

Detecting Zeus | eternal-todo.com

Archived: 2026-04-05 18:56:10 UTC

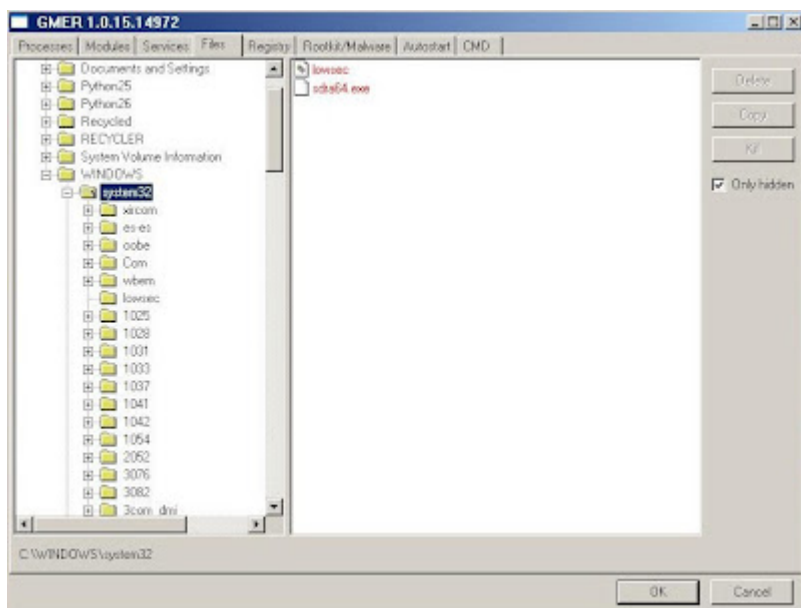
- [Botnet](#)
- [Detection](#)
- [Malware](#)
- [Zeus](#)

In the S21sec blog we [have been talking some time ago](#) about our dear friend, almost one more colleague: Zeus. It is a malware with more than 3 years of life which continues changing and evolving to hide itself better and making the fraud more efficient. But what we maybe have not mentioned yet is how to know if our little friend is here, spying all our movements and reporting all of this to its parents, because sometimes the AV software is not so effective as we expect.

There are several evidences in its different versions which mean that we are infected with Zeus:

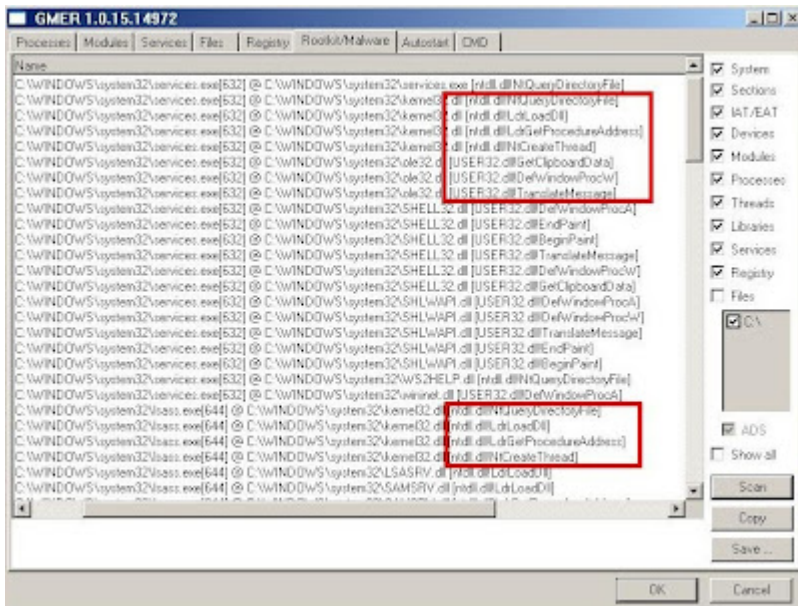
- Filesystem

Zeus leaves a trace in the filesystem when it's installed in the computer, but it hides and blocks all the files it creates, avoiding that a normal user can see and delete them. The solution to find these files is using antirootkit software which will show us the hidden files.



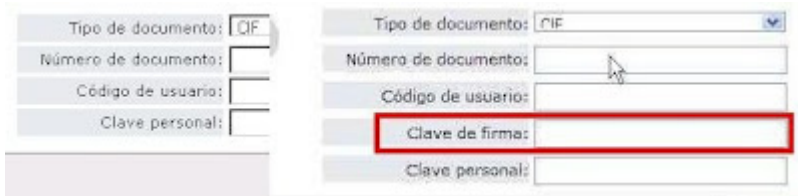
Nowadays the usual binary name is sdr64.exe and its configuration directory c:\windows\system32\lowsec, but this can change depending on the version. [We already mentioned](#) the various names for the configuration files, so now I'll only comment the different names for binary files:

- ntos.exe
- oembios.exe



- Online banking strange behaviour

We can expect that people who use online banking on daily basis can notice some change in the application like an extra field asking for the password needed to make transactions or asking for all the positions of the coordinates card. This is what our trojan makes. Perhaps it will be harder to detect for people who use it occasionally, so the solution here is to pay attention while doing online banking and talk with the bank if there is any suspicious behaviour. This is an example of a login page before and after of Zeus injection:



- Extra parameters (server-side)

Usually when Zeus adds extra fields in the bank page through HTML injection these additional parameters will be sent to the bank server where, depending on the injected code, could be intercepted and being possible to warn the user of a possible infection.

```
338 if(document.teclado.ESpas.value.length<4){alert('Firma electronica no encontrado.')}return false;}document.login.ESpas.value=
  alue;
339
340 ----- 001C -----
341 !
342 name="password"*/TD>*/TD>
343 G
344
345 <TD align="left" colspan="7" valign="bottom"></TD></TR><TR><TD class="textoHome" align="left">3. Clave de Transferencias /TD><TD
  mg4bog/px.gif' border='0' width="20" height="1"></TD><TD align="left"><INPUT type="password" name="ESpas" maxlength="60" tabind
  nido"></TD>+
346 <INPUT type="hidden" name="password">
347 )
348 <INPUT type="hidden" name="ESpas">.
349 <TD align="center"><A href="#" onClick="
350
351 if(document.teclado.ESpas.value.length<4){alert('Firma electronica no encontrado.')}return;}document.datos.ESpas.value=document
352
353 ----- 001D -----
354 name="sl_clave"*/div>
355
356 <div class="fila"><div class="textoform">Firma:</div><div class="inputform"><input type="password" name="ESpas" TABINDEX="3" si
  maxlength="8"></div></div>9
357 <input type="hidden" name="anterior" value="LOGIN">
358 4
359 <input type="hidden" name="ESpas" value="">
360 onclick="javascript:
361
```

- Trojan mutexes

Finally, we can detect the trojan in the system thanks to the mutexes that it creates. For example, with the OpenMutex function we can check if the ZeuS mutexes exist or not, showing the malware trace in the system. Until the moment the mutexes we have seen are:

```
__SYSTEM__64AD0625__
_H_64AD0625_
_AVIRA_2109
_LILO_1909
_SOSI_1909
```

Submitted by jesarpa on Thu, 2009/10/01 - 12:25

Source: <http://eternal-todo.com/blog/detecting-zeus>