request is more specific than claim 1 of the main request, the arguments relating to the auxiliary request apply a fortiori to the main request.

2.2 The board considers D2 to be a suitable starting point for the assessment of inventive step, as did the examining division (see the decision, section 3.1).

2.3 According to the decision (reasons, section 3.2), claim 1 differs from D2 in that it defines the binary signature as being a bit pattern that is associated with a vulnerability in a file. The step of receiving a binary signature of a vulnerable section is considered to be disclosed in D2, paragraph [31] which relates to the download of remediation signatures and vulnerability information as depicted in box 82 of figure 5A.

2.4 The appellant contests in the grounds of appeal (page 3, paragraph 4) that the step of receiving a binary signature being a bit pattern associated with a vulnerability is disclosed in D2.

2.5 The board agrees. However, the decision did not state that bit patterns associated with vulnerabilities were disclosed in D2. Furthermore, it is clear that D2 cannot disclose the receiving of a binary signature being a bit pattern, since bit patterns are not disclosed in D2 (as acknowledged by the decision). In the decision, a binary signature seems to be understood as comprising any information which enables the server to detect vulnerabilities in the files of the client computer.

2.6 The board disagrees with the decision that receiving such a (generalised) binary signature *at the server* is disclosed in D2, since in D2 *not the server* (identified by the decision with the client server 22 in figure 1 of D2) performs the scanning for vulnerabilities, *but (security) intelligence agents* (14) coupled via so-called remediation and flash servers to the client server (see [19], sentences 1-5; claims 20 and 28; and figure 1). Therefore, also the step of "identifying, by the server, a vulnerable binary file ... wherein the identifying ... includes *scanning, by the server* ..." (emphasis added) is not disclosed in D2.

2.7 In summary, the board finds that the claim differs from D2 in that the claimed server performs both the functions of identifying and updating vulnerable files on the client computer, whereas in D2 two separate, but network-connected entities perform them (the security intelligence agent performs the identifying and the client server performs the updating). Furthermore, D2 does not disclose how the search for vulnerabilities in the files is done, whereas the claim specifies that the server compares bit patterns of known vulnerabilities against the files of the client computer.

2.8 The board is of the opinion that the two differences do not interact with each other. Therefore, the invention can be considered to solve two partial objective technical problems in comparison with D2, namely 1) how to distribute the functions on the available computers, and 2) how to search for vulnerabilities in executable program files.

2.9 As to problem 1), the board regards it as obvious to a programmer to let one server perform the functions which were apparently done by two servers in D2 if he only has one server at his disposal, or if he considers it for one reason or another to be appropriate to choose only one server. The advantages and disadvantages of choosing one or two servers are immediately conceivable in advance.

2.10 As to problem 2), the binary signature is claimed as "an exact bit pattern of a vulnerable function within a binary file" and as "associated with a security vulnerability in the binary file on a client computer". According to the only disclosure in the description concerning the nature of these bit patterns ([21], third sentence; see also claim 1 of the auxiliary request), it is an "exact bit pattern of the vulnerable function within the [software] product" and the "binary signature of the vulnerable section in the binary file, which is a component of a software product" ([21], fourth sentence). The board understands this to define the bit pattern as the object code of a vulnerable function (e.g. obtained from a procedure or function of a high-level programming language). This means that the object code of a vulnerable function is searched for in the files of a client computer. This is an obvious solution of problem 2) of how to search for vulnerabilities in executable program files.

2.11 In addition, the board also agrees with the decision (section 3.5) that a programmer concerned with solving the above problem 2) would consider how programs are searched in other fields of software security. He then would think of virus scanners (e.g. D4) which are searching for object code of the programs to be found (i.e. virus programs). It would be obvious to do the same for vulnerabilities in executable program files.

2.12 In the grounds of appeal (page 5, fifth paragraph) it is further argued that the skilled person would not take into account the teaching of D4, since D4 was "concerned with the case when the virus is already present on the client computer".

2.13 The board understands the appellant's argument to be that virus detection is searching for the traces of an attack that already happened rather than for a code segment that is prone to get attacked.

2.14 This does not convince the board, since for the purpose of searching it is immaterial "what" is being searched, especially given that the "vulnerabilities" are not defined in the claim and code is not technically characterized by containing a "vulnerability". Searching whether the code of a virus is present in an executable program file is *technically the same* as searching whether the code of a vulnerable function is present.

2.15 The appellant (grounds of appeal, paragraph bridging pages 5 and 6) further argues that a combination of D2 with D4 would yield a vulnerability detection program with an additional virus scanner.

2.16 However, such a straightforward combination of D2 and D4 is not the way the board argues above about how to solve the objective technical problem 2). It would furthermore not solve that problem.

2.17 In its letter of 1 June 2016, the appellant argued on page 3, second paragraph, that paragraph [24] of D2 disclosed how to search for vulnerable files, namely (see page 2, third paragraph of the letter) in that the client server 22 keeps a profile of the client computer, containing the software applications and versions running on the client computer. Then this profile is compared with the vulnerability information.

2.18 The board notes that paragraph [24] describes a different embodiment than the one in paragraph [31], used by the board (and the examining division). The board agrees that paragraph [24] could give the skilled person a hint how to solve problem 2) (i.e. how to search for vulnerabilities in executable program files).

2.19 The fact, however, that D1 discloses or suggests one way of searching vulnerabilities does not make the claimed alternative less obvious. In fact, it would be simpler for the skilled person to directly search for the vulnerabilities in the executable program files.

2.20 Therefore, this argument does not convince the board.

2.21 Thus, claim 1 of the two requests is not inventive (Article 56 EPC 1973) over D2 in view of first principles on how searches are performed and, separately, in view of existing solutions as known from D4.

Order

For these reasons it is decided that:

The appeal is dismissed.

The Registrar: The Chairman:

B. Atienza Vivancos W. Sekretaruk

Decision electronically authenticated