Run-Time Detection of Self-Replication in Binary Malware

Alexander Volynkin
Victor Skormin
Douglas Summerville
James Moronski

Binghamton University

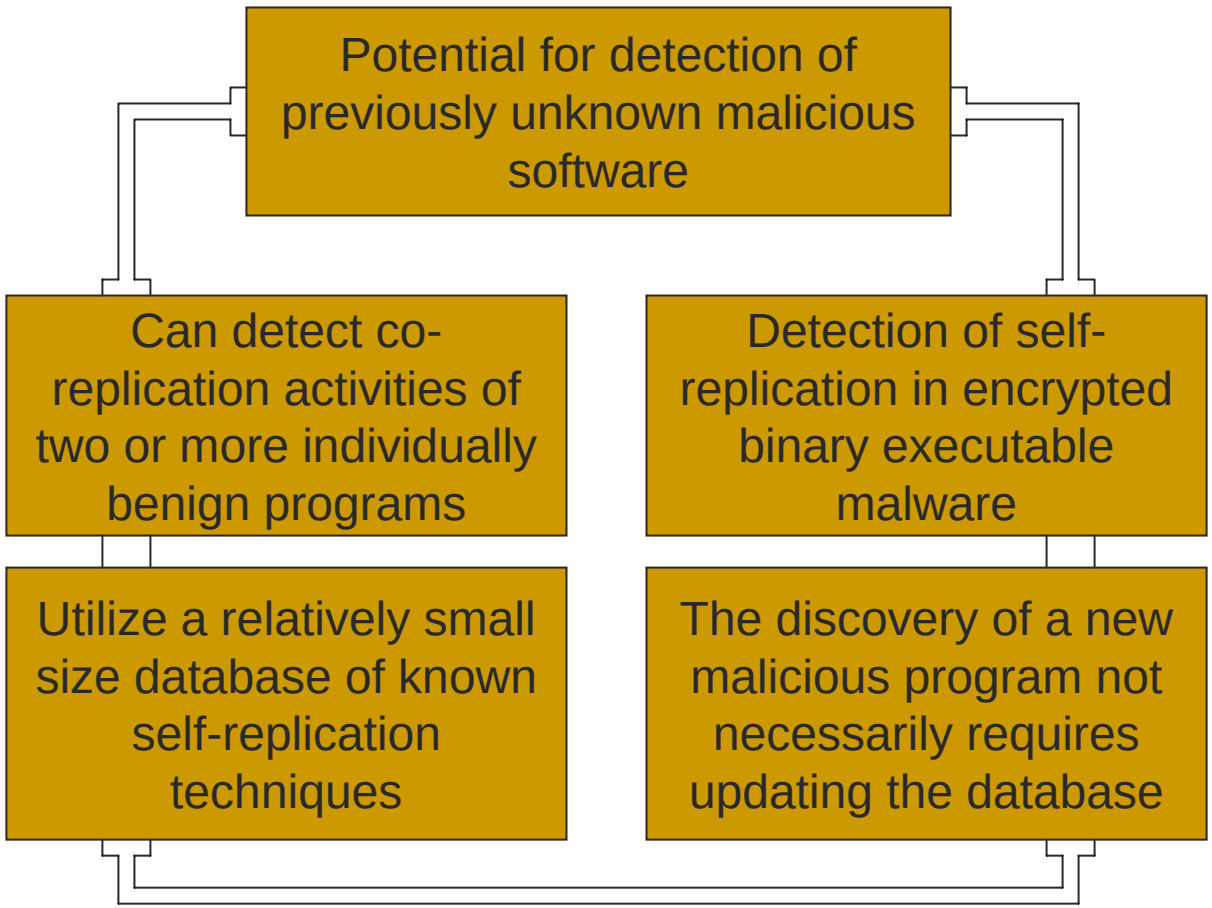
Gene of Self-Replication

Published previously

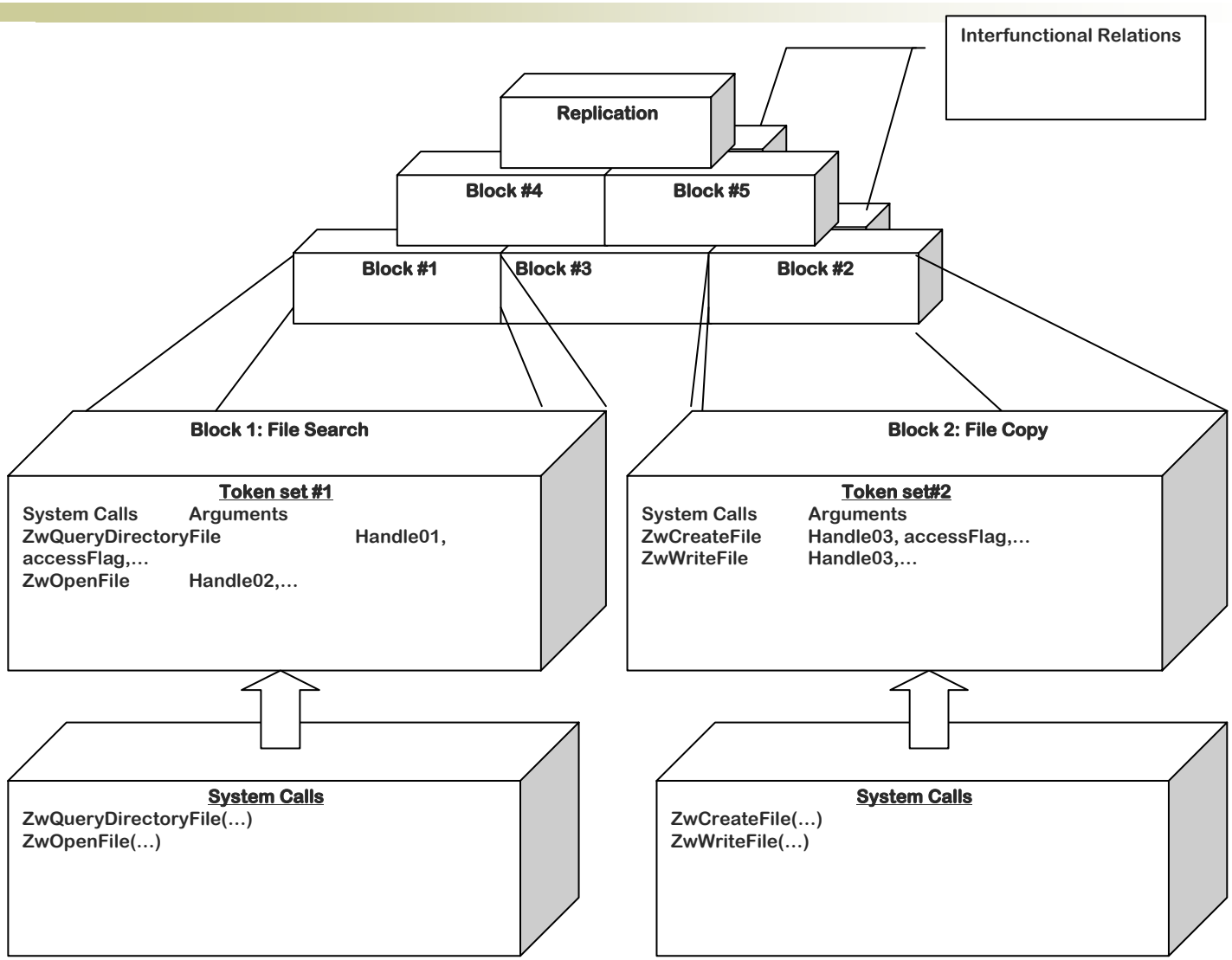
- Malware self-replicates to maximize the impact
- The number of practical techniques to implement self-replication is limited
- Developers of new viruses are destined to rely on a number of existing replication techniques
- Legitimate software seldom self-replicates

Gene of Self-Replication

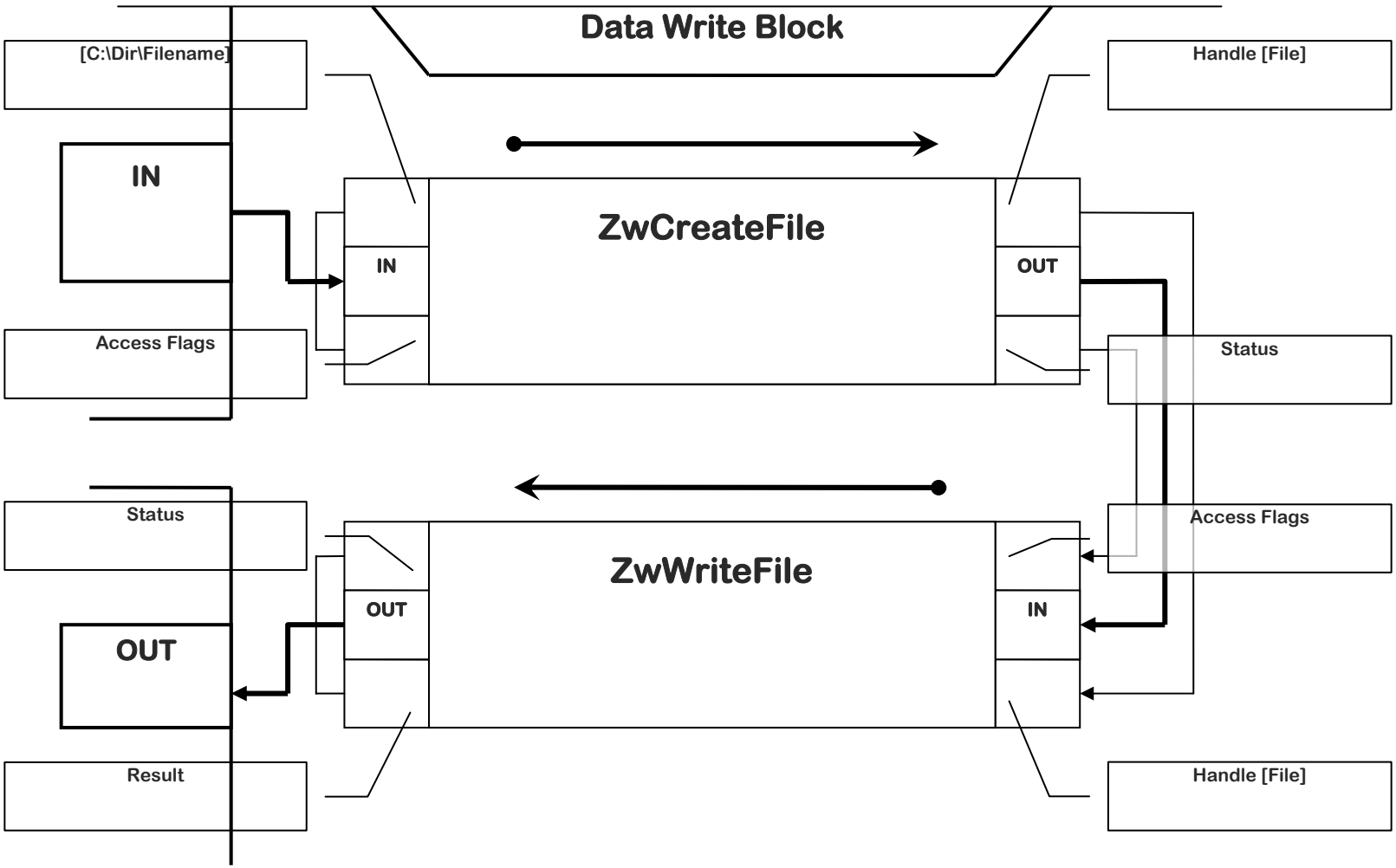
Advantages of our approach:



GSR Structure



GSR Block Structure



Sample Replication Structure

$$G = \{V_N, V_T, P, S\}$$

$$V_N = \begin{Bmatrix} \langle \text{Gene\_of\_self\_replication} \rangle, \langle \text{File\_Search\_Block} \rangle, \\ \langle \text{File\_Copy\_Block} \rangle, \langle \text{Directory\_System\_Call} \rangle, \\ \langle \text{Open\_File\_System\_Call} \rangle, \langle \text{Create\_File\_System\_Call} \rangle, \\ \langle \text{Write\_File\_System\_Call} \rangle \end{Bmatrix}$$

$$V_T = \begin{Bmatrix} ZwQueryDirectoryFile(...), ZwOpenFile(...), \\ ZwCreateFile(...), ZwWriteFile(...) \end{Bmatrix}$$

Sample Replication Structure

$$Gene \rightarrow File\_Search\_Block \cdot File\_Copy\_Block$$

$$File\_Search\_Block \rightarrow Directory\_System\_Call \cdot Open\_File\_System\_Call$$

$$File\_Copy\_Block \rightarrow Create\_File\_System\_Call \cdot Write\_File\_System\_Call$$

$$Directory\_System\_Call \rightarrow input_1 \cdot ZwQueryDirectoryFile \cdot output_1$$

$$Open\_File\_System\_Call \rightarrow input_2 \cdot ZwOpenFile \cdot output_2$$

$$Create\_File\_System\_Call \rightarrow input_3 \cdot ZwCreateFile \cdot output_3$$

$$Write\_File\_System\_Call \rightarrow input_4 \cdot ZwWriteFile \cdot output_4$$

Sample Replication Structure

$$\delta(Gene, ZwQueryDirectoryFile) = \{File\_Search\_Block\}$$
$$\delta(Gene, ZwOpenFile) = \{File\_Search\_Block\}$$
$$\delta(Gene, ZwCreateFile) = \{File\_Copy\_Block\}$$
$$\delta(Gene, ZwWriteFile) = \{File\_Copy\_Block\}$$
$$\delta(File\_Search\_Block, ZwQueryDirectoryFile) = \{Directory\_System\_Call\}$$
$$\delta(File\_Search\_Block, ZwOpenFile) = \{Open\_File\_System\_Call\}$$
$$\delta(File\_Copy\_Block, ZwCreateFile) = \{Create\_File\_System\_Call\}$$
$$\delta(File\_Copy\_Block, ZwWriteFile) = \{Write\_File\_System\_Call\}$$
$$\delta(File\_Search\_Block, ZwCreateFile) = \delta(File\_Search\_Block, WriteFile) = \mathrm{O}$$
$$\delta(File\_Copy\_Block, ZwQueryDirectoryFile) = \delta(File\_Copy\_Block, ZwOpenFile) = \mathrm{O}$$

Replication in Malware

Worm Xanax

NtOpenFile 100020h, {24, 0, 42h, 0, 0, "\??\c:\Virlab\"}, 3, 33 ... 12, 0h, 1) result = 0	1
NtCreateFile 80100080h, {24, 12, 42h, 0, 1243404, "xanax.exe"}, 0h, 128, 3, 1, 96, 0, 0 ... 68, 0h, 1) result = 0	2

System Call	Input Arguments	Output Args	
NtOpenFile 0x100001	{24, 0, 0x40, 0, 0, "\??\C:\WINDOWS\""}, 3, 16417	12, {0x0,1}	3
NtQueryDirectoryFile 12	0, 0, 0, 1243364, 616, 3, 1, "<.exe", 0	{0x0,110}	4

Replication in Malware

Worm

Xanax

(cont)

System Call	Input Arguments	Output Args	
NtCreateSection 0xf001f	0h, 0h, 2, 134217728, 68	72	5
NtMapViewOfSection 72	-1, 0h, 0, 0, {0, 0}, 0, 1, 0, 2	0x980000, 0,0,368 64	6

Virus Handle

Section Handle

System Call	Input Arguments	Output Args	
NtCreateFile 0x40110080	{24, 0, 40h, 0, 1242788, "\??\C:\WINDOWS\calc.exe", 0h, 32, 0, 5, 100, 0, 0	52, {0h,3}	7
NtSetInformationFile 52	1241948, 8, 20	{0h,0}	8
NtWriteFile 52	0, 0, 0, "MZ\220\0\3\0\0\0\4\0\0\0\377\37.....\0\0\0", 33792 0h, 0	{0h, 33792}	9

Victim File

End Of File

Viral Code

Code Size

Virus Replication Data

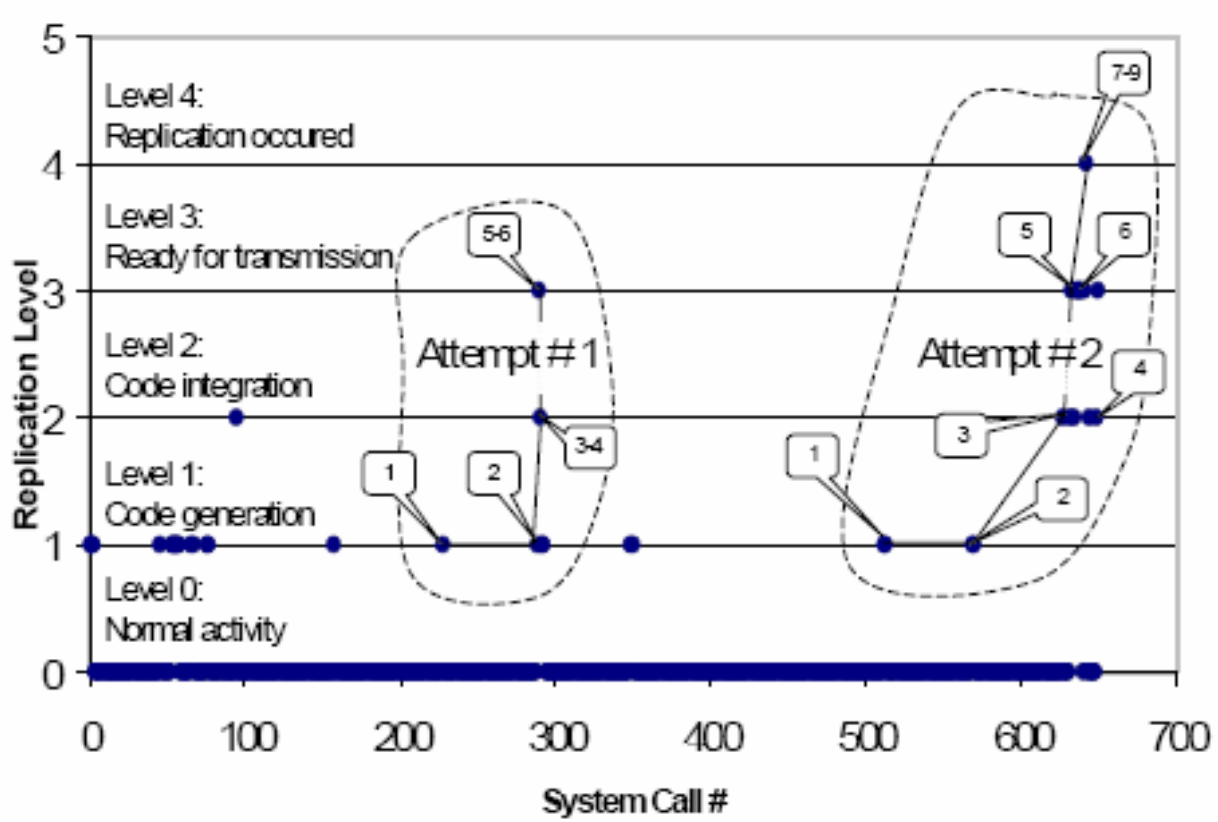


Fig. 4. Sample Virus Replication Data (648 points, 2 attempts).

Extra Features: Weights

Each individual GSR component is assigned with a weight

$$W_B = \sum (W_{B_1}, W_{B_2}, W_{B_3}, \ldots, W_{B_n}) +$$
$$+ \sum (W_{B_{bind\,(1\leftrightarrow 2)}}, W_{B_{bind\,(2\leftrightarrow 3)}}, W_{B_{bind\,(3\leftrightarrow 4)}}, \ldots, W_{B_{bind\,(n-1\leftrightarrow n)}}) +$$
$$+ \sum (W_{B_{1in}}, W_{B_{1out}}, W_{B_{2in}}, W_{B_{2out}}, \ldots, W_{B_{nin}}, W_{B_{nout}}) + W_{\mathrm{Re}\,sult}$$

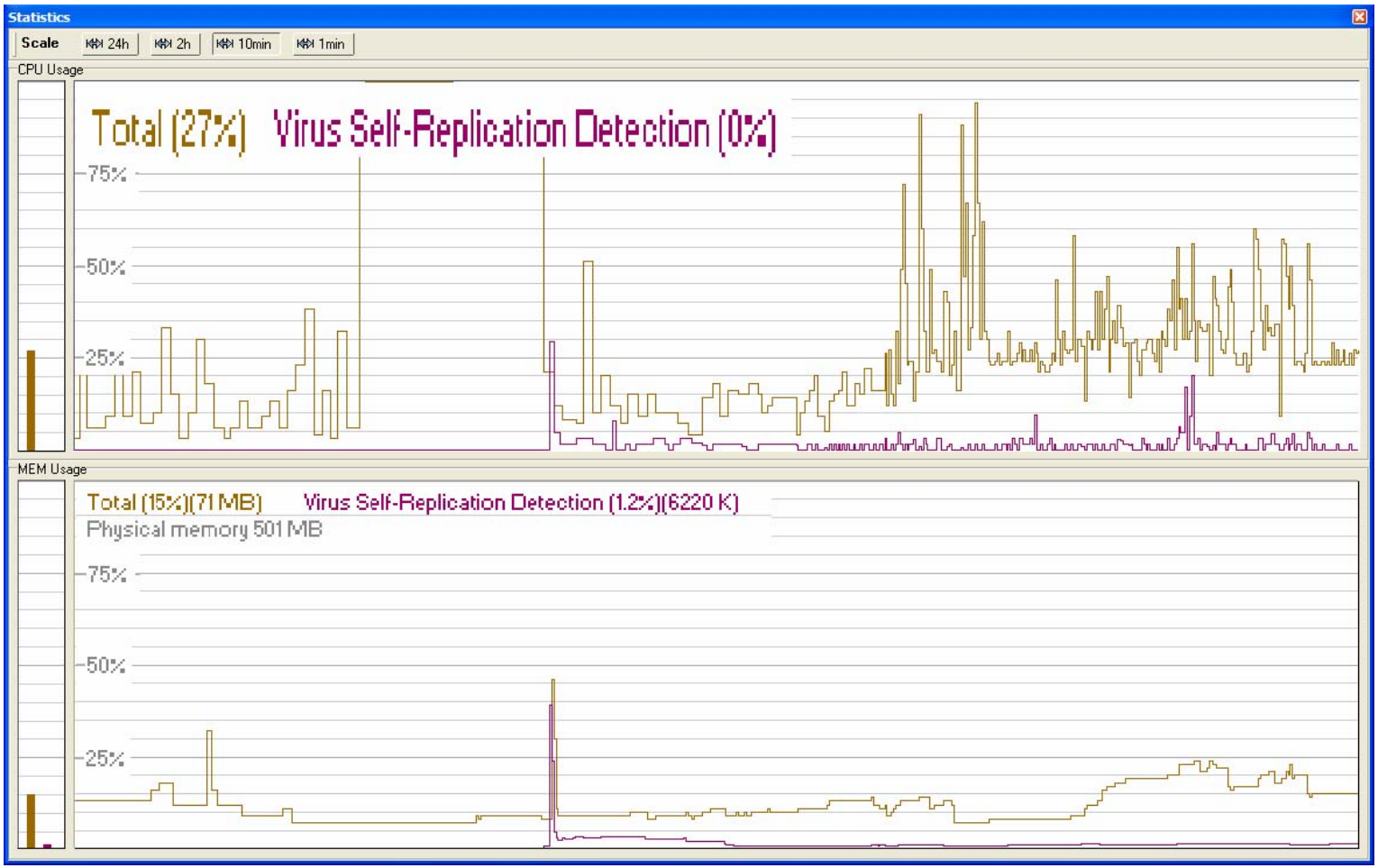
Normalized Replication Score is computed as follows:

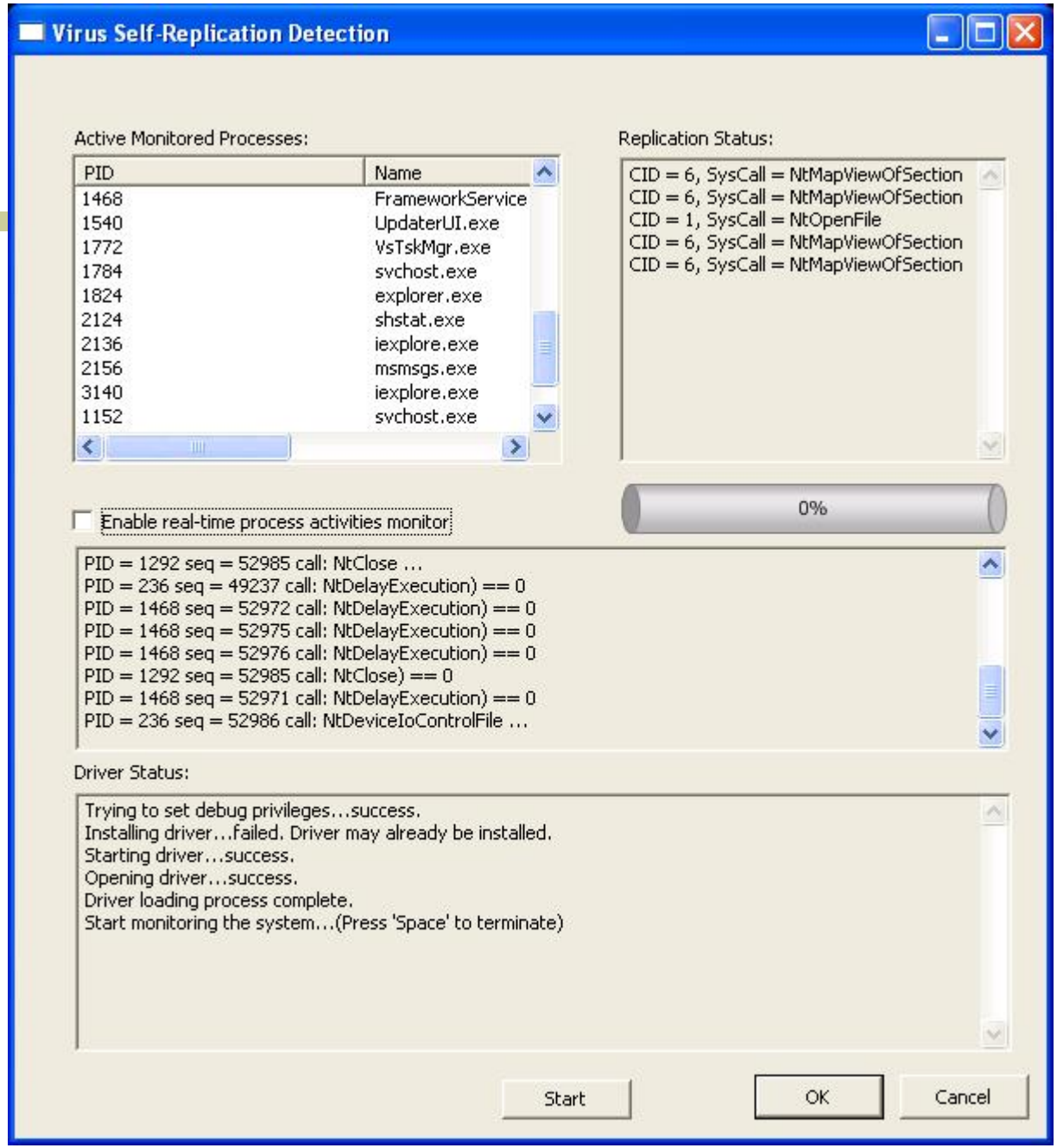
$$R_{norm} = \frac{(R + \sum (W_{rB_1}, W_{rB_2}, \ldots, W_{rB_n}))}{N} \cdot 100\%$$

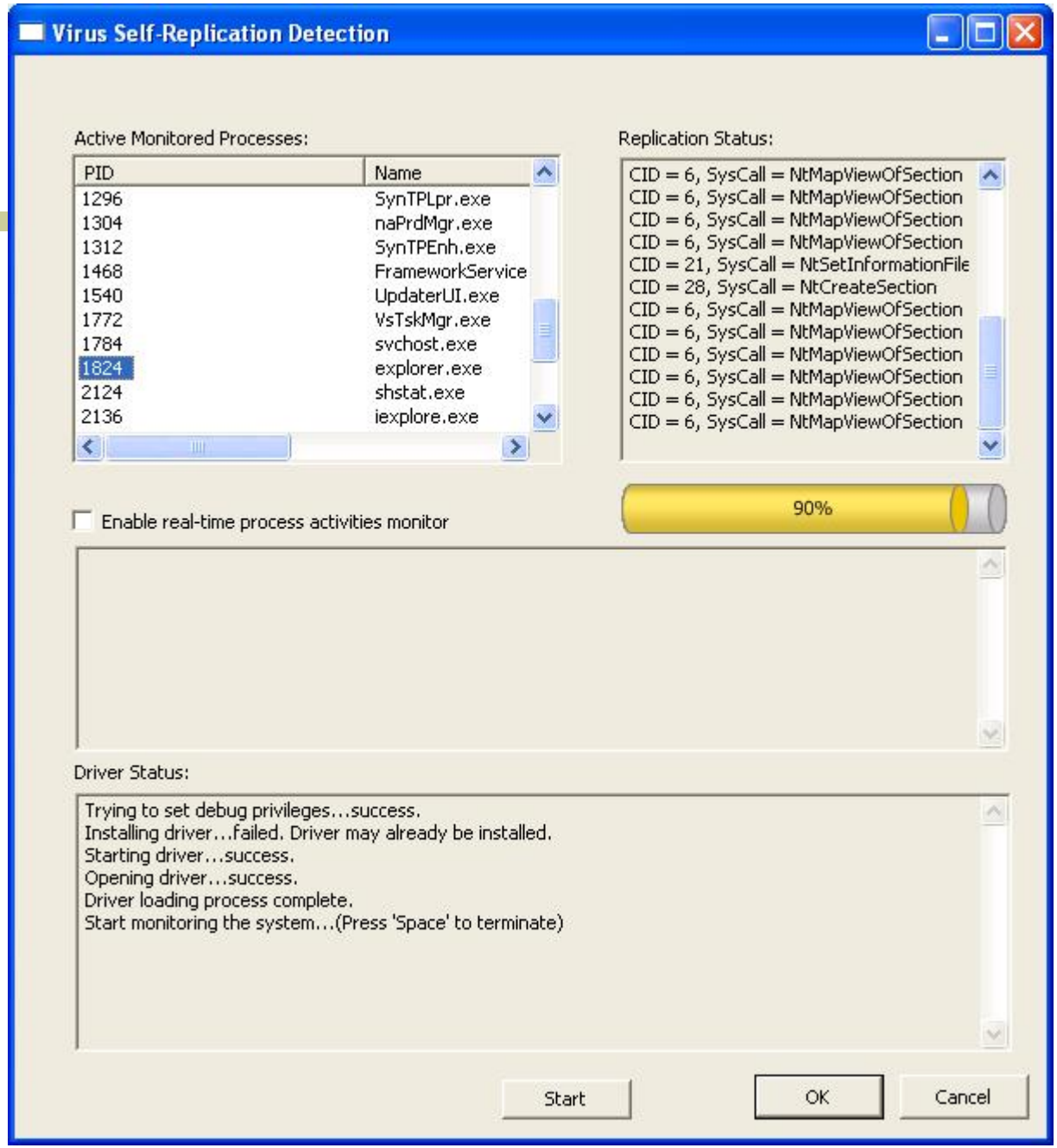
Replication Rates

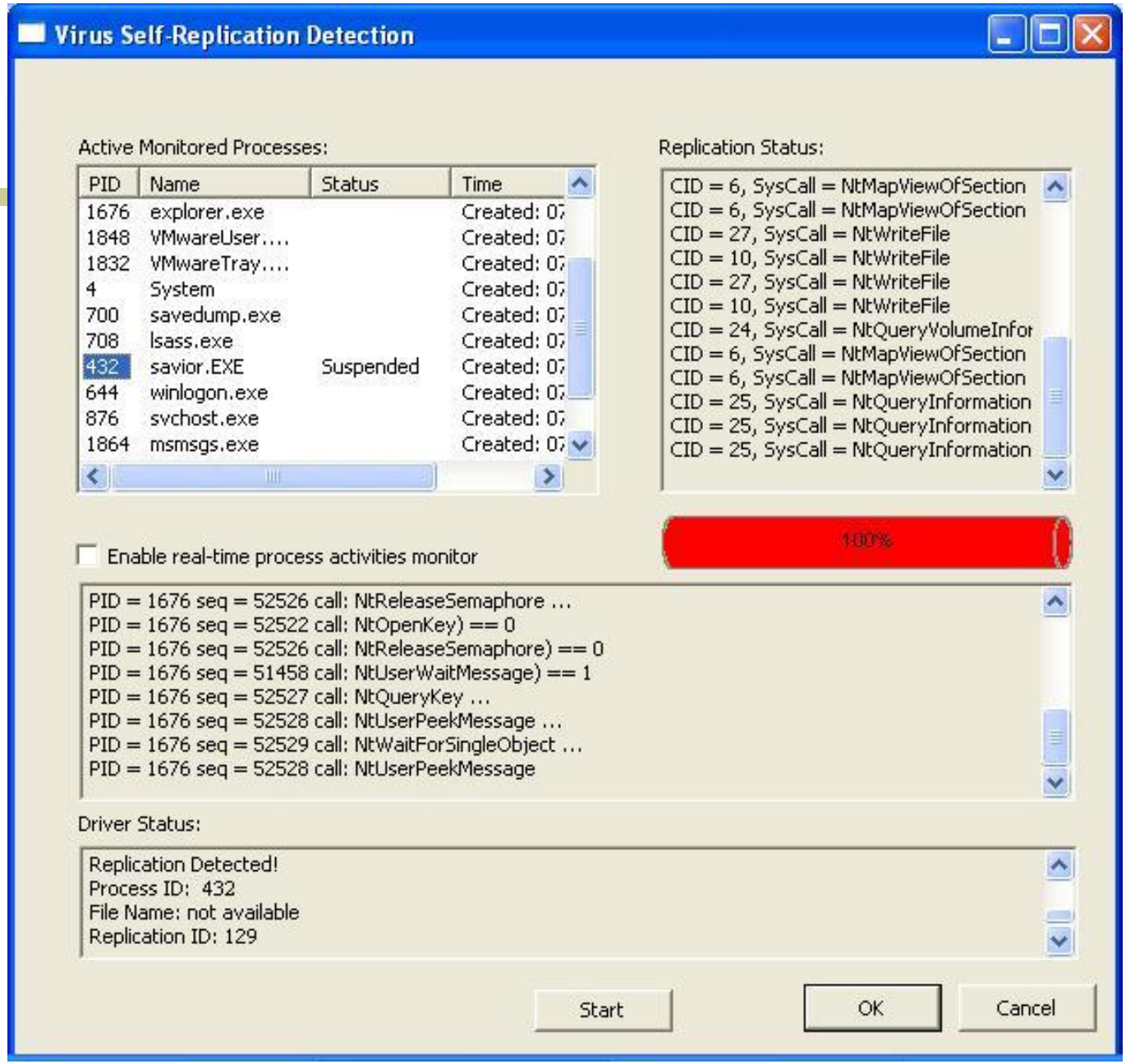
	Host Search	File Access	Networking	Memory	Injection / infection	Normalied Replication (total)
W32.Alicia	100%	100%	100%	32.4%	100%	100%
W32.Bogus	100%	100%	5.3%	3.7%	100%	100%
W32.Crash	100%	100%	0%	100%	100%	100%
W32.Neo	100%	100%	7.0%	100%	100%	100%
W32.Linda	100%	100%	4.3%	100%	100%	100%
W32.Stream	100%	100%	32.5%	100%	100%	100%
Svchost.exe	26.3%	100%	79.4%	100%	36.0%	78.4%
Explorer.exe	14.5%	92.1%	100%	84.5%	47.4%	86.2%

Overheads


Statistics
Scale
24h
2h
10min
1min
CPU Usage
Total (27%)
Virus Self-Replication Detection (0%)
75%
50%
25%
MEM Usage
Total (15%)(71 MB)
Virus Self-Replication Detection (1.2%)(6220 K)
Physical memory 501 MB
75%
50%
25%


Virus Self-Replication Detection
Active Monitored Processes:
PID
Name
1468 FrameworkService
1540 UpdaterUI.exe
1772 VsTskMgr.exe
1784 svchost.exe
1824 explorer.exe
2124 shstat.exe
2136 iexplore.exe
2156 msmsgs.exe
3140 iexplore.exe
1152 svchost.exe
Replication Status:
CID = 6, SysCall = NtMapViewOfSection
CID = 6, SysCall = NtMapViewOfSection
CID = 1, SysCall = NtOpenFile
CID = 6, SysCall = NtMapViewOfSection
CID = 6, SysCall = NtMapViewOfSection
Enable real-time process activities monitor
0%
PID = 1292 seq = 52985 call: NtClose ...
PID = 236 seq = 49237 call: NtDelayExecution) == 0
PID = 1468 seq = 52972 call: NtDelayExecution) == 0
PID = 1468 seq = 52975 call: NtDelayExecution) == 0
PID = 1468 seq = 52976 call: NtDelayExecution) == 0
PID = 1292 seq = 52985 call: NtClose) == 0
PID = 1468 seq = 52971 call: NtDelayExecution) == 0
PID = 236 seq = 52986 call: NtDeviceIoControlFile ...
Driver Status:
Trying to set debug privileges...success.
Installing driver...failed. Driver may already be installed.
Starting driver...success.
Opening driver...success.
Driver loading process complete.
Start monitoring the system...(Press 'Space' to terminate)
Start
OK
Cancel


Virus Self-Replication Detection
Active Monitored Processes:
PID
Name
1296 SynTPLpr.exe
1304 naPrdMgr.exe
1312 SynTPEnh.exe
1468 FrameworkService
1540 UpdaterUI.exe
1772 VsTskMgr.exe
1784 svchost.exe
1824 explorer.exe
2124 shstat.exe
2136 iexplore.exe
Replication Status:
CID = 6, SysCall = NtMapViewOfSection
CID = 6, SysCall = NtMapViewOfSection
CID = 6, SysCall = NtMapViewOfSection
CID = 6, SysCall = NtMapViewOfSection
CID = 21, SysCall = NtSetInformationFile
CID = 28, SysCall = NtCreateSection
CID = 6, SysCall = NtMapViewOfSection
CID = 6, SysCall = NtMapViewOfSection
CID = 6, SysCall = NtMapViewOfSection
CID = 6, SysCall = NtMapViewOfSection
CID = 6, SysCall = NtMapViewOfSection
CID = 6, SysCall = NtMapViewOfSection
Enable real-time process activities monitor
90%
Driver Status:
Trying to set debug privileges...success.
Installing driver...failed. Driver may already be installed.
Starting driver...success.
Opening driver...success.
Driver loading process complete.
Start monitoring the system...(Press 'Space' to terminate)
Start
OK
Cancel


Virus Self-Replication Detection
Active Monitored Processes:
PID Name Status Time
1676 explorer.exe Created: 07
1848 VMwareUser.... Created: 07
1832 VMwareTray.... Created: 07
4 System Created: 07
700 savedump.exe Created: 07
708 lsass.exe Created: 07
432 savior.EXE Suspended Created: 07
644 winlogon.exe Created: 07
876 svchost.exe Created: 07
1864 msmsgs.exe Created: 07
Replication Status:
CID = 6, SysCall = NtMapViewOfSection
CID = 6, SysCall = NtMapViewOfSection
CID = 27, SysCall = NtWriteFile
CID = 10, SysCall = NtWriteFile
CID = 27, SysCall = NtWriteFile
CID = 10, SysCall = NtWriteFile
CID = 24, SysCall = NtQueryVolumeInfor
CID = 6, SysCall = NtMapViewOfSection
CID = 6, SysCall = NtMapViewOfSection
CID = 25, SysCall = NtQueryInformation
CID = 25, SysCall = NtQueryInformation
CID = 25, SysCall = NtQueryInformation
100%
Enable real-time process activities monitor
PID = 1676 seq = 52526 call: NtReleaseSemaphore ...
PID = 1676 seq = 52522 call: NtOpenKey) == 0
PID = 1676 seq = 52526 call: NtReleaseSemaphore) == 0
PID = 1676 seq = 51458 call: NtUserWaitMessage) == 1
PID = 1676 seq = 52527 call: NtQueryKey ...
PID = 1676 seq = 52528 call: NtUserPeekMessage ...
PID = 1676 seq = 52529 call: NtWaitForSingleObject ...
PID = 1676 seq = 52528 call: NtUserPeekMessage
Driver Status:
Replication Detected!
Process ID: 432
File Name: not available
Replication ID: 129
Start
OK
Cancel

Conclusion

- Monitoring and analysis of system calls at runtime is an affordable technology providing unambiguous insight into what the software actually does, including the self-replication indicative of malicious behavior.
- System calls analysis must include arguments analysis for correct behavior detection
- More work to be done to protect the detector
- Additional replication schemes may be introduced for new virus concepts
- Correct GSR definition is the key for keeping false positives down

Questions

Thank you!