-----BEGIN PGP SIGNED MESSAGE-----
Hash: MD5

Adobe Acrobat and PDF security: no improvements for 2 years.
Software released in 2003 contains vulnerabilities disclosured in 2001

July 8, 2003

SUMMARY

In early 2001, we have discovered a serious security flaw in Adobe Acrobat and Adobe Acrobat Reader. In July'2001, we've briefly described it in "eBook Security: Theory and Practice" speech on DefCon security conference. Since there was no reaction from Adobe (though Adobe representative has attended the conference), we have reported this vulnerability to CERT in September'2002 (after more than a year), still not disclosing technical details to the public. Only in March'2003, CERT Vulnerability Note (VU#549913) has been published, and after a week, Adobe has responded officially (for the first time) issuing the Vendor Statement (JSHA-5EZQGZ), promising to fix the problem in new versions of Adobe Acrobat and Adobe Reader software expected in the second quarter of 2003. When these versions became available, we have found that though some minor improvements have been made, the whole Adobe security model is still very vulnerable, and so sent a follow-up to both CERT and Adobe. Both parties failed to respond. Below is the full story.

CONTACT INFORMATION

Name : ElcomSoft Co.Ltd.
E-mail : info@elcomsoft.com <mailto:info@elcomsoft.com>
Web : <http://www.elcomsoft.com>
Phone / fax : +1 866 448-2703

HISTORY

Adobe Systems Inc is referred as "Vendor"
ElcomSoft Co.Ltd. is referred as "Reporter"

07/16/2001: "eBook Security: Theory and Practice" presentation on DefCon 9:
<http://www-2.cs.cmu.edu/~dst/Adobe/Gallery/defcon.ppt>
06/13/2002: Report sent to vendor (PASSKEY:75DF62C56A7DE9F888256BCB0001DF72)
09/10/2002: Report sent to CERT
10/08/2002: More detailed acknowledgment sent to reporter
10/08/2002: Initial attempt to contact vendor via web feedback
10/18/2002: Follow up to PR contact(s); point of contact initiated
10/21/2002: Authentication loop closed; technical details sent
10/29/2002: Ack asked for and received; further details sent related to report
11/21/2002: Reporter asks for status update
11/26/2002: Ping from reporter
11/26/2002: Follow up with vendor to get status of report
11/27/2002: Ack from vendor PR contact asserting more info soon
11/28/2002: Follow up to vendor again asking for confirmation of details;
let the vendor know reporter is willing to wait if details and solution acknowledged
12/02/2002: Conversation with vendor contact verifying details of issue; mention made of issue being resolved in next release
12/04/2002: Initial date identified for potential publication of the report
12/09/2002: Vendor replies that their response is undergoing legal review
12/18/2002: Reporter asks for status update; notes 45-day disclosure period over
12/18/2002: Ack reporter
12/18/2002: Ping vendor for written response again
01/05/2003: Reporter asks for status update
01/14/2003: Ack reporter; tentatively set publication date for 01/20
01/20/2003: Reporter ack
01/21/2003: Private CERT Vulnerability Card published with draft status

03/19/2003: CERT Vulnerability Note (VU#549913) published:
<http://www.kb.cert.org/vuls/id/549913>
03/25/2003: Vendor Statement (JSHA-5EZQGZ) published:
<http://www.kb.cert.org/vuls/id/JSHA-5EZQGZ>
07/02/2003: Updated vulnerability report by reporter to CERT
07/04/2003: Updated vulnerability report sent by reporter to vendor

TECHNICAL DETAILS

Description of the vulnerability

Adobe Acrobat Reader supports plug-ins, i.e. additional modules that extend the functionality of Adobe Acrobat and Adobe Acrobat Reader; plug-ins SDK and plug-ins certification (signing) mechanism are provided. By design, Adobe Acrobat (and Reader) should load only digitally signed plug-ins, while the key (for signing) is provided by Adobe itself -- to developers who has signed a special agreement with Adobe. Besides, some plug-ins are signed by Adobe using their own private Key, and there is a 'certified' (so-called 'trusted') mode in Acrobat, when only Adobe-certified plug-ins are being loaded.

However, the implementation of certification mechanism is weak, and it is easy to write a plug-in that will look like one certified by Adobe, and so will be loaded even in 'certified' mode. Such plug-in can execute ANY code -- i.e. perform file operations (read, write, execute etc), access Windows Registry etc.

At 03/25/2003, vendor (Adobe Systems Inc) issued "Vendor Statement", confirming the existence of the vulnerability:
<http://www.kb.cert.org/vuls/id/JSHA-5EZQGZ>

There, vendor notes:

"The security mechanism for loading certified plug-ins will be updated in an upcoming release of Adobe Acrobat and Adobe Acrobat Reader available in the second quarter of 2003."

In June 2003, new versions of Adobe Reader (6.0) became available, but it is still vulnerable. The details are below.

There are two runtime modes for which they are enabled to load and execute:

- Non-certified mode
- Certified mode

Non-certified mode has not been changed: new versions of Adobe Acrobat and Adobe Reader still load all third party plug-ins that have old signatures, including "forged" ones (as described in VU#549913).

Certified mode has been improved, as promised: now the digital signatures (enabling key) uses stronger algorithms, and so cannot be forged. However, the whole Acrobat security model is still weak, as described below.

Acrobat/Reader could be running in "Certified" mode in two cases:

- "Certified plug-ins only" option was turned on when Acrobat starts. In this case Acrobat/Reader 6 loads only plug-ins with new tamper-resistant certificates, so plug-in with "forged" certificate could not be loaded.

- "Certified plug-ins only" option was turned off, but there is no uncertified plug-ins available to be loaded. In this case Adobe Acrobat loads all available plug-ins (including plug-ins without digital signature at all). Adobe Reader requires all plug-ins to be signed, but does not reject plug-in if it has old-style signature. If all loaded plug-ins are certified by Adobe by new (Acrobat 6+) certification mechanism, Acrobat/Reader automatically switches to "Certified" mode.

Adobe Acrobat contains a special (internal) function that returns active "Certified" status (is all loaded plug-ins are certified or not). Let's call that function "CTIsCertifiedMode". Behavior of all Acrobat components that requires "Certified" mode is based on the value returned by this function.

Therefore, if plug-in with "forged" certificate is loaded, it can patch the code of CTIsCertifiedMode function in memory, and so force Acrobat to believe that it works in "Certified" mode.

It is not a big problem to find CTIsCertifiedMode in memory. Plug-ins gets access to Acrobat/Reader core functions through the set of tables called Host Function Tables (HFTs). One of such tables has the name "CoreTools". The functions referred by that table are not documented by Adobe, but one of the functions within CoreTools HFT is CTIsCertifiedMode.

The impact of this vulnerability is described below.

The impact of the vulnerability
- ------------------------------

There are many Adobe Acrobat and Adobe Reader plug-ins that can load (by design) only in certified mode. One example is all documents protected with "Adobe DRM" security handler (so-called eBooks). Certified more assures that all other plug-ins, loaded with those ones, have been also certified by Adobe.

However, using the vulnerability described above, the plug-in with forged signature can perform virtually everything, including but not limited to:

- removing or modifying any restrictions (from copying text to Clipboard, printing etc) from the documents loaded into Adobe Acrobat or Adobe Reader;
- remove any DRM (Digital Rights Management) schemes from PDF documents, regardless the encryption handler used -- WebBuy, InterTrust DocBox, Adobe DRM (EBX) etc;
- modify or remove digital signatures used within a PDF document;
- affect any/all other aspects of a document's confidentiality, integrity and authenticity.

Systems and configurations that are vulnerable
- ---

Software: Adobe Acrobat 4.x
Adobe Acrobat 5.x
Adobe Acrobat 6.0
Adobe Acrobat Reader 4.x
Adobe Acrobat Reader 5.x
Adobe Reader 6.0

Operating systems: Windows 98
Windows ME
Windows 2000
Windows XP

Possible solutions
- ------------------

Adobe Acrobat and Adobe Reader should NOT be able to load ANY plug-ins that have old (designed for versions 4 and 5) certificates. All plug-ins for version 6 should use new, improved signatures. Besides, Acrobat/Reader should verify the integrity of its own executable code in memory, refusing to run (or just to load plug-ins) if the code has been modified.

CONCLUSION
==========

Unfortunately, Adobe does not pay much attention to vulnerability report. The official response usually is:

"Adobe will evaluate this report, as we do any report we receive. For security reasons, Adobe can't discuss the measures we take as a result. Security is an ongoing effort. We are committed to strengthening the security of our products by using sophisticated, industry-standard levels of software encryption and working with the software community, including 'White Hat' security experts, to incorporate features to advance the quality of our products. However, no software is 100 percent secure from determined hackers."

To implement reliable and secure solutions, it is not enough just to "use sophisticated, industry-standard levels of software encryption" - - it is necessary to use them *properly*. It is well known that the

chain is as weak as it's weakest link.

-----BEGIN PGP SIGNATURE-----
Version: 2.6

iQEVAwUAPwqfHMGOU/F25e8NAQErmggAwtT8+afiS3rFvxX+7QAk4SINVDdZjEQz
Dd2ZZgdLbq0zLWs37CX92F3V6JjvsFFbZXe+NHPZr4begdAnXeF/oV8j8C5rLfs2
YiBmO8SMoS7R/PKWjaJH7/XpCOLbSiyuVH2ndHZYM6U19fp1vE9zfP/6wMq5y24o
FW5+63Bofn5o8V46my0uMPSQJTnrx8NZzprNVI4Jx0OddR6ULi5GF5qLDcOyaj4P
2L9l5pCMCzLZo0iTQb7qPujt+RSNOhmlk8mFfefJ6Pr8E/llY4uqhGLAv774be5m
jfw/78G36PNrN2xGlFJ7ww2yxm/7RI7bBU6HLJpZ2EAhYkfnQwrZXQ==
=O1wf
-----END PGP SIGNATURE-----