

SPOOKS

COLLABORATORS

	<i>TITLE :</i> SPOOKS		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		February 23, 2023	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	SPOOKS	1
1.1	1
1.2	mr216	1
1.3	xvs	3
1.4	4ef9	3
1.5	bastard	4
1.6	bobek	7

Chapter 1

SPOOKS

1.1

Spooky news: 29.5.2001

New xvs.library.
Spooky news: 05.6.2001

212 bytes linkvirus
The best of previous spooky news :-)

4ef9 trojans

BASTARD analyze

BOBEK analyze

1.2 mr216

The virus was/is available as source code,
and isn't on the spread (I hope so).
This was written by known persons:
MadRoger (NeuroticDeath)
Pandamen (Bastard, Bobek).
File removals will be added to VirusExecutor as soon as possible.
File detection and removal will be as easy as it was with Bobek.

Entry.....: 212-bytes
Alias(es).....: -
Virus Strain.....: MadRogerShort
Virus detected when.: 6.2001
 where.: Internet
Classification.....: System/Linkvirus, memory-resident, not reset-resident

Length of Virus.....: 1. Length on storage medium: 212 Bytes
2. Length in RAM: 0 Bytes
(uses system stack to hide it's code)

----- Preconditions -----

Operating System(s)..: AMIGA-DOS Version/Release..: V36+
Computer model(s)...: all models/processors (MC68000-MC68060)

----- Attributes -----

Easy Identification..: none

Type of infection...: Self-identification method in files:
- first byte of first code hunk is \$61.B
Self-identification method in memory:
- checks for "do".W at sysStackLower offset 0
System infection:
- infects the following function:
 Dos Write()
Infection preconditions:
- Hunk Code is found
- File is not infected already
- file is smaller than \$7c0*4

Infection Trigger...: Copying executable files

Storage media affected:
 all dos devices (including RAM:)

Interrupts hooked...: None

Damage.....: Permanent damage:
 - none
 Transient damage:
 - generating of bad files is possible

Damage Trigger.....: Permanent damage:
 - none
 Transient damage:
 - too simply infect code

Particularities.....: Smallest linkvirus for Amiga!
 This is much optimized MadRogerShort
 which was the smallest one until now.

Similarities.....: Code is equal to MadRogerShort.
 First long of first codehunk is replaced with
 jump to virus code.

Stealth.....: -

Armouring.....: -

Comments.....: The main goal of this virus is it's size.
There are some 'bugs' that may cause making
wrong files (lack of clever test routines).
The virus wasn't tested with bigger caches!

----- Agents -----

Countermeasures.....: -
above Standard means.....: -

----- Acknowledgement -----

Location.....: Pawlowice, Poland 6.2001
Classification by...: Zbigniew Trzcionkowski
Documentation by...: Zbigniew Trzcionkowski
Date.....: 6.2001
Information Source..: Analyze of virus and source code
Copyright.....: This documentation is public domain

===== End of 212 bytes virus =====

1.3 xvs

Jan Erik Olausen got xvs.library and as soon as he know
how it does work he will continue it!

1.4 4ef9

This text says some words about trojans recently made by UAE lamer
who thinks that having 4ef9 linker makes him hacker.
Such lame trojans have been seen about five years ago,
but now additional code instead of manipulating BBS:user.data
tries to insult innocent via internet.
This is the only difference during the years, this means the lamers
still on the same level.

BlazeWCP.lha 32862 bytes the file 'BlazeWCP' had been
linked with shitty e-mail sender,
and the file version got faked
said to be: v1.8

FBlit.lha 142086 bytes the file 'FBlit' had been linked
with shitty e-mail sender,
and the version got faked
said to be: v3.84

StackAttack.lha 69229 bytes the file 'StackAttack' had been linked
with shitty e-mail sender,
and the version got faked

said to be: v1.2b

Safe.lha 20737 bytes the file 'Safe' had been decoded,
version had been changed, then code was
linked with the same shitty e-mail sender,
and crunched until size reached 7000 bytes.
Lamer used quite old Safe. Maybe he was
thinking about attack from longer time.
This archive was replaced with correct Safe,
as soon as it was possible (Thanks Error!)
said to be: v14.10

Please check Your system with e.g. VirusZ and note that,
4eb9 or 4ef9 linker can't mean anything good to you.
Those linkers were used to make trainers or add crack intros,
but also (surprise!) were often used to make trojans.
Experience learns that linked code always does more harm than good.
If You were watching Your fresh stuff with good (even old)
AV software You would easily see such strange things.

The linked code is 940 bytes long and is EOR crypted.
Same code was used to make all four trojans.
Safe v15.2 is able to stop processes created by this added code,
so email won't be sent every 60 seconds.
Linked part tries to send some insulting text.
After decoding You can see inside such shit:

```
(...)  
0290 aage-partner.com  
02A0 >..DATA..From: M  
02B0 OS Rul3z y0u bit  
02C0 ch! <>..Subject:  
02D0 MorphOS - The R  
02E0 eal Slim OS4ady.  
02F0 ..Fuck U JERKIN  
(...)
```

The e-mail is sent to haage&partner and contains some insulting text.

Note to lamer:
What da fuck the 'memory leak is'?
I don't have UAE, so I am wondering about that...

Thanks to Jan for sending all the archives, and to Error for
keeping eyes opened.

1.5 bastard

```
Entry.....: BASTARD (temporary name)  
Alias(es).....: -  
Virus Strain.....: Motaba(?)  
Virus detected when.: 4.2001  
                  where.: internet
```

Classification.....: Linkvirus,memory-resident, not reset-resident
Length of Virus.....: 1. Length on storage medium: c.a.2100 Bytes
(uses polimorphic engine)
2. Length in RAM: 8192 Bytes

----- Preconditions -----

Operating System(s):: AMIGA-DOS Version/Release..: 2.04 and above (V37+)
Computer model(s)...: all models/processors (MC68000-MC68060)

----- Attributes -----

Easy Identification.: none

Type of infection...: Self-identification method in files:

- checks first byte of first codehunk for \$61
(part of jump to viruscode)

Self-identification method in memory:

- indirectly the virus is aware of itself:
 - * checks for \$-1 in tc_userdata
field of every process, this value
is stored by exec/TaskWait list scanner,
already checked processes are skipped
 - * the try to hack asl.library fails,
so memory is freed

System infection:

- tries to guess paths to runned programs
via pr_Homedir and task name.
This gives about 2-5 valid filepaths
(mainly in WBStartup) to infect.
- Tries to hack in memory code of AllocRequest
of asl.library with patch that tries to
hack VirusCheckerII process (gets
via seglist Open call of this killer and
patches it!). I don't know
which version(s) author of virus had tested.

Infection preconditions:

- File is between 2000 and 32000 bytes
- Hunk Code is found
- File is not infected already
- device is validated
- device contains free blocks

Infection Trigger...: 1. Accessing files via checking them with VirusCheckerII.
2. Direct infection of some runned programs
after run of an infected file.
Files containing a "l" or "L" or "-" or "v" or "V"
will be not infected.

Storage media affected:

all DOS-devices

Interrupts hooked...: None

Damage.....: Permanent damage:

- Crashes system.

Transient damage:

- none

Damage Trigger.....: Permanent damage:

- File ENV:mui/spirit.1.prefs exists

Transient damage:

- none

Particularities.....: Polimorphic decrypt routine.

The decryptor is 256 bytes long and before

it is always: `movem.l d0-a6,-(sp)`

This engine is (for me) a new one, but doesn't

contain enough stuff to prevent "checksum"

detecting of the infected files.

The truth is even better. We can decode virus

using the technic found inside it

(the crypter and decrypter are same!).

The polimorphic engine always contains

one loop, one `eor`, one `move.l 4.w,a6`,

two `lea.l rest` are random `moveq` and `shifts`

like `lsl.l #2,d4` etc.

The decrypt algo may vary

if in the decrypt loop appear random

instruction that changes cryptkey register,

I didn't get any crashing example.

The virus replaces first longword of the

first codehunk with `bsr.w` to virus code.

The original value is restored by

decrypted virus code. And the stack will

be manipulated to call the program first

and then call the main virus code.

Note that there is no detailed check for

this long, so every file without \$61

at the begin will be infected.

This means also that files with `reloc` instruction

in first long will cause guru after infection.

New ideas at all. The virus looks excellent

compared to Motaba-3 that is supposed to

be the base of this viral engine.

Direct hacking of things that are ram only

is problematic subject and there is incredibly

large amount of things that can be hacked

in future in the same way.

One of these bastards that

if run from an icon will not crash

with the wellknown GURU 87000004. That's because

of the executing of virus code AFTER program.

Similarities.....: Link-method is first hunk increasing.

The main code is comparable to motaba-3.
Length polymorph is same!
The change of lenght is depending on
'a' in filepath.
The path creator is idea comparable to
Antonio and PolishPower viruses.

Stealth.....: FindTask must be pointing to \$fxxxx or virus
will not try to hack VCII.
Open must be pointing to \$fxxxx or virus
will not perform any action.
Write must be pointing to \$fxxxx or virus
will not perform any action.
Lock must be pointing to \$fxxxx or virus
will not perform check for ENV:mui/spirit.1.prefs.
The virus doesn't patch ROM library vectors,
and the hackings of VC and asl.library are done
in quite tricky way.

Armouring.....: Polymorphic decryptor is used, length
of added code is changing in small range and
at the end of the virus is more or less garbage.
The virus contains some of the popular tricks
like bsr and then increasing sp to mix code
with data and some confusing/antidisassembling
instructions.

Comments.....: -

----- Agents -----

Countermeasures.....: -
above Standard means.....: -

----- Acknowledgement -----

Location.....: Pawlowice, Poland 4.2001
Classification by...: Zbigniew Trzcionkowski
Documentation by....: Zbigniew Trzcionkowski
Date.....: 4.2001
Information Source..: Virus disassembly and reverse engineering
Copyright.....: This documentation is public domain

===== End of BASTARD =====

1.6 bobek

Entry.....: Bobek!

Alias(es).....: -

Virus Strain.....: -

Virus detected when.: -

where.: internet

Classification.....: Linkvirus, memory-resident, not reset-resident

Length of Virus.....: 1. Length on storage medium: 460 Bytes

2. Length in RAM: 65535 Bytes

```
----- Preconditions -----
Operating System(s):: AMIGA-DOS Version/Release..: 2.04 and above (V37+)
Computer model(s)...: all models

----- Attributes -----
Easy Identification:: visible text '[BOBEK!]' in every infected file
Type of infection...: Self-identification method in files:

- compares length declared in hunkheader
  with the real length (this also
  avoids infection of some crunched files)

Self-identification method in memory:
- checks for $0 in libOpen address of exec.library.
  Exec base is available by second
  longword of memory, so this routine
  isn't used in normal system. This address is LOST
  forever (till reboot)!
  Note that this makes VT-Schutz not work.

System infection:
- infects ExNext function of dos.library
  The paths to infect are made with
  NameFromLock and filename in returned FileInfoBlock.
  This gives in some cases wrong paths,
  so some directories won't be touched by virus.
  The patch is done in way that prevents VirusZ
  from inviting user to VectorCheck.

Infection preconditions:

- File is between 1000 and 32000 bytes
- Hunk Code is found
- File is not infected already
- device is validated

Infection Trigger...: Scanning directories (filemanagers,
  filerequesters etc.).
  Not all files that could be infected will
  be infected at once.

Storage media affected:
  all DOS-devices

Interrupts hooked...: None

Damage.....: Permanent damage:
  - none
  Transient damage:
  - none

Damage Trigger.....: Permanent damage:
  - none
  Transient damage:
```

- none

Particularities.....: The virus code is highly optimized and looks like work of very experienced assembler programmer. The linker is made with one Open/Close, so there is no so much noise like with some other viruses. The code isn't crypted and the linked part has always static size 460 bytes. Memory allocation is done only once at start because of checking small range of filesizes. The infected file has always replaced first longword of first code hunk with BSR.W to init code which runs the program and then the virus itself. Due to above there is no problem of GURU after running from Workbench.

The programs with reloc entry at the replaced long may crash, but I couldn't find such example, because:
there is test for \$4E at the first LONG,
so files with such jumps will be untouched
(including the reloc ones: 4EF9 and 4EB9).

Similarities.....: Link-method is first hunk increasing. Used known code to check if the Open vector points to ROM.

Stealth.....: The cheating of VirusZ's VectorCheck is done with patchformat known from MCP and PatchControl.

The virus puts the new infected length to FIB. In the other words the ExNext always returns the real size of file.

The known from Motaba-3 trick is used to check if Open points to ROM. Due to that no virus action is visible in SnoopDos or DosTrace.

There are used two additional technics to decrease the noise while scanning directories:
- check if size is dividible by 4 (executables)
- infect try will be performed depending on state of bit 0 of \$dff007
IMHO it isn't good enough.

Armouring.....: Nothing special.

Comments.....: The virus contains VISIBLE string:

' [BOBEK!]'

NOTE!

1. There is no code to restore filedate after infection.
 2. Virus is not crypted.
-

Maybe someone forgot to release it few
years ago... :-)

The best way to switch the virus off in file
is to change visible string 'dos.library' few
bytes before [BOBEK!] to anything else.
Old Filemaster v2.2 is enough!
Xvs.library will be aware of that.

----- Acknowledgement -----

Location.....: Pawlowice, Poland 4.2001
Classification by...: Zbigniew Trzcionkowski
Documentation by....: Zbigniew Trzcionkowski
Date.....: 4.2001
Information Source..: Virus disassembly
Copyright.....: This documentation is public domain

===== End of [BOBEK!] =====
