Apache-Tomcat と冗長な UTF-8 表現
（CVE-2008-2938 検証レポート）

ＮＴＴ コミュニケーションズ株式会社
ＩＴ マネジメントサービス事業部
セキュリティオペレーションセンタ

2009 年 5 月 26 日

Ver. 1.1

1. 調査概要

Apache-Tomcat に存在する UTF-8 の冗長表現を使った DirectoryTraversal 問題について、調査した結果をここに記す。

- UTF-8 の冗長表現を使った手法であること
- 3Byte の冗長な UTF-8 表現でも、問題が発現すること
- 「.(ピリオド)」の冗長な UTF-8 表現だけではなく、「/(スラッシュ)」の冗長な UTF-8 表現でも、問題が発生する
- Windows 上の Tomcat では、「\(バックスラッシュ){U+005C}」の冗長な UTF-8 表現である「%C1%9C」、「%E0%81%9C」でも発現した
- Windows 上の Tomcat では、「¥(円記号){U+00A5}」である「%A5」、UTF-8 表現である「%C2%A5」、3Byte の冗長な UTF-8 表現である「%E0%82%A5」では発現しなかった
- Apache-Tomcat 上で、CGI(バイナリ) を実行している場合、任意のコマンドが実行される危険性があること
  - 条件によっては、より危険性の高い「コマンド実行」という危険性がある

2. UTF-8 とは

UTF-8(8-bit UCS Transformation Format)とは、UNICODE の表現法のひとつである。
1Byte の UTF-8 は、ASCII コードと同一になるように設計されている。

以下のように、1Byte の文字(7bit ASCII と、8bit 文字(半角カタカナなど))、と 2Byte の文字(UNICODE の漢字など)のビット列を定義する。

1Byte us-ascii 文字	0zzzzzzz
1Byte 8bit 文字	yzzzzzzz
2Byte 文字	xxxxxxxx yzzzzzzz

ただし、(x,y,z) は (0 or 1) とする。

これらの文字は以下のようなエンコード法で表現することで UTF-8 表現となる。

1Byte UTF-8 表現	0zzzzzzz	1Byte us-ascii 文字
2Byte UTF-8 表現	110ooooz 10zzzzzz	1Byte us-ascii 文字
	110oooyz 10zzzzzz	1Byte 8bit 文字
3Byte UTF-8 表現	1110oooo 10oooooz 10zzzzzz	1Byte us-ascii 文字
	1110oooo 10oooyz 10zzzzzz	1Byte 8bit 文字
	1110xxxx 10xxxxyz 10zzzzzz	2Byte 文字

ただし、「o」は、「0(ゼロ)」を示す(UTF-8 上では (0 or 1) の領域であるが)

しかし、赤で書かれた部分については、より短いでバイトで表現できるため、冗長であり、現在は仕様として禁止されている。

この冗長表現を、本文以降では「冗長な UTF-8 表現」と呼ぶことにする。

3. CVE-2008-2938

3.1. 「.(ピリオド)」について

CVE-2008-2938 で指摘されている Apache-Tomcat に存在する UTF-8 の冗長表現を使った Directory Traversal 問題は、「.(ピリオド)」の冗長な UTF-8 表現を使うことで、発現して問題である。

攻撃コード「%C0%AE」は、「0xC0」と「0xAE」という 2Byte に URL デコードされる。
つまり、「0xC0」→「11000000」、「0xAE」→「10101110」である。
これは、2Byte の冗長な UTF-8 表現であり、赤字の部分だけを抜き出し、上位ビットの「0」を省くと、「00000101110」→「0x2E」となる。
「0x2E」は、「.(ピリオド)」の us-ascii 表現である。
以上より、「%C0%AE」は「.(ピリオド)」と同義であるため、「%C0%AE%C0%AE」は「..(ピリオド二個)」と同義になる。(「..(ピリオド二個)」は、上位ディレクトリを示す)

このような冗長な UTF-8 表現によるセキュリティ問題は、過去に Microsoft の Web サーバ IIS に存在した MS00-057 が有名である。

4. CVE-2008-2938 と 3Byte の冗長な UTF-8 表現

3Byte の冗長な UTF-8 表現でも CVE-2008-2938 の問題が発現した。
「.(ピリオド)」つまり、「0x2E」の 3Byte の冗長な UTF-8 表現は、「0xE0」「0x80」「0xAE」となる。
(URL エンコードすると「/%E0%80%AE」)

IDS でカスタム・シグネチャを用意している組織などでは、3Byte の冗長な UTF-8 表現による IDS 回避が行われないように注意することを推奨する。

5. CVE-2008-2938 と「/(スラッシュ)」について

「/(スラッシュ)」は、「0x2F」である。
「/(スラッシュ)」の 2Byte の冗長な UTF-8 表現は「0xC0」「0xAF」である。
また、3Byte の「/(スラッシュ)」の冗長な UTF-8 表現は「0xE0」「0x80」「0xAF」である。
「/(スラッシュ)」の冗長な UTF-8 表現でも、CVE-2008-2938 の問題が発現した。

6. CVE-2008-2938 とファイル読み出しに関する実験結果

以下の環境で、実験を行った。

- CentOS 5.1
- JDK 1.5.0_16
- Apache-Tomcat 5.5.26 および Apache-Tomcat 6.0.16

6.1. Apache-Tomcat 5.5.26 の場合

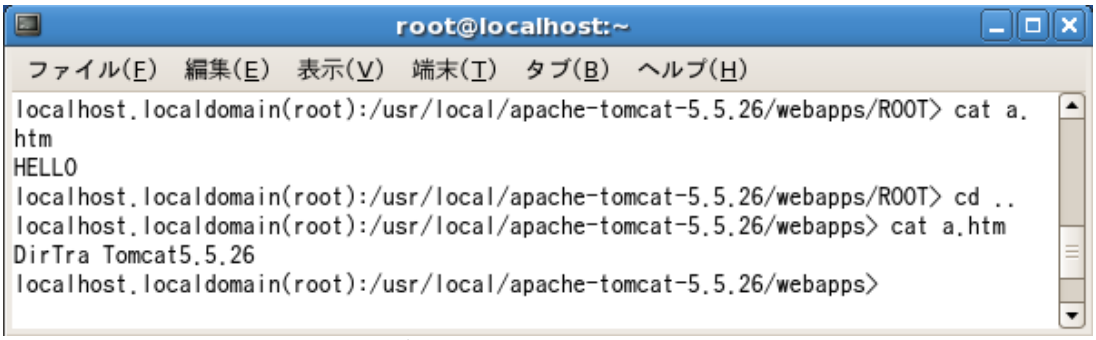

図 6.1-1 : ウェブルートの「a.htm」の中身は「HELLO」。

その一つ上の非公開ディレクトリ上の「a.htm」の中身は「DirTra Tomcat5.5.26」となっている。

CVE-2008-2938 を使って、非公開ディレクトリ上の

「a.htm」が読み出せるかどうかがこの試験内容である

図 6.1-2 : ウェブルート「/a.htm」のアクセス結果。

検査対象の Web サーバ(Tomcat)は正常に動作している

図 6.1-3 : 2Byte の冗長な UTF-8 表現に対しての結果。

CVE-2008-2938 の指摘どおり再現している

図 6.1-4 : 3Byte の冗長な UTF-8 表現でも、問題の再現ができている事が確認できる

図 6.1-5：「.(ピリオド)」ではなく、「/(スラッシュ)」に対しての冗長な UTF-8 表現でも、問題の再現ができている事が確認できる

図 6.1-6：「/(スラッシュ)」の 3Byte の冗長な UTF-8 表現でも、問題の再現ができている事が確認できる

6.2. Apache-Tomcat 6.0.16 の場合

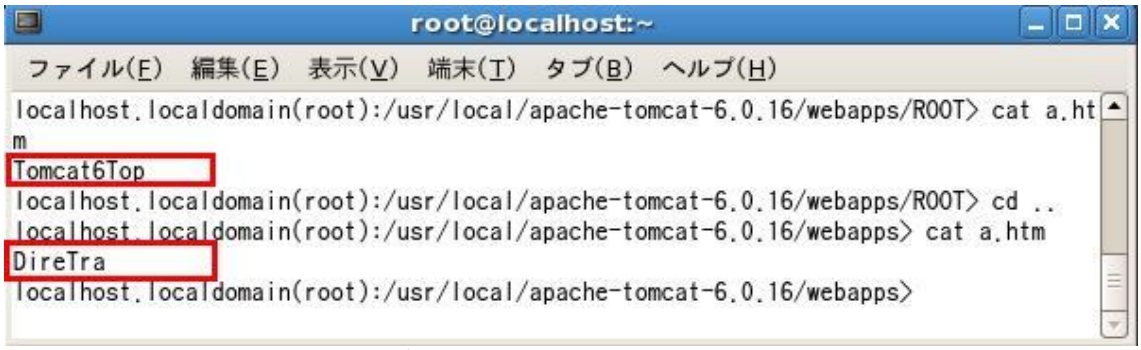

図 6.2-1 : ウェブルートの「a.htm」の中身は「Tomcat6Top」。

その一つ上の非公開ディレクトリ上の「a.htm」の中身は「DireTra」となっている。

CVE-2008-2938 を使って、非公開ディレクトリ上の

「a.htm」が読み出せるかどうかがこの試験内容である

図 6.2-2 : ウェブルート「/a.htm」のアクセス結果。

検査対象の Web サーバ(Tomcat)は正常に動作している

図 6.2-3 : 2Byte の冗長な UTF-8 表現に対しての結果。

CVE-2008-2938 の指摘どおり再現している

図 6.2-4 : 3Byte の冗長な UTF-8 表現でも、問題の再現ができている事が確認できる

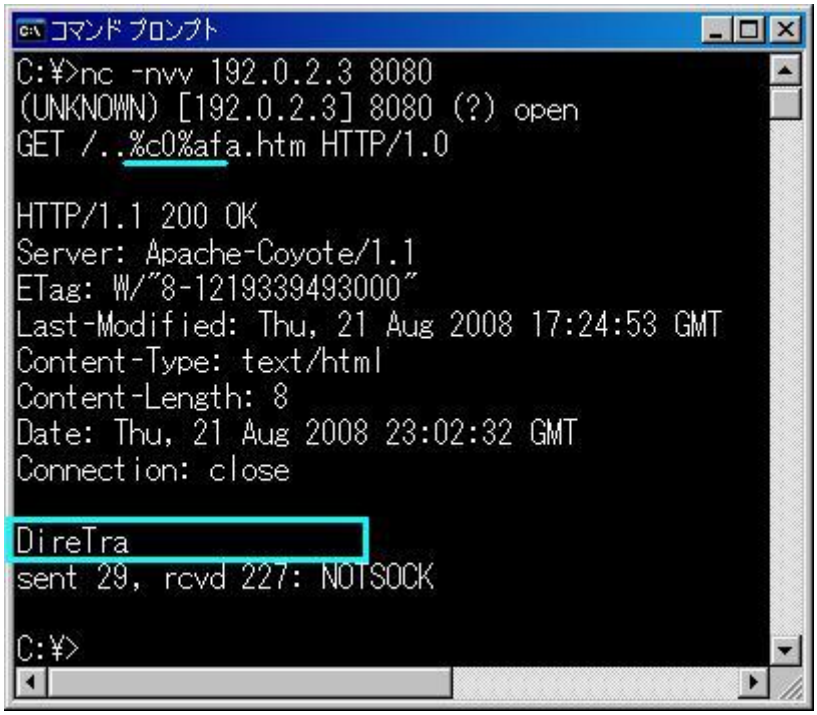

図 6.2-5：「.(ピリオド)」ではなく、「/(スラッシュ)」に対しての冗長な UTF-8 表現でも、問題の再現ができている事が確認できる

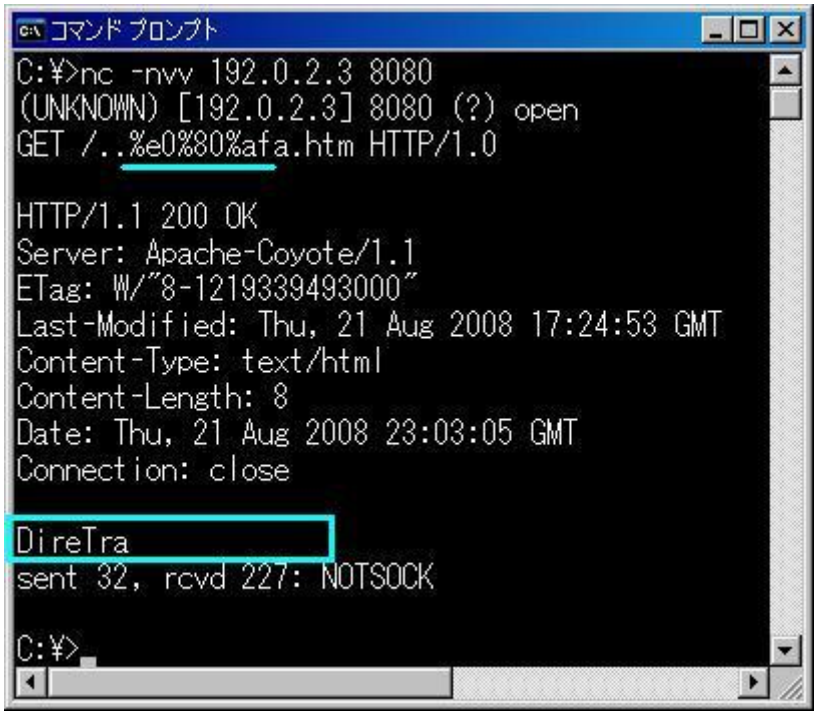

図 6.2-6：「/(スラッシュ)」の 3Byte の冗長な UTF-8 表現でも、問題の再現ができている事が確認できる

7. CVE-2008-2938 と OS Command Injection に関する実験結果

以下の環境で、実験を行った。

- CentOS 5.1
- JDK 1.5.0_16
- Apache-Tomcat 5.5.26

既定は、Perl スクリプトであるが、OS バイナリの実行プログラムの CGI を許可している設定にしている場合、Apache-Tomcat の動作権限で許可された任意のプログラムが実行させられる危険性がある。

7.1. Apache-Tomcat 5.5.26 の場合

図 7.1-1 : スクリプトを用意した。Web ルートの「test-pl.cgi」の内容。

Perl スクリプトである

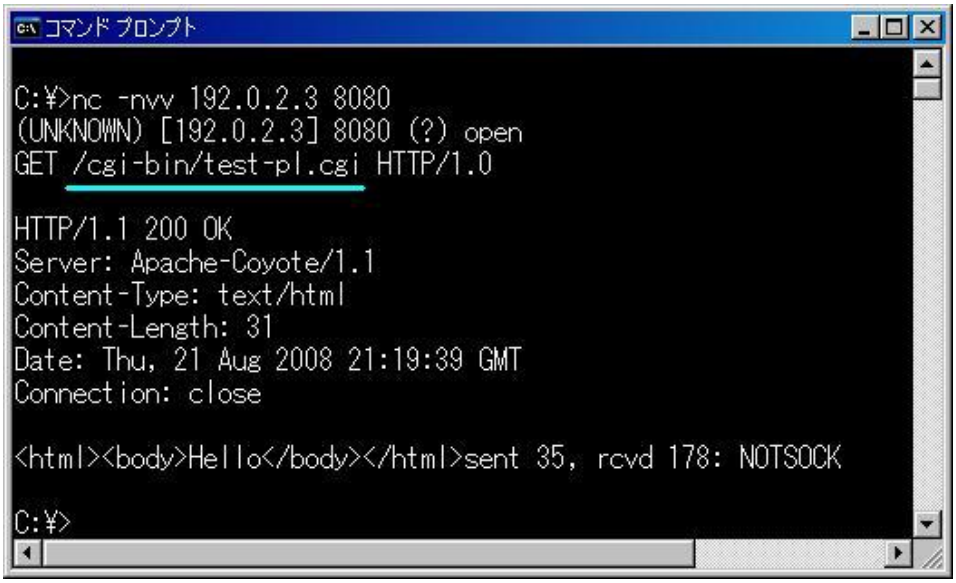

図 7.1-2 : 「/cgi-bin/test-pl.cgi」の結果。

Tomcat の CGI (Perl)が正常に動作していることが分かる

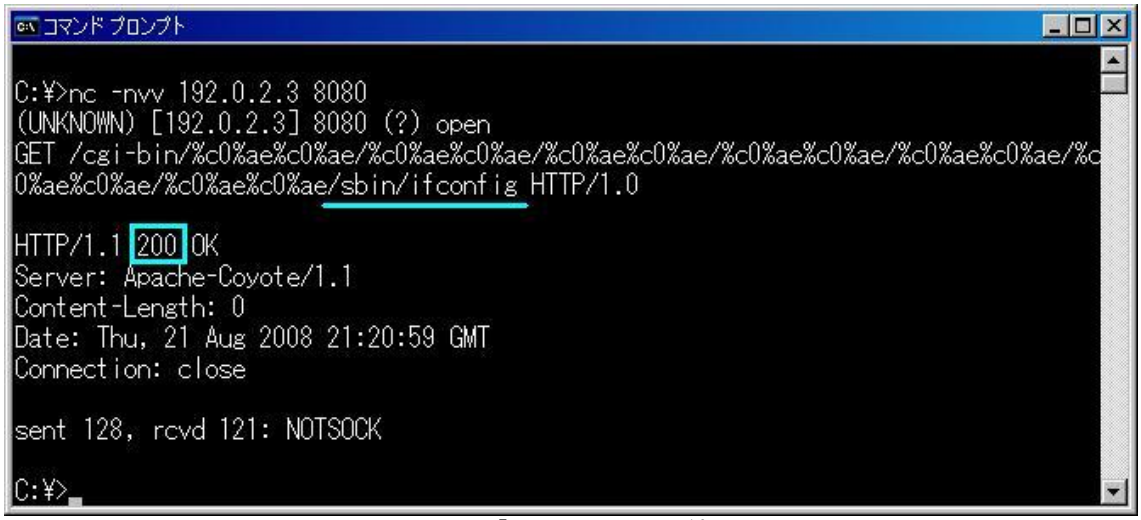

図 7.1-3 : 「/sbin/ifconfig」の結果。

Perl スクリプト用の Tomcat CGI の設定ではバイナリの起動はできない。

ファイルの存在有無が分かる程度である

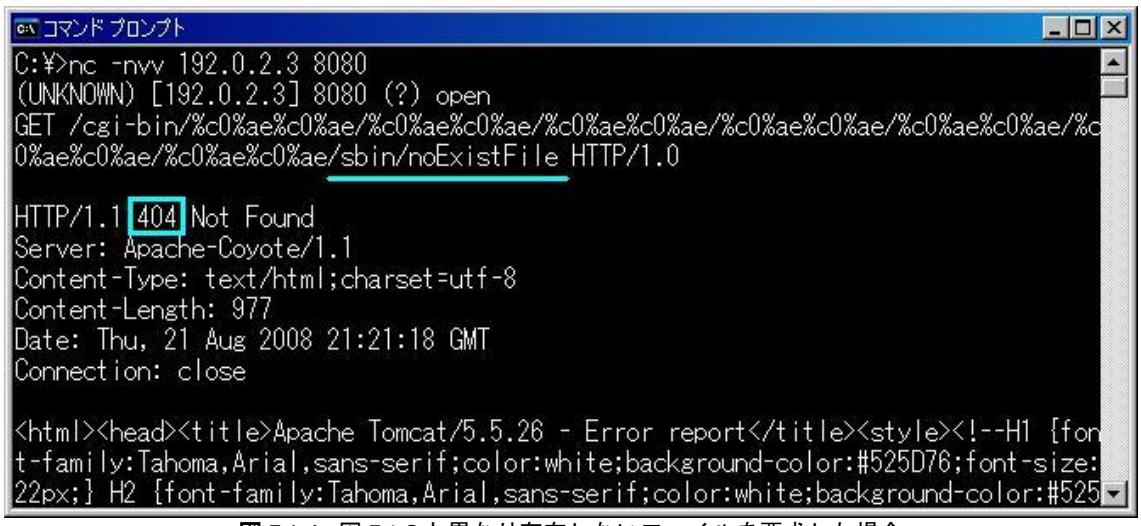

図 7.1-4 : 図 7.1-3 と異なり存在しないファイルを要求した場合

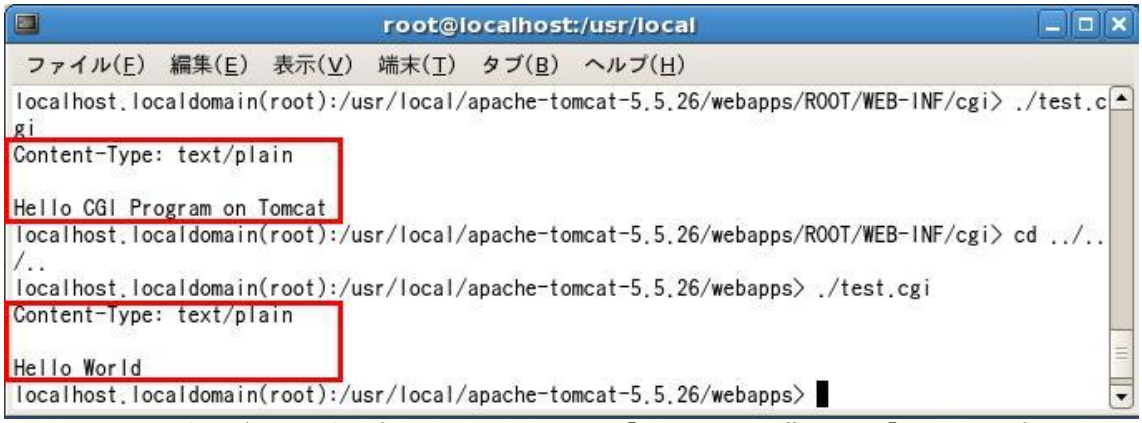

図 7.1-5 : 次にバイナリを用意した。Web ルートの「test.cgi」と三階層上の「test.cgi」である

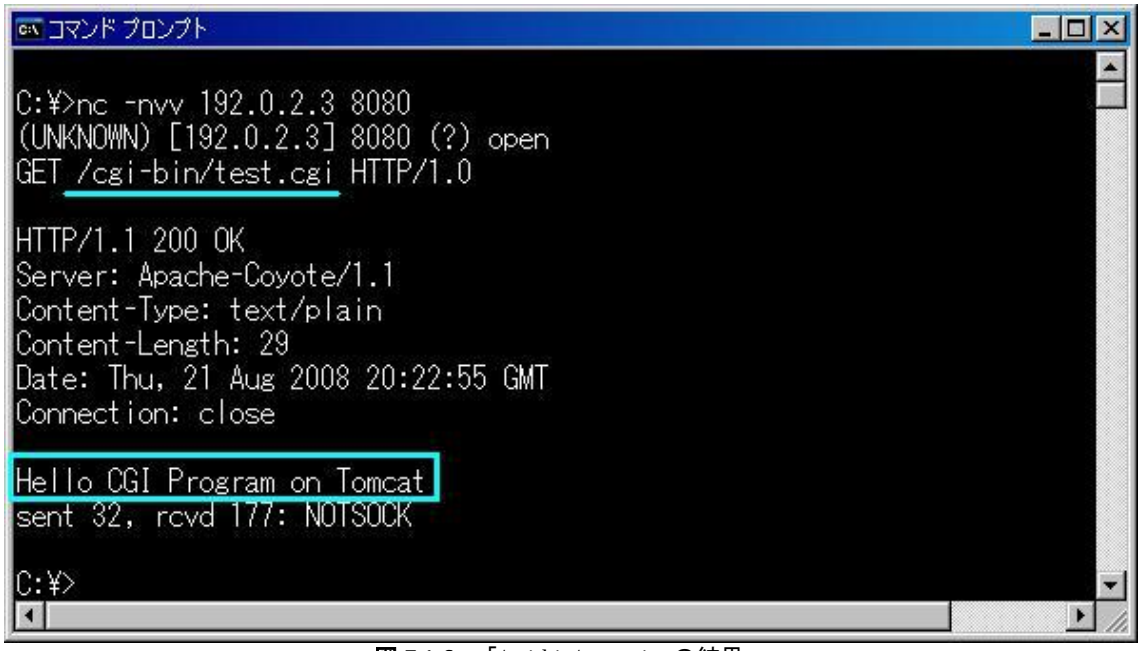

図 7.1-6：「/cgi-bin/test.cgi」の結果。

Tomcat が CGI(バイナリ)を正しく処理していることが確認できる

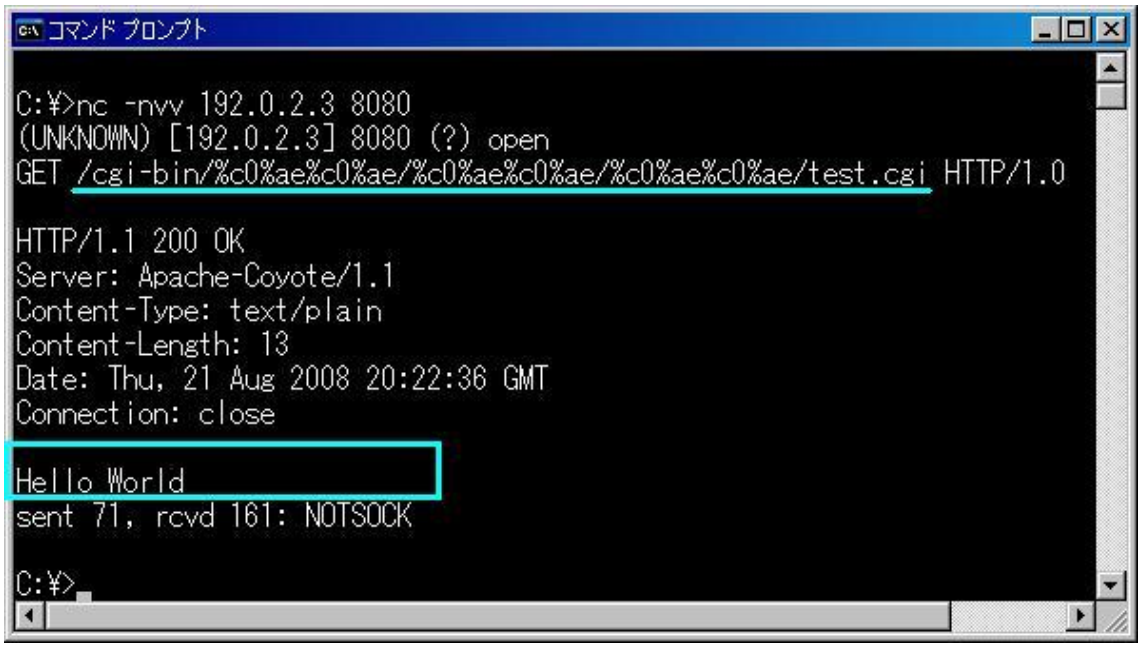

図 7.1-7：「/cgi-bin/%C0%AE%C0%AE/%C0%AE%C0%AE/%C0%AE%C0%AE/test.cgi」の結果。

三階層上位の「test.cgi」が呼び出されていることから、任意の OS Command を実行できると確認された

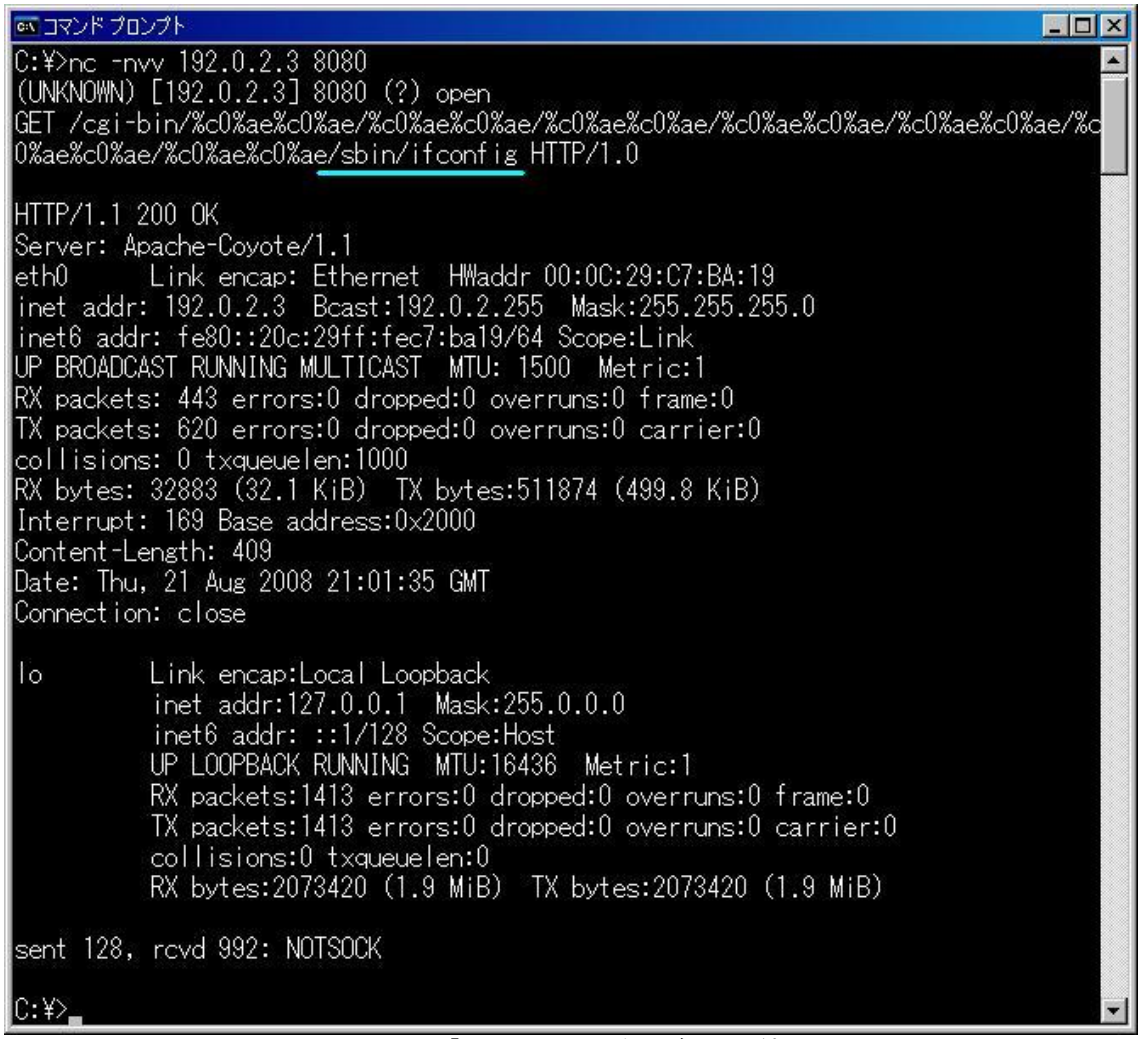

図 7.1-8：「/sbin/ifconfig」を呼び出した結果

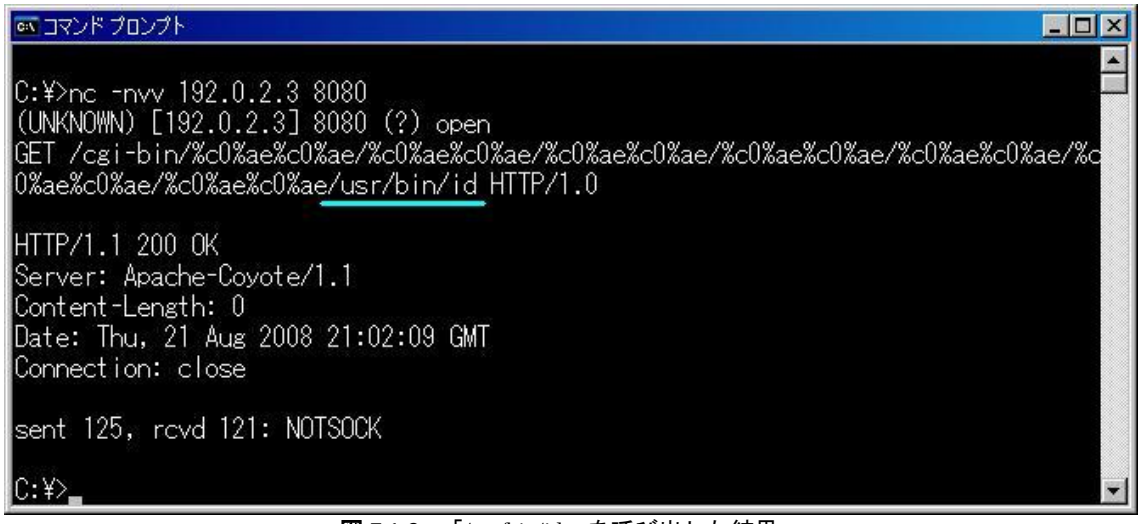

図 7.1-9：「/usr/bin/id」を呼び出した結果。

どうも、空行を返さないコマンドは、出力されないようだ。

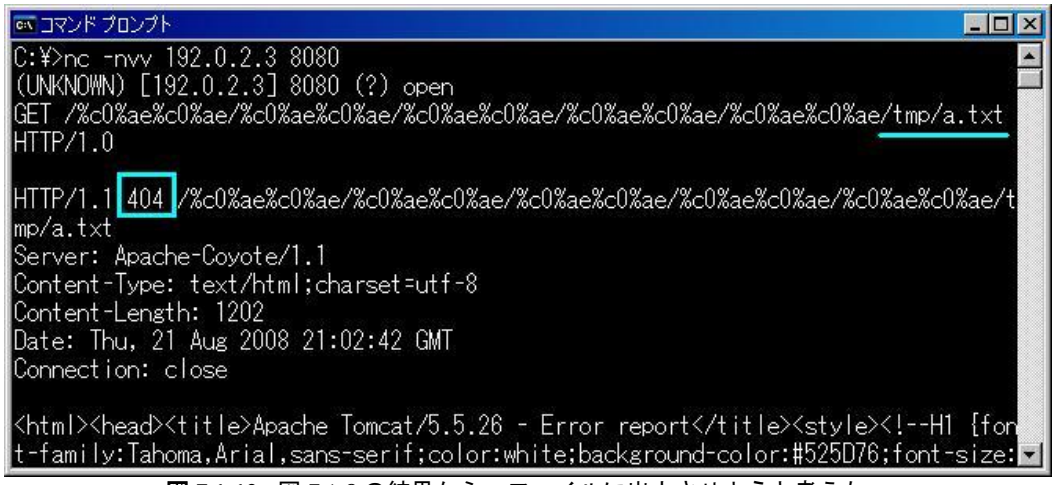

図 7.1-10：図 7.1-9 の結果から、ファイルに出力させようと考えた。

まずは、「/tmp/a.txt」がないことを確認した

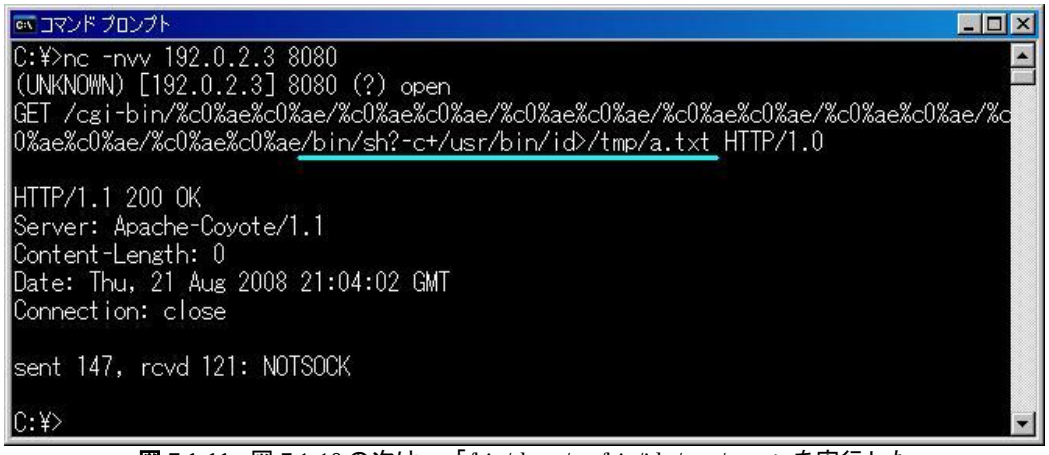

図 7.1-11：図 7.1-10 の次は、「/bin/sh -c /usr/bin/id>/tmp/a.txt」を実行した

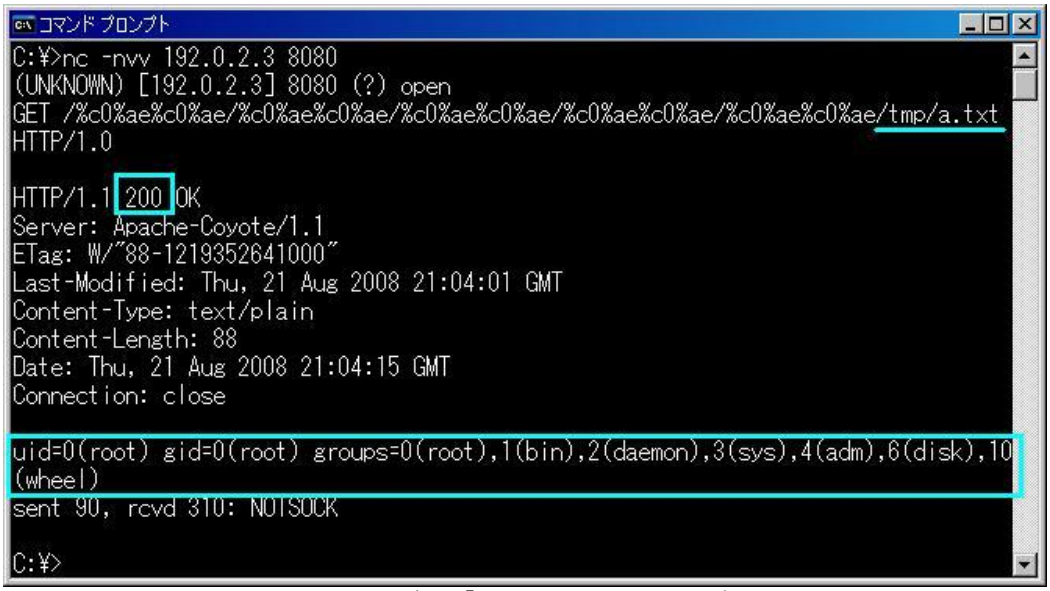

図 7.1-12：図 7.1-11 後に「/tmp/a.txt」にアクセスすることで、

一行だけ返すコマンドの実行結果も取得することができる

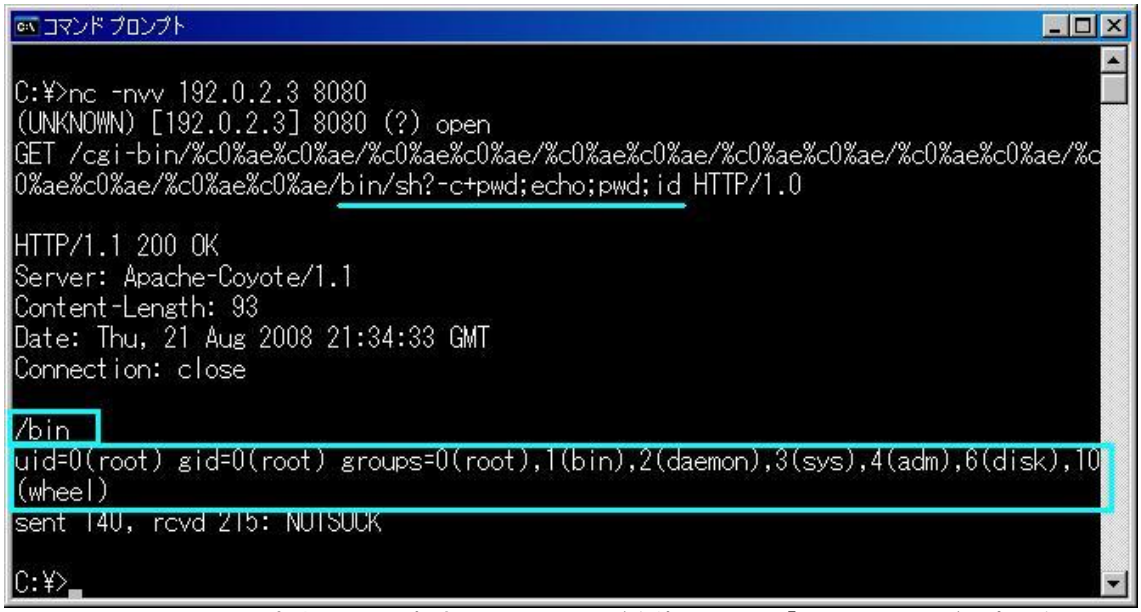

図 7.1-13 : 上記の方法とは別の視点で、sh コマンドを使い、かつ「echo」コマンドで空行を出力させることで、「pwd」と「id」の二つのコマンドの実行結果を表示させた

8. Windows 上の Tomcat での「\(バックスラッシュ)」の冗長な UTF-8 表現

「\(バックスラッシュ)」は「0x5c」である。「\(バックスラッシュ)」の 2Byte の冗長な UTF-8 表現は「0xC1」「0x9C」となる。
3Byte の冗長な UTF-8 表現は、「0xE0」「0x81」「0x9C」である。
MS-Windows 版の Apache-Tomcat で再現させてみた結果、CVE-2008-2938 が再現した。

以下の環境で、実験を行った。

- Microsoft-WindowsXP SP3
- JDK 1.6.0_07
- Apache-Tomcat 5.5.26

8.1. Apache-Tomcat 5.5.26 の場合

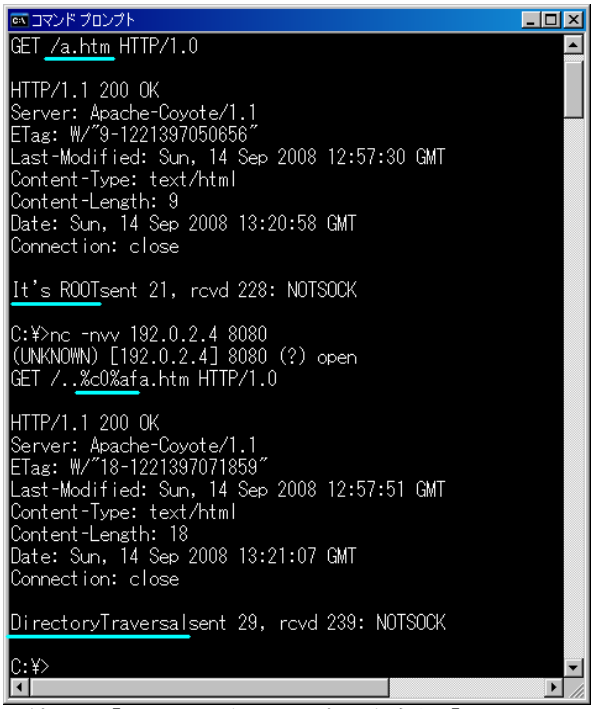

図 8.1-1：「/a.htm」の結果と「/」の冗長な UTF8 表現を含む「/..%C0%AFa.htm」の違いを確認

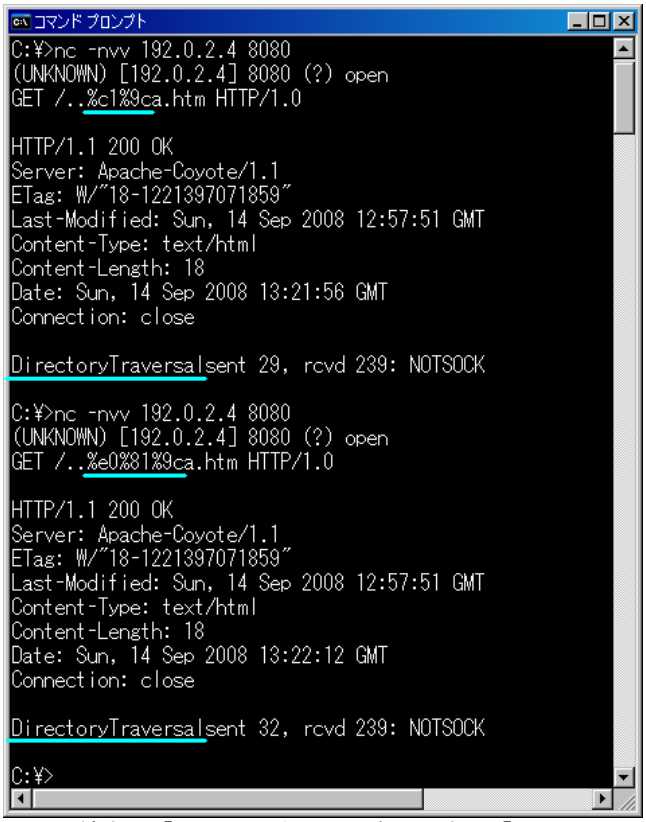

図 8.1-2：図 8.1-1 の続き。「\」の冗長な UTF8 表現付きの「/..%C1%9Ca.htm」でも、
3Byte の冗長な UTF8 表現付き「/..%E0%81%9Ca.htm」でも発現している。

9. Windows 上の Tomcat での「¥(円記号(U+00A5))」と冗長な UTF-8 表現

UNICODE では、「¥(円記号)」を「バックスラッシュ」とは異なる「U+00A5」に割り当てられている。
よって、「%A5」、UTF-8 表現である「%C2%A5」、および 3Byte の冗長な UTF8 表現
「%E0%82%A5」について、MS-Windows 版の Apache-Tomcat で再現させてみた。
その結果、CVE-2008-2938 は再現しなかった。

以下の環境で、実験を行った。

- Microsoft-WindowsXP SP3
- JDK 1.6.0_07
- Apache-Tomcat 5.5.26

9.1. Apache-Tomcat 5.5.26 の場合

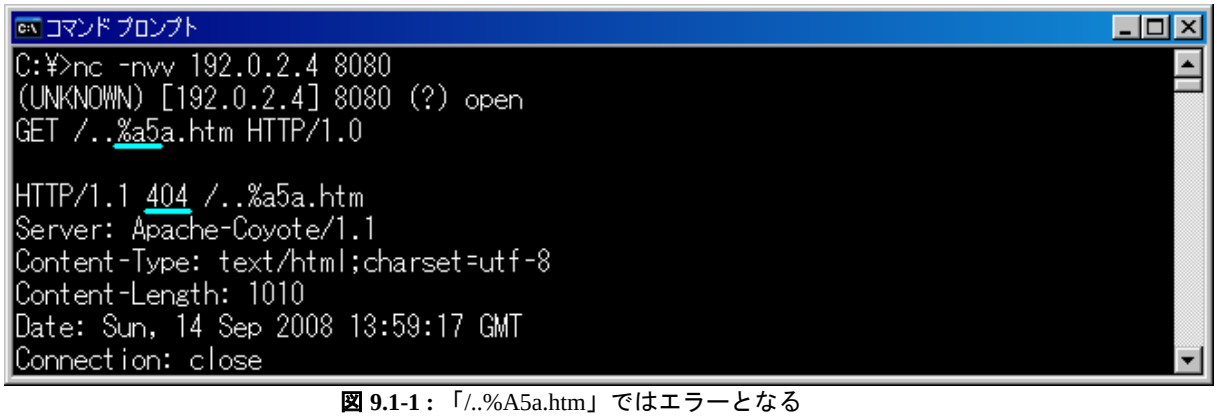

図 9.1-1：「/..%A5a.htm」ではエラーとなる

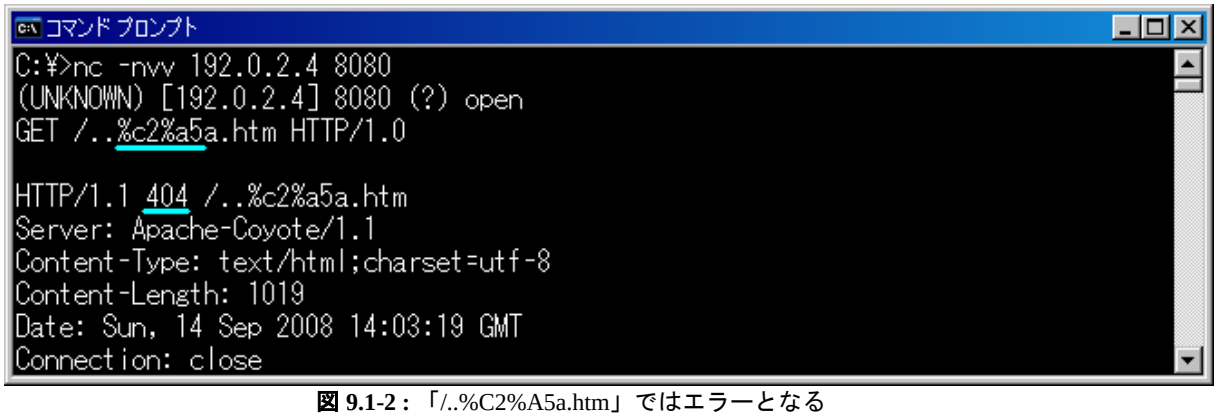

図 9.1-2：「/..%C2%A5a.htm」ではエラーとなる

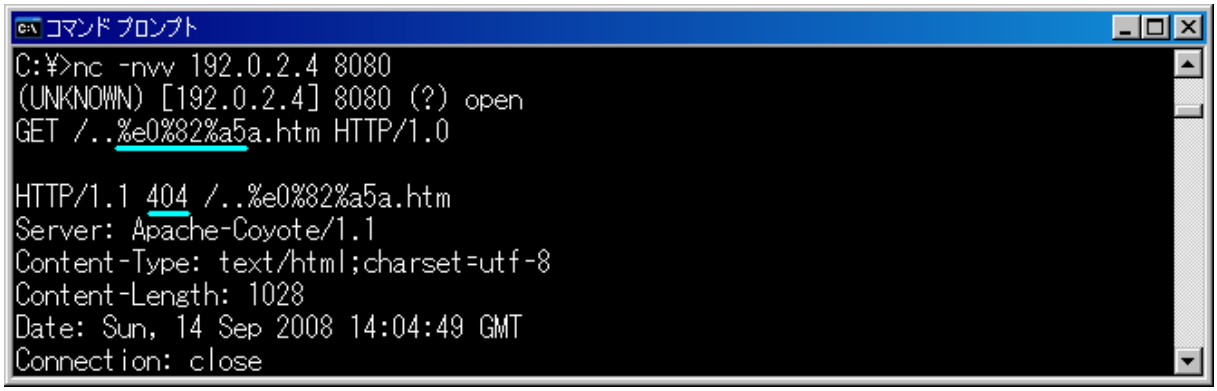

図 9.1-3：「/..%E0%82%A5a.htm」ではエラーとなる

10. 検証作業者

NTT コミュニケーションズ株式会社
IT マネジメントサービス事業部ネットワークマネジメントサービス部
セキュリティオペレーションセンター
　佐名木 智貴

11. 参考

- CVE-2008-2938
  http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-2938
- milwOrm.com
  http://www.milw0rm.com/exploits/6229
- Tomcat のディレクトリトラバーサルの脆弱性に関する検証レポート
  http://www.nttdata-sec.co.jp/column/report20080813.pdf
- Tomcat にディレクトリトラバーサル脆弱性、NTT データ・セキュリティが注意喚起
  http://www.atmarkit.co.jp/news/200808/13/tomcat.html
- 「正規化エラーによる、ファイルへの誤ったアクセス権の適用」の脆弱性に対する対策 (MS00-057)
  http://www.microsoft.com/japan/technet/security/bulletin/ms00-057.mspx
- Microsoft IIS and PWS Extended Unicode Directory Traversal Vulnerability
  http://www.securityfocus.com/bid/1806

12. 履歴

- 2008 年 08 月 26 日 : ver0.1 仮公開
- 2008 年 08 月 27 日 : ver0.2 仮公開 (誤字脱字を修正)
- 2008 年 09 月 15 日 : ver0.3 「\」の UTF8 冗長表現が「%C1%1C」という間違いを「%C1%9C」に修正した。円記号(U+00A5)についての実験を追加した。
- 2008 年 09 月 25 日 : ver1.0 本公開 (公開位置(URL)の移動と、誤字脱字を修正)
- 2009 年 05 月 26 日 : ver1.1 Web サイト移転に伴う最新版公開 URL の変更

13. 最新版の公開 URL

http://www.ntt.com/icto/security/data/soc.html#security_report

14. 本レポートに関する問合せ先

NTT コミュニケーションズ株式会社
IT マネジメントサービス事業部ネットワークマネジメントサービス部
セキュリティオペレーションセンター

e-mail: scan@ntt.com

以 上