NATIONAL SECURITY AGENCY
CENTRAL SECURITY SERVICE

DRAFT

(C//REL) Kaspersky User-Agent Strings

By

[redacted]

S3T/TECH/xxx/2008

September 2008

S-xxx,xxx

Derived From: NSA/CCSM 1-52
Dated: 8 January 2007
Declassify On: 20320108

(C//REL) Kaspersky User-Agent Strings

S3T/TECH/XX/2008

September 2008

S-xxx,xxx

WRITTEN BY:

____

[redacted]. IDA/CCR-P

REVIEWED BY:

[redacted], IDA/CCS

RELEASED BY:

____

[redacted]. Chief, S3T1

REPORT DOCUMENTATION PAGE	*Form approved* *OMB No. 0704-0188*

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)	**2. REPORT DATE** **September 2008**	**3. REPORT TYPE AND DATES COVERED** Technical SIGINT Report

4. TITLE AND SUBTITLE

(C//REL) Kaspersky User-Agent Strings

5. FUNDING NUMBERS

6. AUTHOR(S)

[redacted]

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

National Security Agency
Ft. George G. Meade, MD 20755-6400

8. PERFORMING ORGANIZATION REPORT NUMBER

S3T/TECH/–/2008

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

S-

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
THIS DOCUMENT MAY NOT BE RELEASED OR REPRODUCED IN WHOLE OR IN PART WITHOUT PRIOR APPROVAL OF THE ISSUING OFFICE.

12b. DISTRIBUTION CODE

13. ABSTRACT

(S//SI//REL) We discovered that Kaspersky User-Agent strings contain encoded versions of the Kaspersky serial numbers and that part of the User-Agent string can be used as a machine identifier.

14. SUBJECT TERM
Kaspersky, User-Agent, machine identifier

15. NUMBER OF PAGES
8

16. PRICE CODE
N / A

17. SECURITY CLASSIFCATION OF REPORT TOP SECRET//COMINT//REL USA, FVEY	**18. SECURITY CLASSIFICATION OF THIS PAGE** SECRET//COMINT//REL USA, FVEY	**19. SECURITY CLASSIFICATION OF ABSTRACT** SECRET//COMINT//REL USA, FVEY	20. LIMITATION OF ABSTRACT

S3T/TECH/XX/2008

Table of Contents

(C//REL) Kaspersky User-Agent Strings

(U) Introduction

(U//FOUO) Kaspersky Lab is a privately held company with headquarters in Moscow, with regional offices elsewhere. Kaspersky has (at least) three products Kaspersky Internet Security (KIS), Kaspersky Anti-Virus (KAV), and Kaspersky Mobile Security (KMS). The Anti-Virus engine is used by other security vendors. Kaspersky products are quite popular in some parts of the world.

(U//FOUO) This work was begun with [redacted] at SCAMP 2008 at IDA/CCR-Princeton..

(U) Data

(TS//SI//REL) We used YACHTSHOP metadata records for our study of Kaspersky User-Agent strings, as well as some information discovered by using Google searches on the Internet.

(U) User-Agent Strings

(TS//SI//REL) The Kaspersky client sends its own User-Agent strings when requesting updates. Some examples are

```
GET /diffs/bases/ids/i386/idsbase.kdz.7f- HTTP/1.0
Host: dnl-r01.kaspersky-labs.com
User-Agent: wnB4BwnB_p_ujCg31C0BANOC4wLjAuMzU3
```

```
GET /index/u0607g.xml.klz HTTP/1.0
Host: dnl-us5.kaspersky-labs.com
User-Agent: wnBAAAs9sHIBANOC4wLjAuNDU0
```

The Kaspersky User-Agent strings are of three types

1. E@
2. pD-kBpDm5jf46pEQ@
3. umB09BumBgmgvrB-sXFANNy4wLjAuMTI1

The User-Agent strings use the characters [A-Za-z0-9-_], which is the same alphabet as is used in base64 encoding. Further, the last twelve characters of the third type are, in fact, base64 encoding of the version number. These version numbers range from 6.0.2.614 to 8.0.0.357 in our data.

(U) Updates

(S//SI//REL) The update requests we observed often occurred on a regular basis, often every 20, 40, 120 or 140 minutes when the machine is on-line. They began with a GET request for an index page, i.e. /index/u0607g.xml.dif and/or /index/rt60.xml.klz.

(S//SI//REL) This was immediately followed by a set of requests for update files: first, a set of files such as

```
/bases/blst/blst-0607g.xml.dif
/bases/ids/i386/ah-i386-0607g.xml.dif
/bases/av/avc/i386/av-i386-0607g.xml.dif
/bases/av/avc/i386/av-i386-0607g.xml.klz
/bases/pdm/pdm-0607g.xml.dif
/bases/info/info-0607g.xml.dif
/bases/rt/rt-0607g.xml.dif
/bases/upd/upd-0607g.xml.dif
```

(blst = black list, ids = intrusion detection system, av = antivirus, upd = update),

then, a set of files such as

```
/AutoPatches/kav6/kav6-0607g.xml.dif
```

and a set of files such as

```
/diffs/bases/blst/black.lst.nsr
/diffs/bases/av/avc/i386/ext001.avc.wvg
/diffs/AutoPatches/kav6/7.0.1.321/avpgui.ppl.ryh.
```

We did not see any use of query strings or cookies in the update requests.

(S//SI//REL) User-Agent Fields and Encoding

(S//SI//REL) Now we turn our attention to the User-Agent strings themselves. Let us take a typical example, as above:

umB09BumBgmgvrB-sXFANNy4wLjAuMTI1

(S//SI//REL) The last 12 characters are the base64 encoded string Ny4wLjAuMTI1, which, in this case, decodes to 7.0.0.125 (the version number) and leaves us with

umB09BumBgmgvrB-sXFAN

(S//SI//REL) At first, it appears that there are fields separated by "B" characters, but upon looking more closely, it can be seen that there are two sets of characters, [A-Za-f] and [g-z0-9-_], i.e. two sets of 32 characters each. It appears that each field is a set of characters from the second set followed by a character from the first set. We believe that the characters are composed of a leading flag bit followed by five intelligence bits, where the flag bit indicates the end of a field. Thus we parse the above string into fields:

umB 09B umB gmgvbB -sX F A N

(S//SI//REL) Field one and field three are normally the same and the first three fields are usually two or three long. Field six is always one or two long, about half of the time two long. If it is two long, the second character is a "B". Since "B" is the second character of the first alphabet set, its natural value is 1. That is, if we reverse the order of the characters in this field, the entire field takes on values 0-63. Taking this as our cue, we concluded that the order of the characters in each field should be reversed, and each field represents a number encoded base32, with end-of-field flags.

(S//SI//REL) With this interpretation, field five appears to be flat over the range 0-2^{18}, the largest value seen being H_g7 = 261147, while 2^{18}=262144.

(S//SI/REL) The first five fields appear to match with specific clients. The main exception is Dp Bk- Dp fj5m Ko (another parsed Kaspersky User-Agent) which is seen with a large number of clients. As we shall see, fields two, three, and four are the serial number, and this particular serial number is one of those being passed around on the Internet.

(S//SI//REL) Studying the flow of GET requests, we observe that in many cases, there is an update request at regular intervals. Probably such a request is made in all cases in which the machine is on-line. These requests begin with a request for one or two index pages, followed by further requests for update files, all with the same User-Agent. Later, on the regular beat, the next request will have the same User-Agent string, except that field six will have changed:

```
Thu Jul  3 21:13:56 2008 Dp Esi Il Qjl0m Eq-0 M  A N  6.0.2.618
Thu Jul  3 23:33:56 2008 Dp Esi Il Qjl0m Eq-0 Bz A N  6.0.2.618
```

(S//SI//REL) The beat at which the requests are made appears to be correlated with the type and version number. The type 2 requests often come every 20 minutes, and about 20% of the time ticks up with an increment alternating between 30 and 34, mod 63.

(S//SI//REL) The type 3 requests often come at beats of 120 or 140 minutes, ticking up by 24 in the 120 case, and 39 in the 140 case, both mod 64.

(S//SI//REL) Field seven, if present in type 2 strings, is an "E", or in type 3 strings there is a field seven, "A", and possibly a field eight, "N". In our data, in the type 2 strings, unless there is a field seven with an "E", the requests only ask for klz files, such as /index/6/a0607g.xml.klz. Also in our data, the type 3 strings with no eighth field, i.e. no "N" value, were all from version 6.0.2.614, while none of the version 6.0.2.614 strings had a seventh field. We believe these indicate services and/configurations.

(U) Types of User-Agent Strings

(S//SI//REL) There isn't much to say about the first type of User-Agent string, "E@". It presumably represents some limited capability trial version.

(S//SI//REL) The second type is more interesting as it parses as described above. The parsed version usually begins Dp Bk- Dp fj5m, followed by a field five mostly of length three or four, a sixth field which ticks up as discussed before, and possibly a seventh field consisting of an "E@".

(S//SI//REL) The one exception is the Dp Bk- Dp fj5m Ko mentioned above, in which field five is two long. Further, this one does not tick up, but always appears the same.

(S//SI//REL) We have more information about the third type, in which the last 12 characters are the encoded version numbers. We observed:

Version	First Field
6.0.2.614	Dp, Bkt, or Bkx
6.0.2.618	Dp, or Dr
6.0.2.621	Dp, Bkx, or Bm7
6.0.3.837	Dt, or Dz
7.0.0.119	Bmt
7.0.0.124	Bmt
7.0.0.125	Bkt, or Bmt
7.0.1.321	Bmt, or Bmu
7.0.1.323	Bmu
7.0.1.325	Bmt, or Bmu, or Bnr
8.0.0.357	Bnw, or Bnx

(S//SI//REL) So it seems that the first field is tracking along with the version numbers, so it could relate to the date at which the product is activated, but is not directly equivalent to the version number.

(S//SI//REL) Type 3 strings are all seven or eight fields long.

(S//SI//REL) Kaspersky Serial Numbers

(S//SI//REL) Kaspersky products come with key files with names such as KI-KSS/00102CCB.key which contain a serial number and a license number, in this case the keys with serial number 03c2-000486-00102cc. Here are a set of such pairs (or one-half of such pairs) of serial numbers and license numbers, which we found posted on the Internet.

Serial number	License number	Product
0038-000069-0007ec85		
02d6-00006b-0009e727		
049e-000069-000f8f26	049E-060310-101621	
04f2-000069-000c1d94		
02d6-000486-000add31		
03c2-000486-00102ccb	03C2-060410-103939	Kaspersky Security Suite Personal International Edition. 1-Desktop 3 months NFR Licence Pack
03c2-000069-00102cc8	03C2-060410-103415	103415 Kaspersky Anti-Virus Personal International Edition. 1-Desktop 3 months NFR Licence Pack
03D4-000491-0099B28A		
07CA-000491-00DFA4CF		
	02F1-080128-120816	
	05FE-070221-143729	
0007-0003F5-036E91E6	0007-070493-133518	
092C-0004CE-03BC70BE	092C-080530-134222	
02C0-00045F-03ADBF24	02C0-080519-091046	
07B4-0004CE-02B780BA		
070A-0004CE-01E223F3	070A-070807-132355	
0494-0004CD-02439E4C	0494-071015-170322	
0007-000491-04432441	0007-060821-110449	Kaspersky Anti-Virus 6.0 International Edition. 1-Desktop 1 month Trial Licence Pack

(S//SI//REL) Notice that the first segment of the serial number and the license number is the same. Also it appears that the serial number is entirely in hex, while only the first segment of the license number appears to be in hex. With one exception, the second segment of the license number appears to be related to the version number; all the cases we observed in our data begin with a 6, 7, or 8.

(S//SI//REL) One of the frequently occurring User-Agent strings, when parsed, becomes (Dp Bk- Dp fj5m H47w). By converting these strings into hexadecimal numbers, this becomes

00000069 0000049e 00000069 000f8f26 0003e370.

(S//SI//REL) Now examine the lines above. We find serial number 049e-000069-000f8f26.

(S//SI//REL) We also have (Bnx Cps Bmu B7481- Eh1v), which equates to 000004f1 0000092c 000004ce 03bc70be 000206af 00000032, which matches 092C-0004CE-03BC70BE, above.

(S//SI//REL) There are also close matches:

000004ce 000007b4 000004ce 02b780c0 00005d9e

with

07B4-0004CE-02B780BA,

and

000004cd 00000494 000004cd 02439bf2 0001362b

with

0494-0004CD-02439E4C.

(U) Key Files

(S//SI//REL) We located three key files and examined them. The first four bytes of the key files contain the signature KLsw. After an initial header, the key files can be parsed into records with an algorithm something like type-length-value. In general, the fields of the records are as follows (in hex):

Position	Content
1	01
2	00
3	00
4	09
5	01, 03, or 05
6	Specific kind of information in the value field
7	00

8	00 or 08
9	Format (01= blank, 09=4-byte word, 0b=8-byte word, 28=string beginning with 2-byte little endian length, 49=16 byte word,
10+	Value

(S//SI//REL) For example:

01 00 00 09 05 01 00 01 28 16 00 4b 61 73 70 65 72 73 6b 79 20 6c 61 62 20 6b 65 79 20 66 69 6c 65

(S//SI//REL) This is a record in which the data is in string format of length 0x0016 =22, with the string being "Kaspersky lab key file". Byte 5 was also 01 in the (two) other key files we looked at.

(S//SI//REL) The version number was given in the record in which byte 5 = 0x05. The GUID is given in the record in which byte 5 = 0x17. The customer name and license number is given in the records in which byte 5 =0x08. However, other lines with byte 5 = 0x08 contain an email address. The serial number is given in a header, which doesn't play by any of these rules.

(S//SI//REL) The key files contain the user's name, location, ISP, a GUID, and what appears to be a base64 encoded key string. It might be possible to get more information out of these files with sufficient examples.

(S//SI//REL) We discovered what appear to be some activation keys being passed around on the Internet, either because these keys represent the free trial keys or because the software has been hacked:

HTWUA-543AX-SZ9DB-YQZSQ
T1JVS-NNMBD-K1QTN-SUBP8
T6B6K-8YK22-VBQH7-ZUZJG
M38RS-DZJS7-X4ZT7-UACTR
2P67K-E9TG7-EF7QQ-YM2XZ

(U) Update Files

(S//SI//REL) Update requests retrieve various types of files which are identified by short printable strings at the beginning of the file. In a small survey, we observed APDB, AVZ, DIFF, DIFT, EK.8, KDC1, KL, KLBL, KLD2, KLZF, PK, SFDB, PK, SFDB, and SQZE.

(S//SI//REL) Among other things, these files contain base64 encoded strings, such as

B+ANCjs6MTA0NmRONW5CandscXVWb1MwOERWL09yaE15dlN4ampKSERsV1k
rdnJ5OUNPVk15R1lEWHYxTy8yMHdvOFlIV1lLUm82b3VrSWthcWJlbzFwaGdjbnZ
Pbi

which decodes to:

<cr><lf>;:1046dN5nBjwlquVoS08DV/OrhMyvSxjjJHDlWY+vry9COVMyGYDXv1O/2
0wo8YHWYKRo6oukIkaqbeo1phgcnvOn

and/or

0XLSznpdI71fB300e7Uwj1FQiJdREynaxjSBwY/592TW8JvlVEx1VX8CgX

which does not decode to anything immediately interesting

(U) Conclusion

(S//SI//REL) It appears that the Kaspersky User-Agent string used to request updates is in general unique to the client, carries a coding for the serial number, and can be used for machine identification. We also believe the User-Agent string carries information about services contracted for or configurations. Study of a few matched versions numbers, User-Agent strings, serial numbers, license numbers, activation keys, and key files should settle this.

DISTRIBUTION

Hardcopy
DC324

Softcopy
C - [redacted]
I73 - [redacted]
I7 - Technical Library - Ms. [redacted]
R1 - [redacted], [redacted]
R1 Library – [redacted] – distribute to IDA/CCR-P, IDA/CCR-LJ, IDA/CCS
R2 - [redacted]
R21 - [redacted], [redacted]
R22 - [redacted]
R23 - [redacted]
SSG1 - [redacted]
SSG111 - [redacted]
SSG2 – [redacted]
S3T- [redacted]
S3T1 - [redacted], [redacted]
S3T Classified Technical Library, Ms. [redacted]
S31 - [redacted]
S311 -- [redacted]
S3111 - [redacted]
S31114 - [redacted]
S3112 - [redacted]
S3114 - [redacted]
S3117 - [redacted]
S312 - [redacted]
S3121 - [redacted]
S31212 - [redacted], [redacted]
S3122 - [redacted]
S3124 - [redacted]
S313 - [redacted]
S314 - [redacted]
S32 – [redacted]
SUSLO-2 - [redacted]
CSE (H4) – [redacted]
CSE (HIO) - [redacted]
DSD - [redacted], [redacted]
GCHQ (OPC-ALTO) – Technical Library – [redacted]
GCHQ (B13B) - [redacted]
GCSB – [redacted], [redacted], [redacted]