Adon'tbe an Adobe victim

An overview of how recent Adobe-related flaws affect your web application

Joshua Stabiner – EY Advanced Security Center

Ernst & Young
Quality In Everything We Do

Agenda

- Introductions
- Background
- Cross-site scripting (PDF)
  - Overview
  - Exploit
  - Mitigation
- Cross-site request forgery (SWF)
  - Overview
  - Exploit
  - Mitigation
- Arbitrary command execution (PDF)
  - Overview
  - Exploit
  - Mitigation
- Questions

Background information

- PDF exploits account for over 80% of all exploits tracked by ScanSafe *(Computerworld*).
- Adobe Flash Player has also been affected.
- The majority of Adobe exploits rely on JavaScript being enabled.
- 107 Adobe vulnerabilities in 2009 were logged into the Common Vulnerabilities and Exposures (CVE) database

Background information

- Many web applications utilize PDFs or SWFs to one degree or another.
- PDF/SWF objects are downloaded to the client and rely entirely on client-side controls (browser, plugin, application, OS etc.) for security as well as functionality.
- The type of browser being used affects the display of PDFs/SWFs as well as the version of the browser plugin, which may differ in versions from the actual application.

Cross site scripting (PDF) - Overview

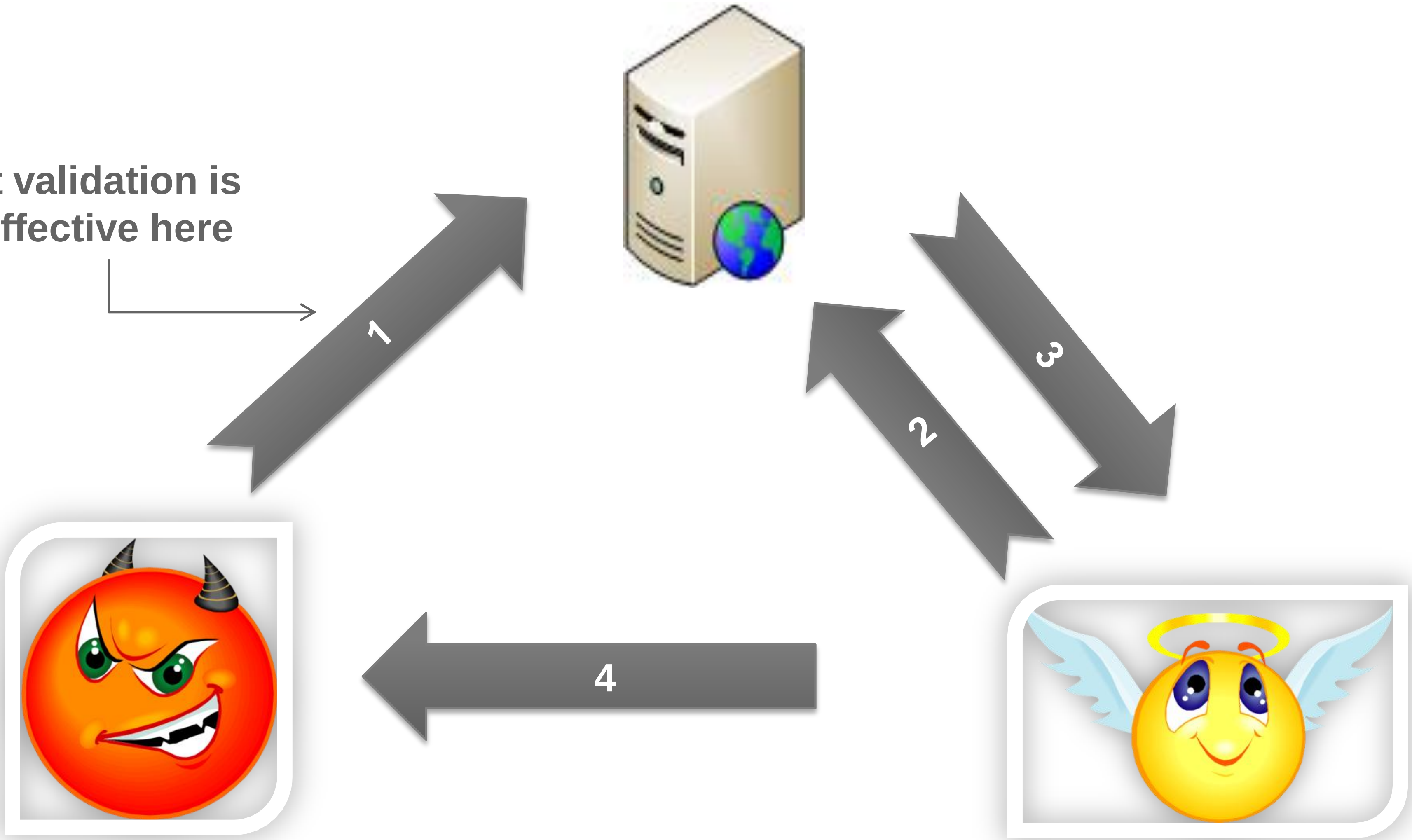
- PDFs are JavaScript enabled: it’s a feature, not a bug – Adobe refuses to disable.
- Victim will *usually* click a link to the PDF document.
- The document itself will often be legitimate.
- The code is executed within the context of the site hosting the document.
- Impossible to detect on the server.

Cross site scripting (PDF) - Exploit

- Adding JavaScript to PDF is simple:
  - Page Action
  - Select JavaScript
- Get the PDF on the server
  - File upload
  - Social engineering
  - Malicious insider
- Just like persistent XSS

Cross site scripting (PDF) - Exploit

Input validation is not effective here



Cross site scripting (PDF) – Mitigation

- Client-side protection
  - Ensure that Adobe, both the Reader and the plugin, are patched and updated to the latest version.
  - Turn off JavaScript in PDF Reader – do you really need it?

- Server-side protection
  - Force PDF documents to be downloaded, instead of displayed in the browser.
  - Keep PDF documents on a separate domain.
  - Review all PDF documents for unwanted JavaScript before hosting.

Cross-site request forgery (SWF) - Overview

- Flash is designed to operate under the restrictions of the Same Origin Policy
  - Prevents a document or script loaded from one origin from getting or setting properties of a document/script from a different origin.
- SWF needs no special extension (or content header) and can even be embedded in other files making it great for file upload functionality.
- Acts like Persistent CSRF
- The Adobe POV – not our problem! (and its not… entirely)

Cross-site request forgery (SWF) - Exploit

- Create a malicious SWF and upload to the server
- Convince victim to load SWF while logged in to target application.
- SWF runs in the background with full access to target application.

Cross-site request forgery (SWF) – Exploit *Gmail Example*

- Mike Bailey’s attack:
  (http://www.foregroundsecurity.com/flash-origin-policy-issues.html)
- Create a gmail account and upload an SWF “attachment”
- Use CSRF to log the victim into the malicious Gmail account. You can then log them out.
- Use social engineering to convince the user to log into Gmail… the SWF now has access to their whole account.
- Let’s see it in action

Cross-site request forgery (SWF) - Mitigation

- Follow file upload leading practices:
  - Bounds Checking should be performed to ensure that uploaded file sizes do not exceed reasonable limits
  - Uploaded files should be placed into a directory that is not web accessible
  - The application should handle all file naming (regardless of the original file name)
  - ***Uploaded files should be hosted on a separate domain to allow the same origin policy to do its job.***
    - Imagine the previous example if the SWF upload was not hosted at mail.google.com

Arbitrary code execution (PDF) - Overview

- There is a stack overflow in the collab.getIcon() function in Acrobat and Acrobat Reader – This is a bug, not a feature!

- The exploit allows an adversary to run arbitrary code on a victim machine.

- Tools such as *metasploit* can easily be used to generate the malicious PDF.

Arbitrary code execution (PDF) - Exploit

- No more stressing over shellcode – Metasploit 1-2-3:
  - Exploits → Adobe collab.getIcon() exploit
  - Set CMD to "cmd.exe /K ipconfig && echo Look what I can do“

- Send corrupt PDF via:
  - Email
  - Application file upload
  - Social engineering

- Once run, game over.

Arbitrary code execution (PDF) - Mitigation

- Run all PDF documents through an anti-virus before producing them to end users.
- Where possible, strip out all JavaScript from the PDF document.
  - Most malicious documents will have a zlib-encoded JavaScript section.
- User education:
  - Patch your Adobe – this issue is fixed already
  - Update your AV and enable live file system auto-protect.
  - Don’t open PDFs (or any file) sent by an unknown party

Going forward

- We like Flash and Acrobat – they're not going away.
- The issues presented are not new:
  - XSS
  - CSRF
  - Buffer overflow

Just new ways of delivering the attacks that evade traditional filters. Almost all are "blended attacks".

- Use common sense, stay up to date and be mindful of the content you host:
  - Where did it come from
  - Is it necessary?
  - Host it safely

Further Reading

- Milw0rm.com – Site containing exploit code.
- http://www.owasp.org/images/7/77/Protecting_Web_Applications_from_Universal_PDF_XSS.ppt
- http://xforce.iss.net/xforce/xfdb/49312 - IBM Internet Security Systems
- http://www.milw0rm.com/exploits/8569 - exploit code
- http://securitylabs.websense.com/content/Blogs/3202.aspx
- http://vrt-sourcefire.blogspot.com/2009/02/have-nice-weekend-pdf-love.html
- http://www.milw0rm.com/exploits/8099
- http://isc.sans.org/diary.html?storyid=6847
- http://www.milw0rm.com/ - interesting exploit site.
- http://www.web2secure.com/2009/05/adobe-reader-exploits-poc.html - proof of concepts
- http://www.gnucitizen.org/blog/danger-danger-danger/ - XSS
- http://blog.trendmicro.com/adobe-reader-vulnerability-actively-being-exploited/
- http://www.foregroundsecurity.com/flash-origin-policy-issues.html

Thanks

- Special thanks to:
  - EY’s Advanced Security Center – particularly Aaron Katz
  - EY Partner Greg Raimann
  - OWASP South Florida leader Rishikesh Pande
  - All of you for coming!

Joshua Stabiner
5 Times Square
New York, NY 10036
Joshua.Stabiner@ey.com

Questions?