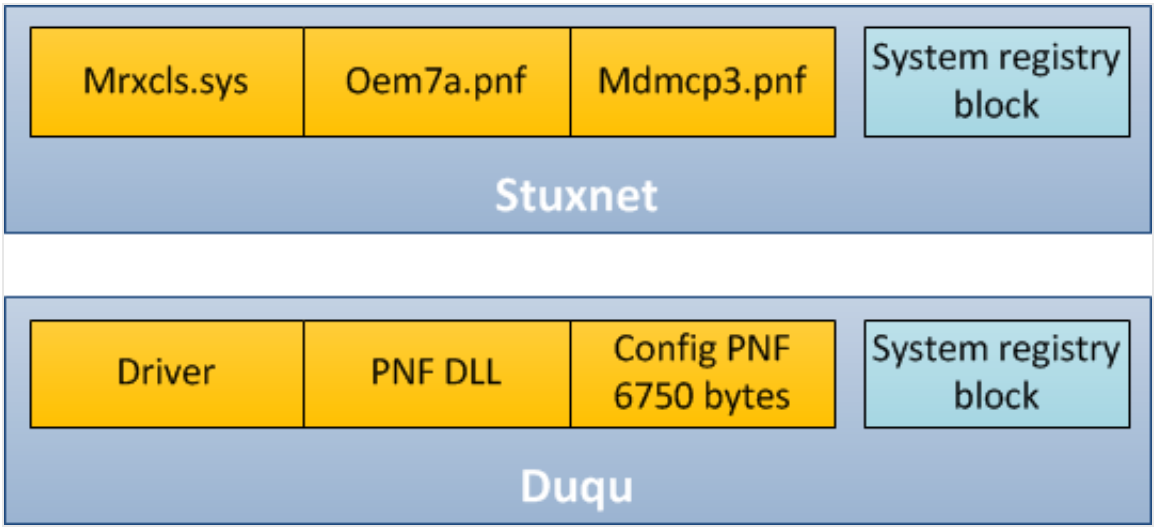
Stuxnet/Duqu: The Evolution of Drivers

We have been studying the Duqu Trojan for two months now, exploring how it emerged, where it was distributed and how it operates. Despite the large volume of data obtained (most of which has yet to be published), we still lack the answer to the fundamental question - who is behind Duqu?

In addition, there are other issues, mostly to do with the creation of the Trojan, or rather the platform used to implement Duqu as well as Stuxnet.

In terms of architecture, the platform used to create Duqu and Stuxnet is the same. This is a driver file which loads a main module designed as an encrypted library. At the same time, there is a separate configuration file for the whole malicious complex and an encrypted block in the system registry that defines the location of the module being loaded and name of the process for injection.



Conventional platform architecture for Stuxnet and Duqu

This platform can be conventionally named as 'Tilded' as its authors are, for some reason, inclined to use file names which start with "~d".

We believe Duqu and Stuxnet were simultaneous projects supported by the same team of developers.

Several other details have been uncovered which suggest there was possibly at least one further spyware module based on the same platform in 2007-2008, and several other programs whose functionality was unclear between 2008 and 2010.

These facts significantly challenge the existing "official" history of Stuxnet. We will try to cover them in this publication, but let us first recap the story so far.

The 'official' Stuxnet story

Let me start with a question: how many Stuxnet driver files are known? As of today, the correct answer would be **four**. See below for more information about them.

File name	Size (bytes)	Compilation date	Where and when it was used	Digital signature/signing date
Mrxcls.sys	19840	01.01.2009	Stuxnet (22.06.2009)	No
Mrxcls.sys	26616	01.01.2009	Stuxnet (01.03.2010/14.04.2010)	Realtek, 25.01.2010
Mrxnet.sys	17400	25.01.2010	Stuxnet (01.03.2010/14.04.2010)	Realtek, 25.01.2010
Jmidebs.sys	25552	14.07.2010	Presumably, Stuxnet	Jmicron, unknown

The first modification of the Stuxnet worm, created in 2009, used only one driver file - mrxcls.sys without a digital signature.

In 2010, the authors created the second driver mrxnet.sys (to hide the worm's component files on USB drives) and equipped mrxnet.sys and mrxcls.sys drivers with digital certificates from Realtek. The mrxnet.sys driver is of no great significance to our story, as it is a separate module not included into the general architecture of the platform.

On 17 July 2010, ESET detected another driver "in the wild" - jmidebs.sys - which was very similar to the already known mrxcls.sys, but had been created just three days before it was discovered. This driver was backed with a new certificate - this time from Jmicron.

Until recently it was unclear what the purpose of this file was, but popular opinion held that it was related to Stuxnet. One theory is that the Stuxnet C&C was trying to replace the old version with the Realtek certificate with a new one. In doing so, the authors of the worm were either hoping to prevent it being picked up by antivirus programs, or were replacing a certificate which had been blocked.

Unfortunately, this theory has not been confirmed - Jmidebs.sys has never been detected anywhere. A new version of Stuxnet capable of installing the file has also not been found.

This is the official history of Stuxnet. However, as I mentioned above, in the course of our research we have discovered new evidence which will be discussed below.

Previously unknown drivers

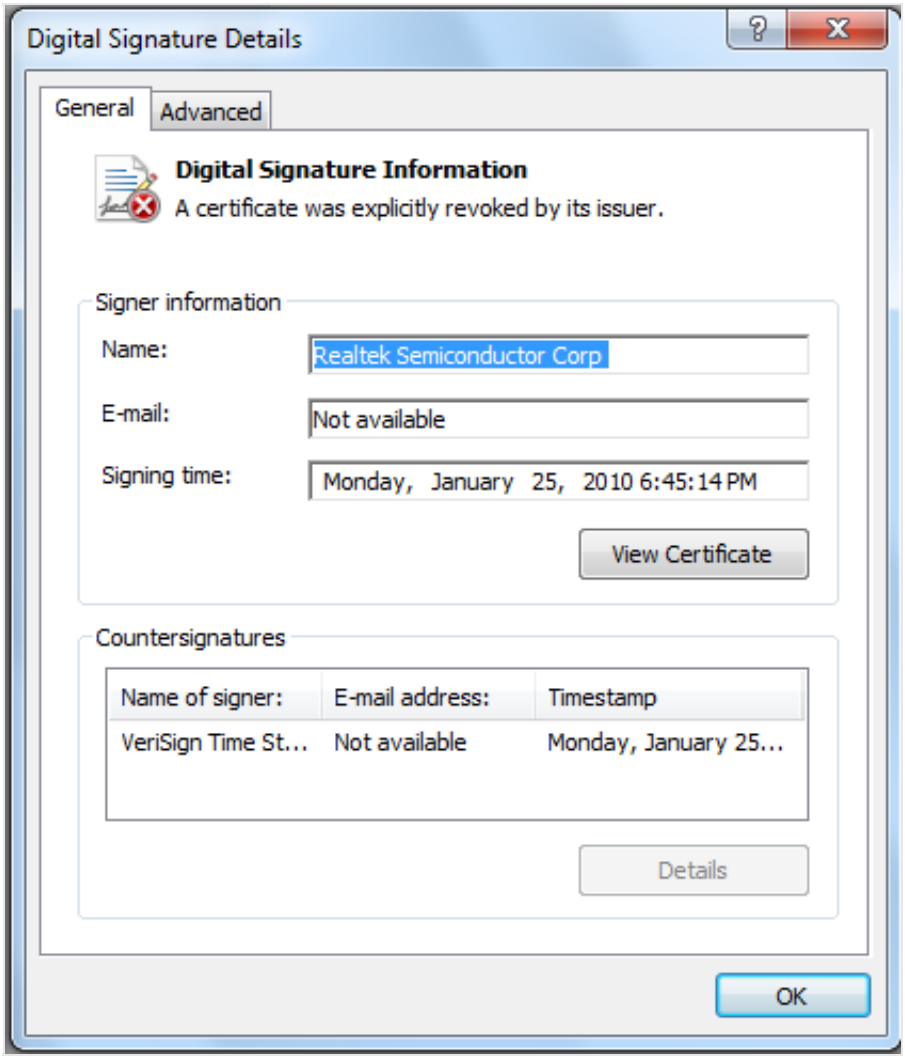
rtniczw.sys

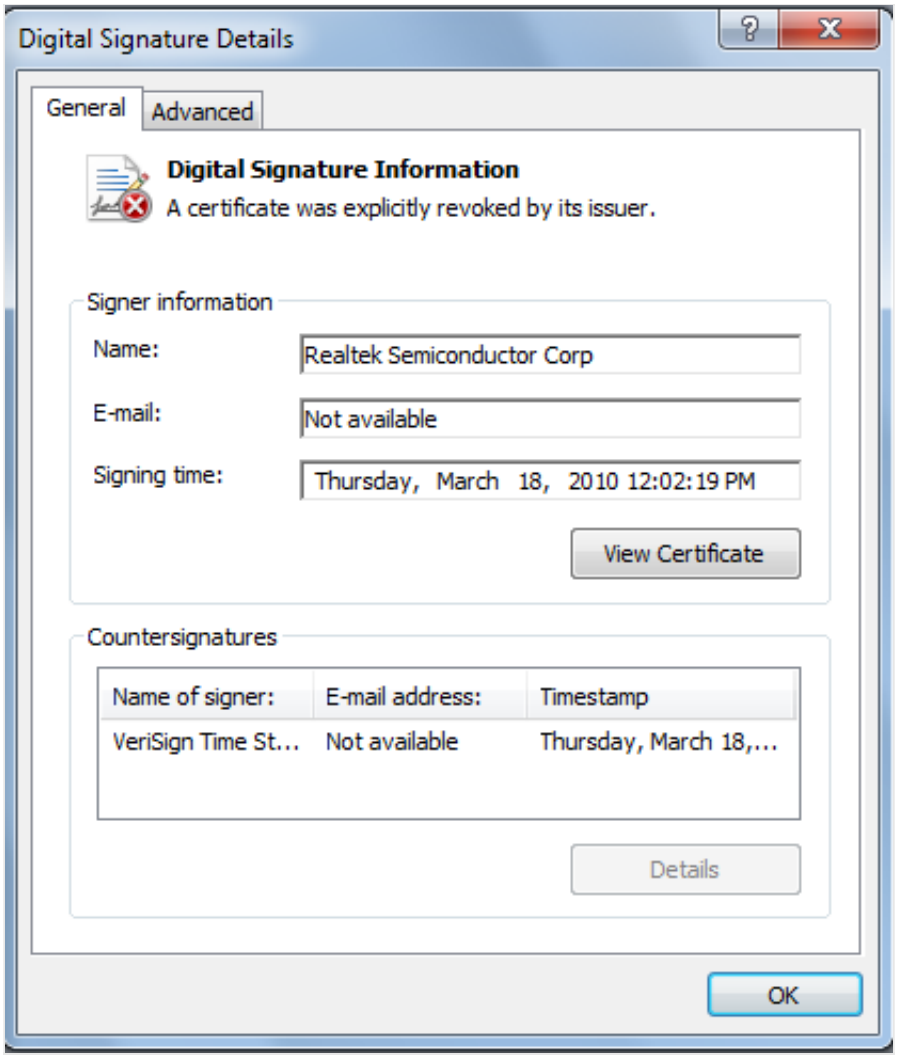
While analyzing a user incident involving Duqu, we discovered something new - something that could, potentially, affect the whole Stuxnet story as we know it.

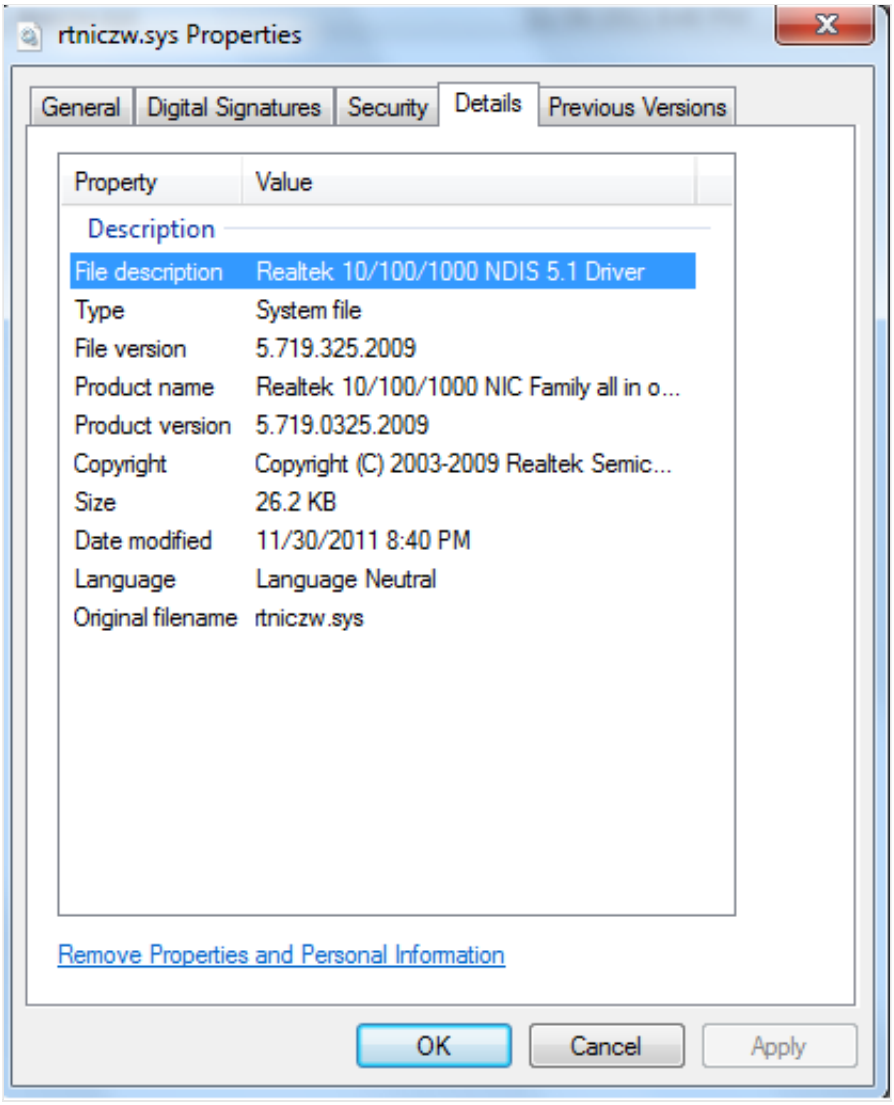
A strange file was discovered on the victim's computer, which was detected by our antivirus engine as Rootkit.Win32.Stuxnet.a. This verdict was supposed to correspond to the known file mrxcls.sys described above, but the detected file's name and checksum were different!

The file was rtniczw.sys, 26,872 bytes in size, MD5 546C4BBEBF02A1604EB2CAAAD4974DE0.

The file was a little larger than mrxcls.sys, which had a Realtek digital signature. This implied that rtniczw.sys also had a digital signature. We managed to get a copy of the file, and we were amazed to find that it used the same Realtek certificate, but with a different file signing date from mrxcls.sys: rtniczw.sys was signed on 18 March 2010, while the mrxcls driver had been signed on 25 January of the same year.




Digital Signature Details
General
Advanced
Digital Signature Information
A certificate was explicitly revoked by its issuer.
Signer information
Name:
Realtek Semiconductor Corp
E-mail:
Not available
Signing time:
Thursday, March 18, 2010 12:02:19 PM
View Certificate
Countersignatures
Name of signer:
E-mail address:
Timestamp
VeriSign Time St...
Not available
Thursday, March 18,...
Details
OK



In addition, rtniczw.sys used a registry key and configuration data block that was not used in Stuxnet. Stuxnet used the key "MRxCls" and the value "Data", while in the case of rtniczw.sys, the key was "rtniczw" and the value was "Config".

Detailed analysis of the code found in rtniczw.sys identified no other differences from the 'reference' driver: this was the same mrxcls.sys file, created in the same year, on the same day and hour - on 1 January 2009.

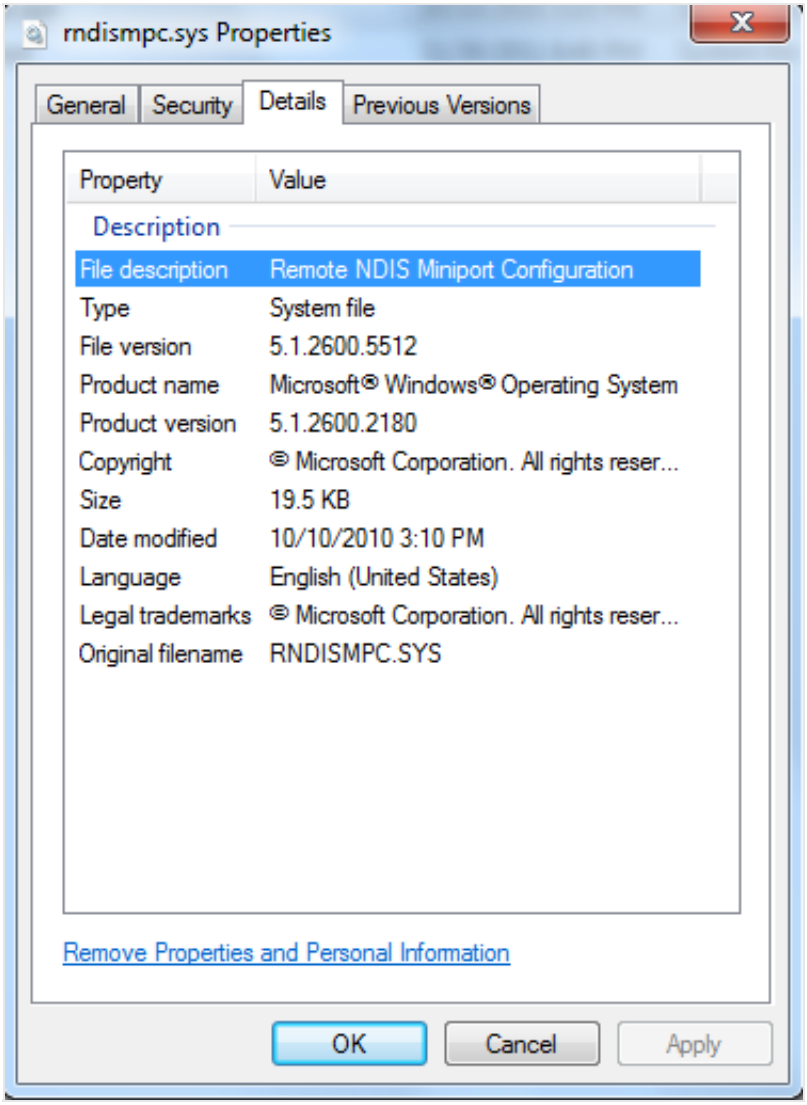
We searched for additional information about other users who had the same file, but were unable to find anything! Moreover, we could find no information at all about the file's name (rtniczw.sys) or its MD5 in any search engine. The file had been identified only once: it had been sent for scanning to VirusTotal from China in May 2011.

Apparently, the system that we were studying had been infected in late August 2011. It should be noted that we did not find a Stuxnet infection on the system: no additional files from the Stuxnet kit had been found. However, we did find Duqu files.

We came to the conclusion that there could be other driver files similar to the "reference" file mrxcls.sys, which are not among known variants of Stuxnet.

rndismpc.sys

A check in our malware collection helped identify another interesting file that was included in the collection over a year ago. The file is named rndismpc.sys, it is 19,968 bytes in size, MD5 9AEC6E10C5EE9C05BED93221544C783E.



This turned out to be another driver, with functionality very nearly identical to that of mrxcls.sys apart from the following exceptions:

1. rndismpc.sys uses a registry key that is different from the keys used by both mrxcls and rtniczw - it is the key "rndismpc" with the value "Action";
2. it uses an encryption key that is different from that used by mrxcls/rtniczw - 0x89CF98B1;
3. the file's compilation date is **20 January 2008**, i.e. a year before mrxcls/rtniczw were created.

Like rtniczw.sys, the file rndismpc.sys had never been encountered on our users' machines. We do not know which malicious program installed it or which main module it was supposed to work with.

The connecting link: mrxcls.sys --> jmidebs.sys -->Duqu drivers

The data obtained and the available information about the drivers used in Duqu (see The Mystery of Duqu, Part One and Part Two) can be summed up in the table below:

Name	Size	Compi-lation date	Main module	Encryption key	Registry key	Value	Device name
rndismpc.sys	19968	20.01. 2008	Unknown	0x89CF98B1	rndismpc	"Action"	"rndismpc"
mrxcls.sys	19840	01.01. 2009	Stuxnet.a	0xAE240682	MRxCls	"Data"	"MRxClsDvX"
mrxcls.sys (signed)	26616	01.01. 2009	Stuxnet.b/.c	0xAE240682	MRxCls	"Data"	"MRxClsDvX"
rtniczw.sys (signed)	26872	01.01. 2009	Unknown	0xAE240682	rtniczw	"Config"	"RealTekDev291"
jmidebs.sys (signed)	25502	14.07. 2010	Unknown	0xAE240682	jmidebs	"IDE"	{3093983-109232-29291}
.sys*	Different	03.11. 2010	Duqu	0xAE240682		"FILTER"	{3093AAZ3-1092-2929-9391}
.sys*	Different	17.10. 2011	Duqu	0x20F546D3		"FILTER"	{624409B3-4CEF-41c0-8B81-7634279A41E5}

*Known Duqu drivers have unique file names for each of the variants. Their functionality, however, is absolutely identical.

According to our analysis, **jmidebs.sys** is the connecting link between mrxcls.sys and the drivers later used in Duqu.

The code of mrxcls and jmidebs drivers is largely similar. Some small differences may be due to different settings and minimal changes in the source code, while the point of the code remains the same.

However, more significant changes can be found in several functions:

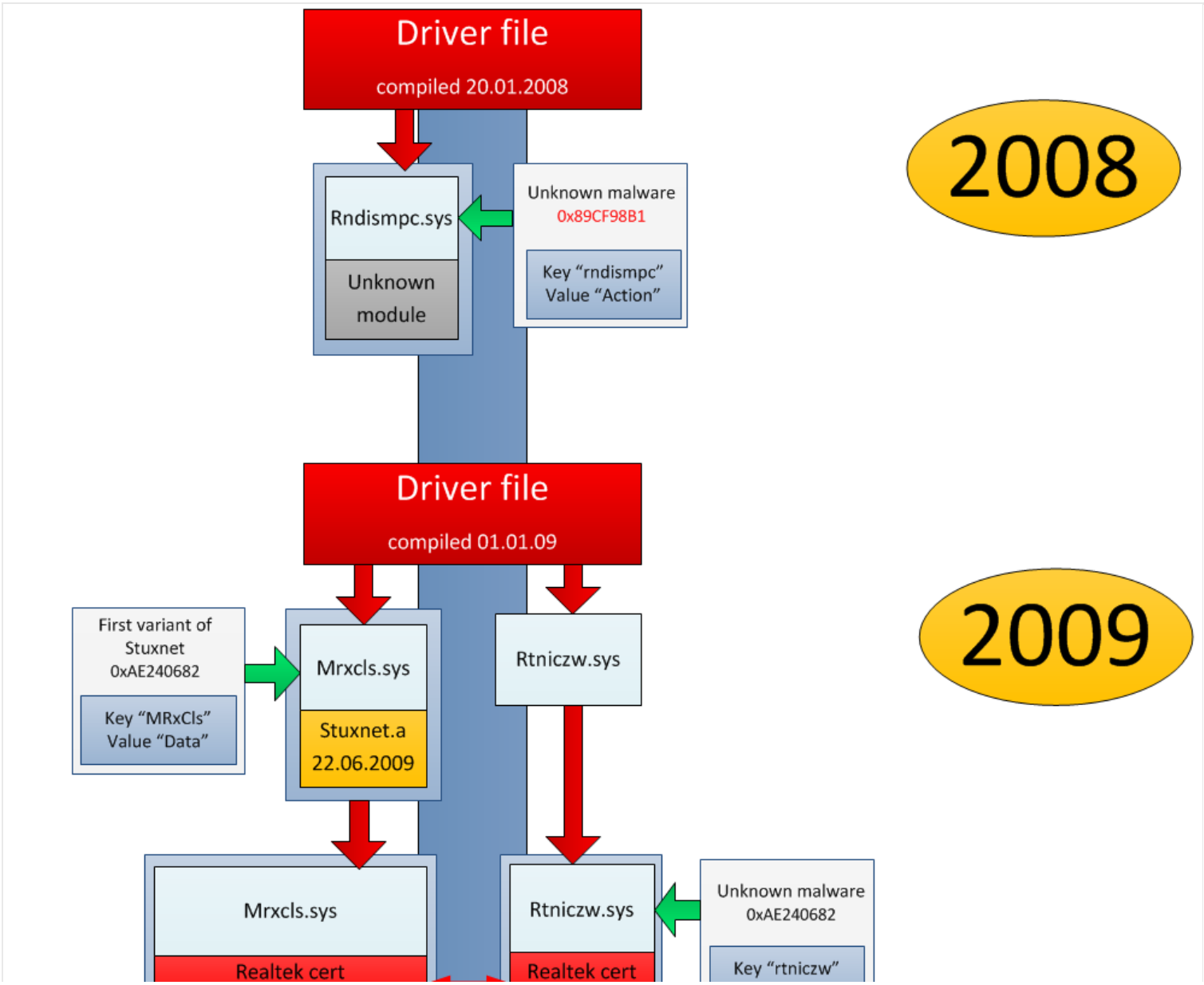
1. The service function which obtains addresses of API functions:
   The version in mrxcls uses the function MmGetSystemRoutineAddress for this purpose and the respective text names of the addresses of incoming API functions. The version in jmidebs calls its own functions to obtain API addresses using hash-sums of their names. The same functions are used in Duqu drivers, while the list of functions' hashes is twice as long.
2. The function which creates stubs to inject PNF DLL into processes:
   The mrxcls version uses a stub with a total size of 6332 bytes.

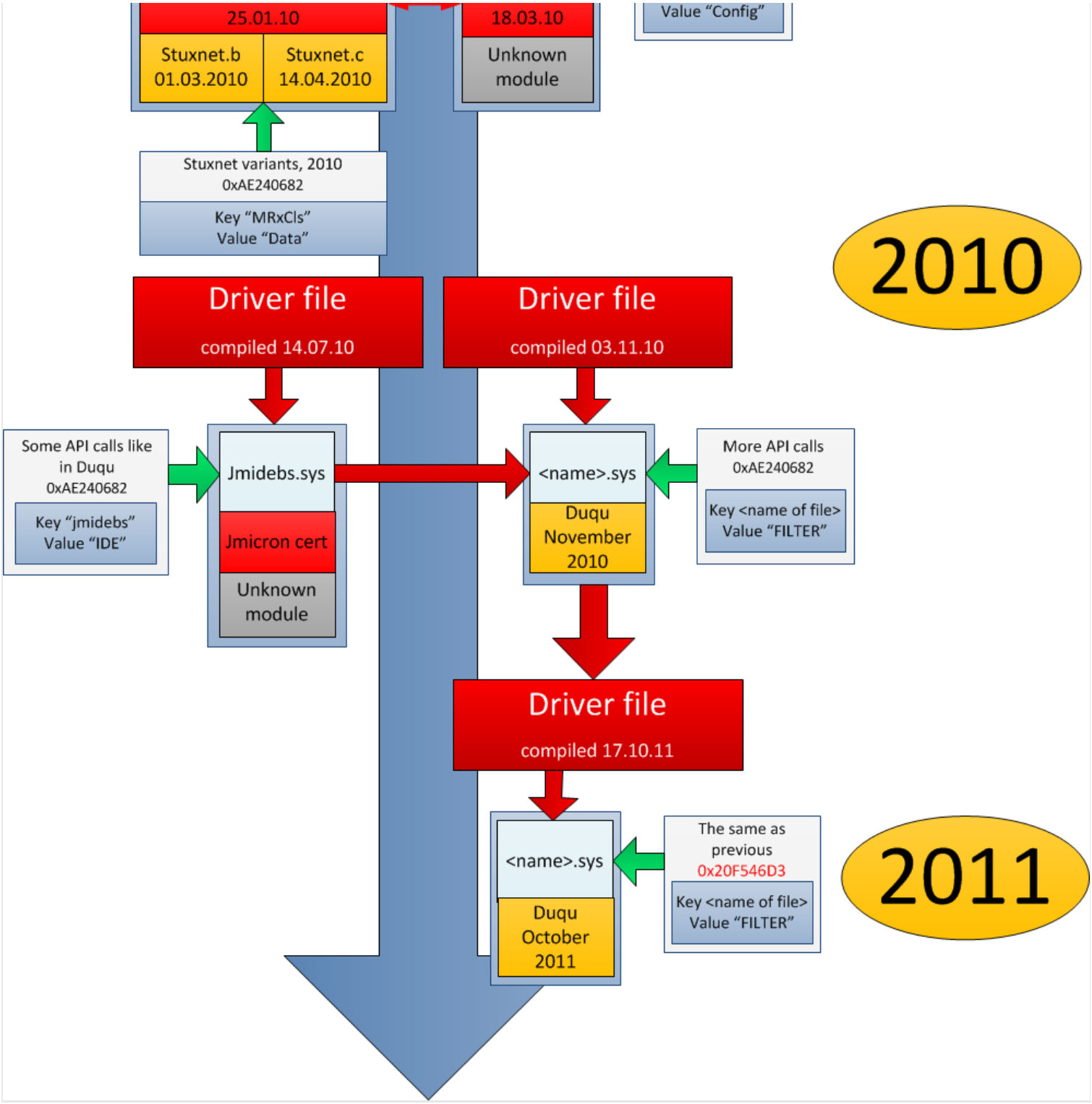
The jmidebs and Duqu drivers use stubs with a total size of 7061 bytes. The code used in the stub modules for these drivers is identical, but has different compilation dates.

	Mrxcls (Stuxnet)	jmidebs	Duqu
API	RtlGetVersion, KeAreAllApcsDisabled, obtained by calling MmGetSystemRoutineAddress	RtlGetVersion, KeAreAllApcsDisabled, PsGetProcessSessionId, PsGetProcessPeb obtained with their own functions	Similar to jmidebs, 4 more functions added
Injected EXE	6332 Jan 01 22:53:23 2009	7061 Jul 14 13:05:31 2010	7061 Different compilation dates

Driver evolution

We have mapped out the links between known drivers whose evolution and key stages of development are easy to track.





Driver evolution from 2008 to 2011

rndismpc.sys, rtniczw.sys and jmidebs.sys

As you can see from the diagram, it is not known which malicious program interacted with the following three drivers: rndismpc.sys, rtniczw.sys and jmidebs.sys. The obvious question would be: **were they used in Stuxnet?** In our opinion, the answer would have to be '**no**'.

First of all, if they had been used in Stuxnet, they would have left a far bigger footprint than the individual cases we have detected. Secondly, there hasn't been a single variant of Stuxnet that is capable of interacting with these drivers.

The file rtniczw.sys was signed on 18 March 2010, but on 14 April 2010 the Stuxnet authors created a new variant of the worm that made use of the "reference" mrxcls.sys. It is obvious that rtniczw.sys was intended for some other use. The same can be said of jmidebs.sys. We believe that the three drivers are only indirectly related to Stuxnet and can safely be erased from Stuxnet history.

Then there is another question: **could these drivers have been used with Duqu?**

There is no clear-cut answer here. Although all known variations of Duqu are from the period November 2010-October 2011, we believe there were earlier versions of the Trojan spy created to steal information. The three drivers in question could easily have been used in early versions of Duqu or with other Trojans based on the Stuxnet/Duqu platform. Like Duqu, those Trojans were most probably used in targeted attacks before the appearance of Stuxnet (dating back to at least 2008), both while it was active and after its C&C was shut down. They were likely to have been parallel projects, and Stuxnet was subsequently based on that accumulated experience and the code that had already been written. It seems highly unlikely that this was the only project that its authors were involved in.

The driver creation process

Let's try to imagine what the driver creation process looks like. A few times a year the authors compile a new version of a driver file, creating a reference file. The primary purpose of this file is to load and execute a main module, which is created separately. It could be Stuxnet, or Duqu or something else.

When it is necessary to use a driver for a new module, the authors use a dedicated program to modify information in the driver's "reference" file, i.e. its name and service information as well as the registry key and its value.

It's important to note that they tweak ready-made files and don't create a new one from scratch. This means they can make as many different driver files as they like, each having exactly the same functionality and creation date.

Depending on the aim of the attack and the Trojan's victim, several driver files can then be signed with legitimate digital certificates whose origins remain unknown.

Conclusion

From the data we have at our disposal, we can say with a fair degree of certainty that the "Tilded" platform

was created around the end of 2007 or early 2008 before undergoing its most significant changes in summer/autumn 2010. Those changes were sparked by advances in code and the need to avoid detection by antivirus solutions. **There were a number of projects involving programs based on the "Tilded" platform throughout the period 2007-2011. Stuxnet and Duqu are two of them - there could have been others, which for now remain unknown.** The platform continues to develop, which can only mean one thing - we're likely to see more modifications in the future.