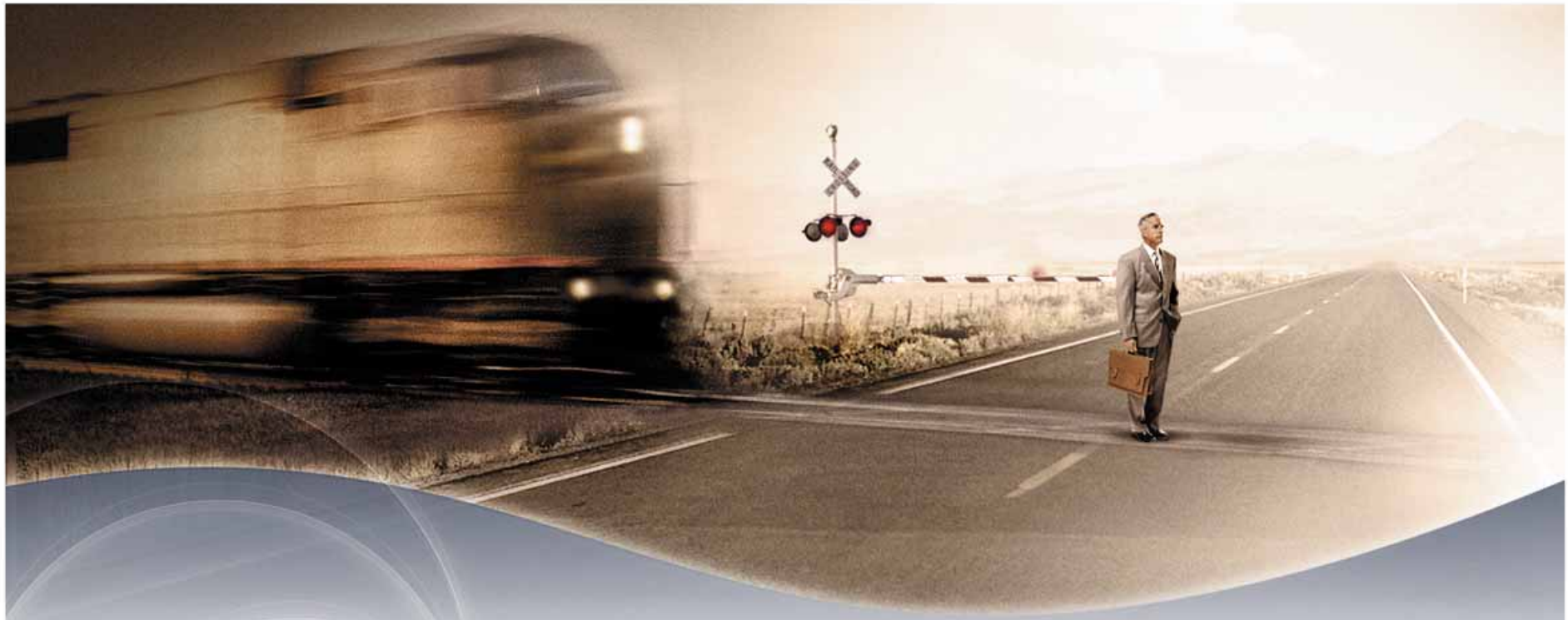


Windowsサーバシステムのログ管理

インターネット セキュリティ システムズ株式会社
テクニカル・ソリューション課 マネージャ
エグゼクティブ セキュリティ エンジニア
小倉 秀敏

INTERNET | SECURITY | SYSTEMS®

対象アプリケーション

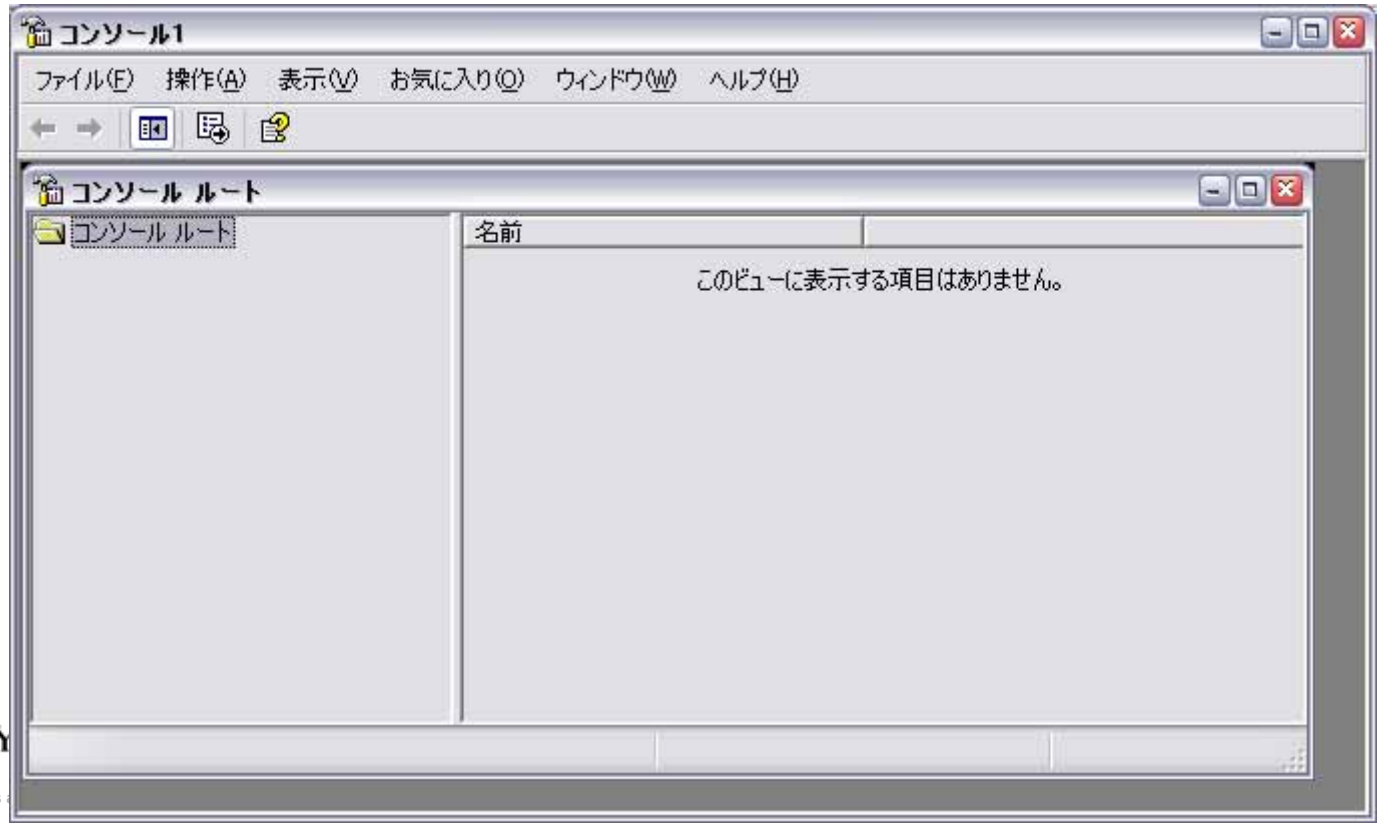
- Internet Information Server
- SQL Server

ログ管理の目的

- 目的
  - 定常状態を知る
  - セキュリティ問題が発生している兆候を知る
- 不正侵入･不正利用の防止は不可能
  - 事後の発見のみ
  - 防御には防御ソリューションが必要
  - サーバに対する「悪意を持った全ての操作」が記録できるわけではない
- フォレンジックスを目的にしない方がよい
  - フォレンジックスには専門のトレーニングが必要
    - Explorerでログファイルを参照するだけで証拠能力は消滅

ログ管理を容易にするために・・・
カスタムMMCの構成

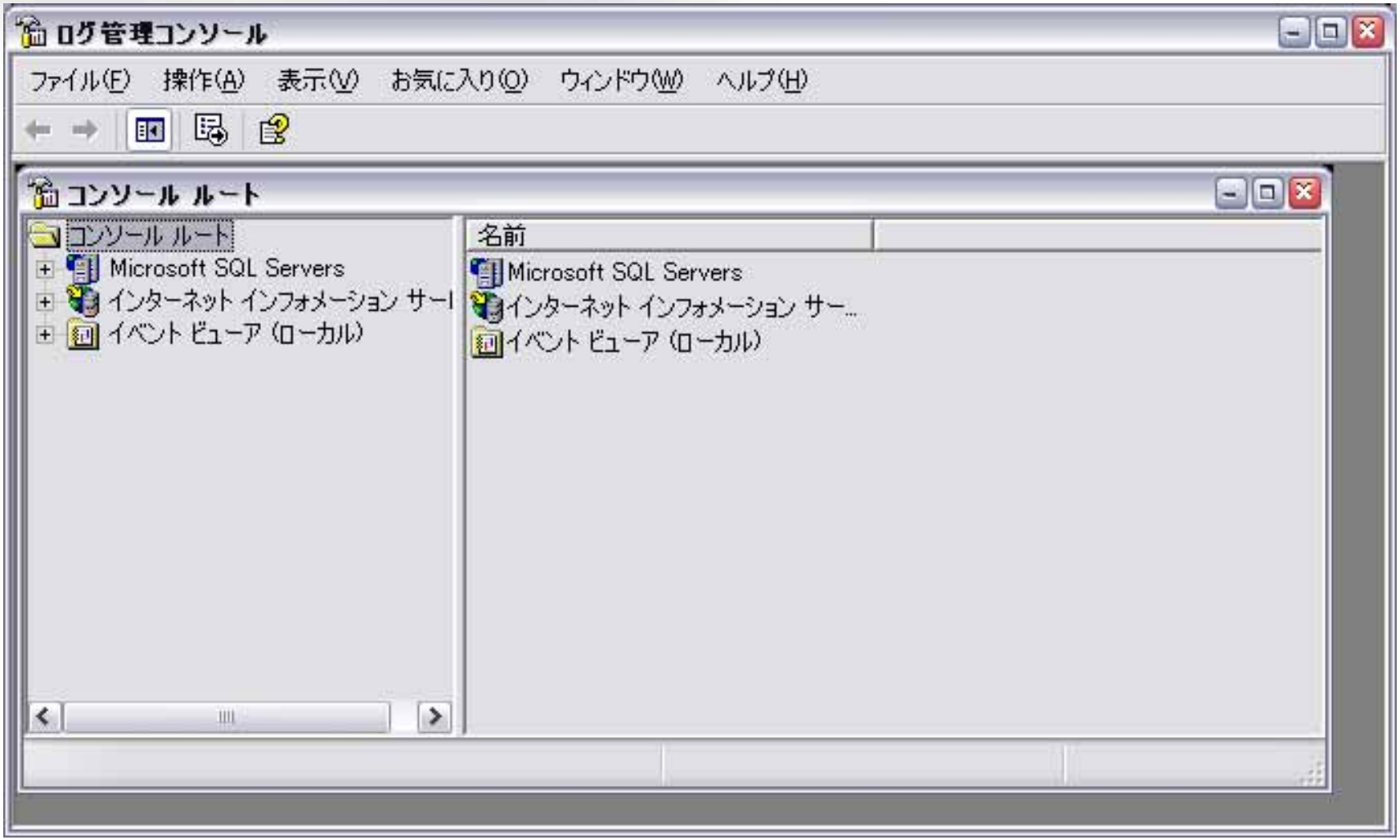
- **専用のMMCコンソールファイルの作成**
  - mmc /aコマンドによるカスタムコンソールファイルの作成
  - SQL Enterprise Manager、IISコンソール、イベントビューア等をひとまとめに

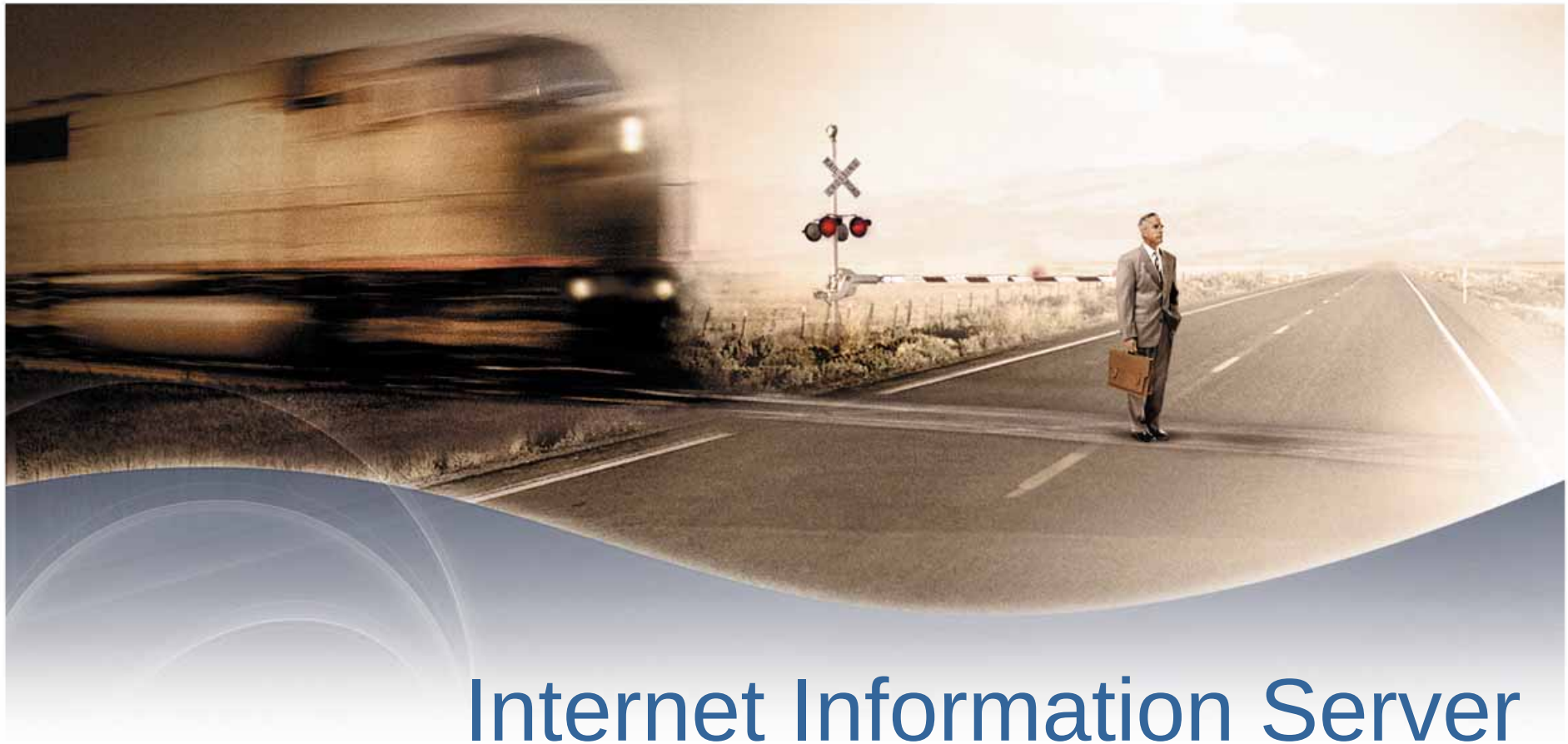


INTERNET SECURITY

MMCへのスナップインの追加

カスタマイズが終了したMMC





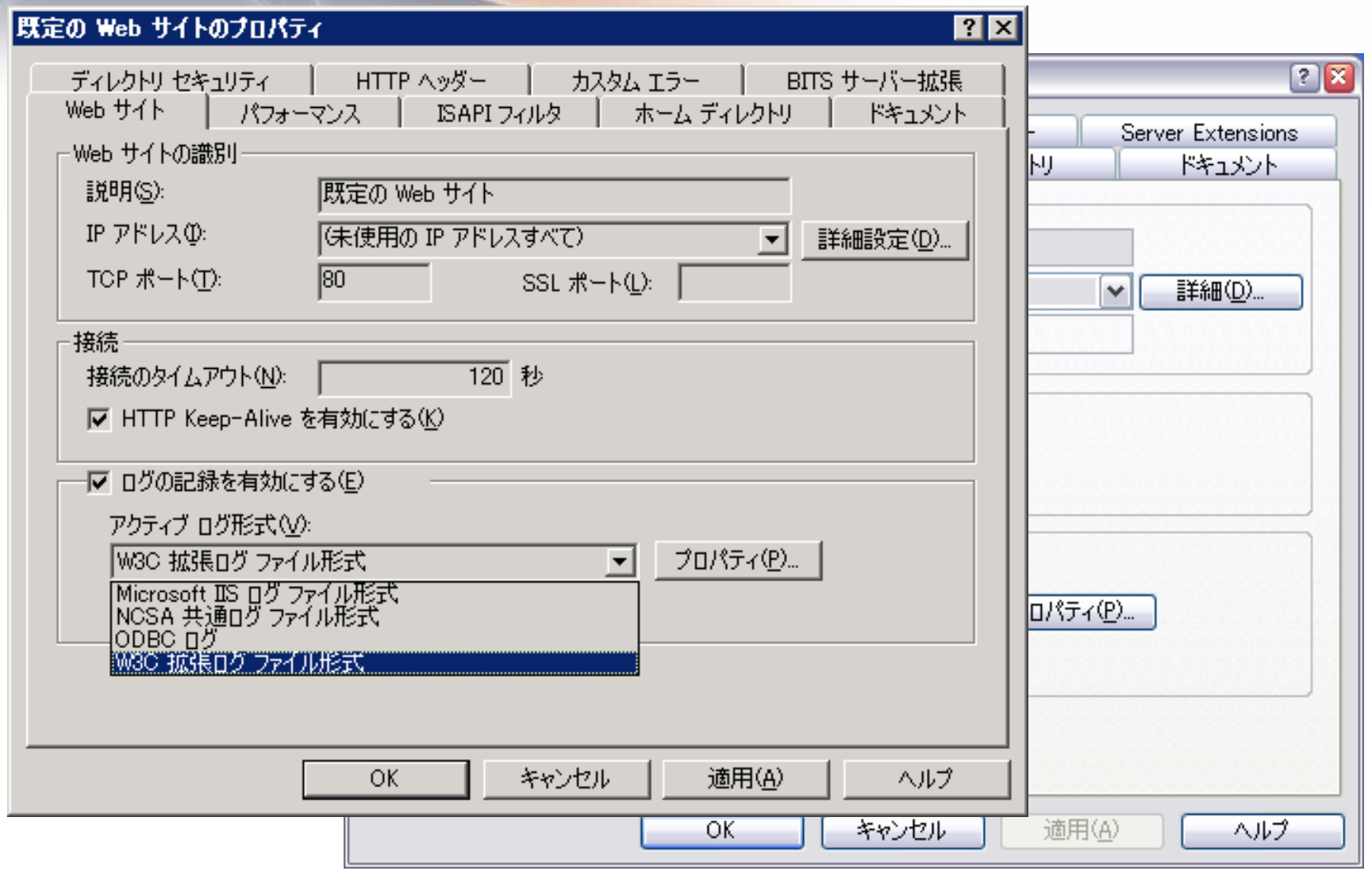
Internet Information Server

INTERNET | SECURITY | SYSTEMS®

IISのログ解析

- **対象ログ**
  - %SystemRoot%¥system32¥Logfilesディレクトリに保存されるログ
- **解析する内容とは**
  - IISに対するアクセスログ
  - 制限的であることを理解する必要あり
    - バッファオーバーフローにより動作させられたプログラム等をログすることはできない
    - Webアプリケーションの構成によりログに保存できるレベルが影響を受ける

IISログ形式の設定



ODBCログ

- **データベースへのログ保管**
  - Windows 2003 ServerのIIS 6.0から利用可能
  - ログ保管先にデータベースを利用
  - データベースに保管された他システムとのログの突き合わせには有効
    - IPアドレスなどを利用した複数ログの結合クエリ
  - IIS自体のパフォーマンスに影響を与えるため積極的には使用されない

    ClientHost, Username, LogTime, Service, Machine, ServerIP, ProcessingTime, BytesRecvd, BytesSent, ServiceStatus, Win32Status, Operation, Target, Parameters

IISで記録可能な項目

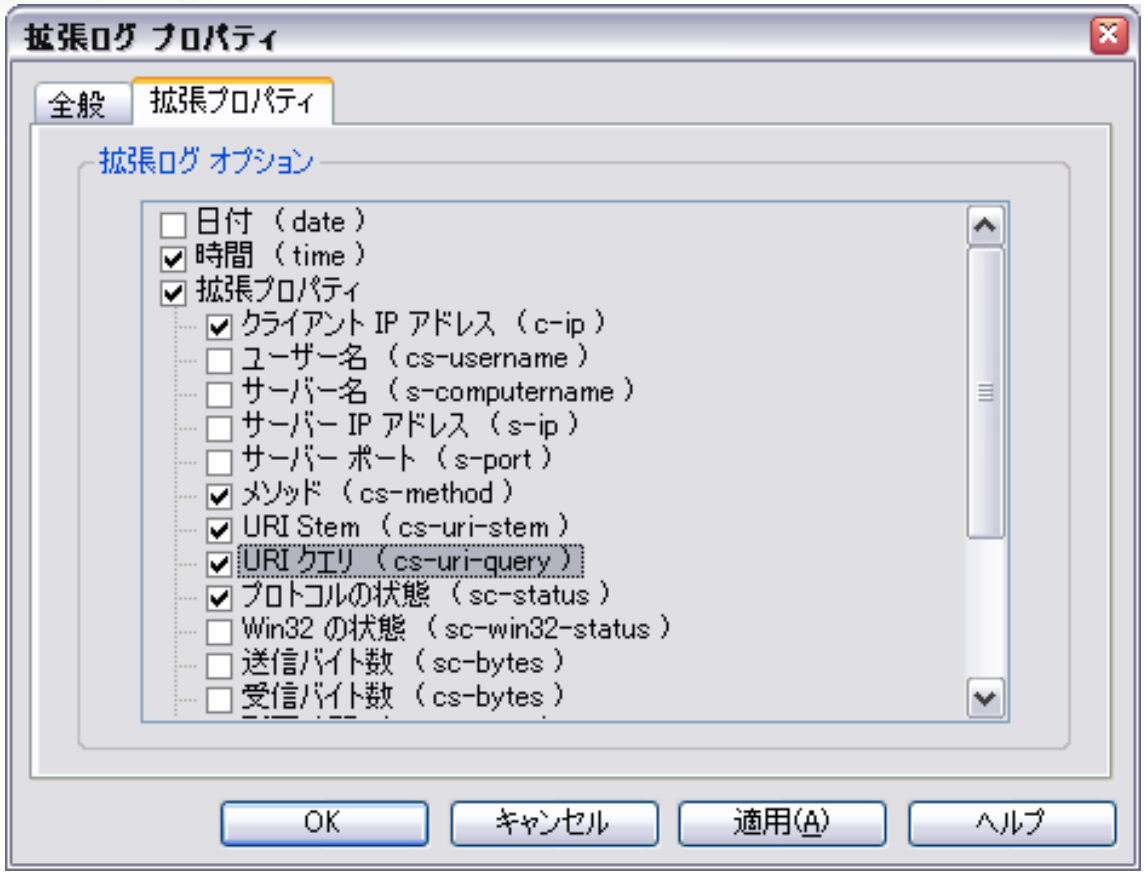
拡張設定項目	説明
クライアント IP アドレス	クライアントのIPアドレス
ユーザ名	サーバにアクセスしたユーザの名前
サービス名	クライアントで実行されているインターネット サービス
サーバー名	サーバの名前
サーバー IP	サーバのIPアドレス
サーバー ポート	クライアントが接続したポート番号
メソッド	クライアントが実行を試みたアクション
URI Stem	HTML ページ、CGI プログラム、スクリプトなどの、アクセスされたリソース
URI クエリ	クライアントが実行しようとしたクエリ文字列
プロトコルの状態	HTTP 関連の処理の状態
Win32の状態	Windows 関連の処理の状態
送信バイト数	サーバが送信したバイト数
受信バイト数	サーバが受信したバイト数
所要時間	処理にかかった時間
プロトコルバージョン	クライアントが使用したプロトコルバージョン
ホスト	コンピュータ名
ユーザエージェント	クライアントが使用したブラウザ
Cookie	送受信された Cookie の内容
参照者	ユーザを現在のサイトに導いたサイト
プロトコルの副状態	HTTP 関連の処理のその他の状態

IISログ設定

- URIクエリを追加しない場合
  - #Fields: time c-ip cs-method cs-uri-stem sc-status
  - 00:57:12 xx.xx.xx.xx POST /iishelp/iis/misc/Query.asp 200
- URIクエリを追加した場合
  - #Fields: time c-ip cs-method cs-uri-stem cs-uri-query sc-status
  - 00:58:45 xx.xx.xx.xx POST /iishelp/iis/misc/Query.asp SearchType=0 200
- ユーザーエージェントはあまり意味がない
  - 01:14:55 xx.xx.xx.xx POST /iishelp/iis/misc/Query.asp SearchType=3 200 Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.1;+.NET+CLR+1.1.4322)

IISログ設定

- **URIクエリをログに残す**
  - URIを残すことで、クエリとして渡された文字列を確認することが可能



どんなログを残すか

- **利用状況を把握した上での設定**
  - **利用しているメソッドに応じた設定**
  - GET**メソッドのみであれば**URI、POST**メソッドを使っているのであれば** Cookie**等その他の情報**

#Fields: time c-ip cs-method cs-uri-stem cs-uri-query sc-status cs(Cookie) cs(Referer)

02:35:53 xx.xx.xx.xx GET /iishelp/iis/misc/search.asp Searchset=3&SearchString=**オブジェクト** 200 ASPSESSIONIDGGQGGVFC=IFFFPKHBBLNMNEIFJJBPHOMB http://localhost/iishelp/iis/misc/default.asp

02:35:55 xx.xx.xx.xx POST /iishelp/iis/misc/Query.asp SearchType=3 200 ASPSESSIONIDGGQGGVFC=IFFFPKHBBLNMNEIFJJBPHOMB http://localhost/iishelp/iis/misc/search.asp?Searchset=3&SearchString=**オブジェクト**

ログの設定法

- **取得する情報の選択**
  - Webアプリケーションで利用しているものを確認して項目を設定
  - GETメソッドが利用される場合URIが重要
  - POSTメソッドを使用している場合は渡される内容を直接取得は不可能
  - 受信バイト数、送信バイト数
    - XSS攻撃の際、リクエスト、レスポンスが大きくなる可能性があるため送信バイト数や受信バイト数が利用できる可能性あり

攻撃とログの例

- ディレクトリ・トラバーサル
- IDS検知回避
- バッファ・オーバーフロー

ディレクトリ・トラバーサル

- **/cgi-binディレクトリ配下からディレクトリ・トラバーサルを利用**

2005-01-25 04:44:31 192.168.35.52 GET /cgi-bin/main.cgi board=FREE_BOARD&command=down_load&filename=../../../../ 80 - 192.168.35.217 - 404 0 3 1800 150

2005-01-25 04:44:31 192.168.35.52 GET /cgi-bin/main.cgi board=FREE_BOARD&command=down_load&filename=../../../../../../../../../../../etc/passwd 80 - 192.168.35.217 - 404 0 3 1800 178

2005-01-25 04:44:31 192.168.35.52 GET /cgi-bin/main_menu.pl - 80 - 192.168.35.217 - 404 0 3 1800 97

2005-01-25 04:44:31 192.168.35.52 GET /cgi-bin/majordomo.pl - 80 - 192.168.35.217 - 404 0 3 1800 97

2005-01-25 04:44:31 192.168.35.52 GET /cgi-bin/makechanges/easysteps/easysteps.pl - 80 - 192.168.35.217 - 404 0 3 1800 119

2005-01-25 04:44:31 192.168.35.52 GET /cgi-bin/man.sh - 80 - 192.168.35.217 - 404 0 3 1800 91

ディレクトリ・トラバーサル

- **InteInfoディレクトリからのトラバーサル**

2005-01-25 05:12:46 192.168.35.52 GET /..%2f..%2f..%2f..%2fwinnt/system32/cmd.exe /c+dir+c: 80 - 192.168.35.217 - 404 0 3 1800 137

2005-01-25 05:12:46 192.168.35.52 GET /..%5c..%5c..%5c..%5c..%5c../winnt/system32/cmd.exe /c+dir+c:¥ 80 - 192.168.35.217 - 404 0 3 1800 148

2005-01-25 05:12:46 192.168.35.52 GET /..%5c..%5c..%5c..%5cwin2000/system32/cmd.exe /c+dir 80 - 192.168.35.217 - 404 0 3 1800 136

2005-01-25 05:12:46 192.168.35.52 GET /..%5c..%5c..%5c..%5cwindows/system32/cmd.exe /c+dir 80 - 192.168.35.217 - 404 0 3 1800 136

2005-01-25 05:12:46 192.168.35.52 GET /..%5c..%5c..%5c..%5cwinnt/system32/cmd.exe /c+dir 80 - 192.168.35.217 - 404 0 3 1800 134

2005-01-25 05:12:46 192.168.35.52 GET /..%5c..%5cwinnt/system32/cmd.exe /c+dir+c: 80 - 192.168.35.217 - 404 0 3 1800 123

Nimdaが残すディレクトリ・トラバーサル

- **Nimdaワームが残す典型的なログ（URIのみ）**

GET /scripts/root.exe?/c+dir
GET /MSADC/root.exe?/c+dir
GET /c/winnt/system32/cmd.exe?/c+dir
GET /d/winnt/system32/cmd.exe?/c+dir
GET /scripts/..%5c../winnt/system32/cmd.exe?/c+dir
GET /_vti_bin/..%5c../..%5c../..%5c../winnt/system32/cmd.exe?/c+dir
GET /_mem_bin/..%5c../..%5c../..%5c../winnt/system32/cmd.exe?/c+dir
GET /msadc/..%5c../..%5c../..%5c/..¥xc1¥x1c../..¥xc1¥x1c../..¥xc1¥x1c../winnt/system32/cmd.exe?/c+dir
GET /scripts/..¥xc1¥x1c../winnt/system32/cmd.exe?/c+dir
GET /scripts/..¥xc0/../winnt/system32/cmd.exe?/c+dir
GET /scripts/..¥xc0¥xaf../winnt/system32/cmd.exe?/c+dir
GET /scripts/..¥xc1¥x9c../winnt/system32/cmd.exe?/c+dir
GET /scripts/..%35c../winnt/system32/cmd.exe?/c+dir
GET /scripts/..%35c../winnt/system32/cmd.exe?/c+dir
GET /scripts/..%5c../winnt/system32/cmd.exe?/c+dir
GET /scripts/..%2f../winnt/system32/cmd.exe?/c+dir

IDS等の検知回避系

- **IDSの検知回避もしくはディレクトリ・トラバーサルを目的**
  - Nimda等の攻撃にも含まれる
  - URIに含まれる文字列をエンコード
  - **現在この手法で回避されるIDSはない**

2005-01-25 04:44:58 192.168.35.52 GET /.%2e/.%2e/.%2e/winnt/boot.ini - 80 - 192.168.35.217 - 404 0 3 1800 112

2005-01-25 04:44:58 192.168.35.52 GET /.%2e/.%2e/.%2e/winnt/repair/sam._ - 80 - 192.168.35.217 - 404 0 3 1800 116

2005-01-25 04:44:58 192.168.35.52 GET /..%2f..%2f..%2f..%2f..%2f../windows/repair/sam - 80 - 192.168.35.217 - 404 0 3 1800 133

2005-01-25 04:44:58 192.168.35.52 GET /..%2f..%2f..%2f..%2f..%2f../winnt/repair/sam - 80 - 192.168.35.217 - 404 0 3 1800 131

2005-01-25 04:44:58 192.168.35.52 GET /..%2f..%2f..%2f..%2f..%2f../winnt/repair/sam._ - 80 - 192.168.35.217 - 404 0 3 1800 133

バッファオーバーフロー系

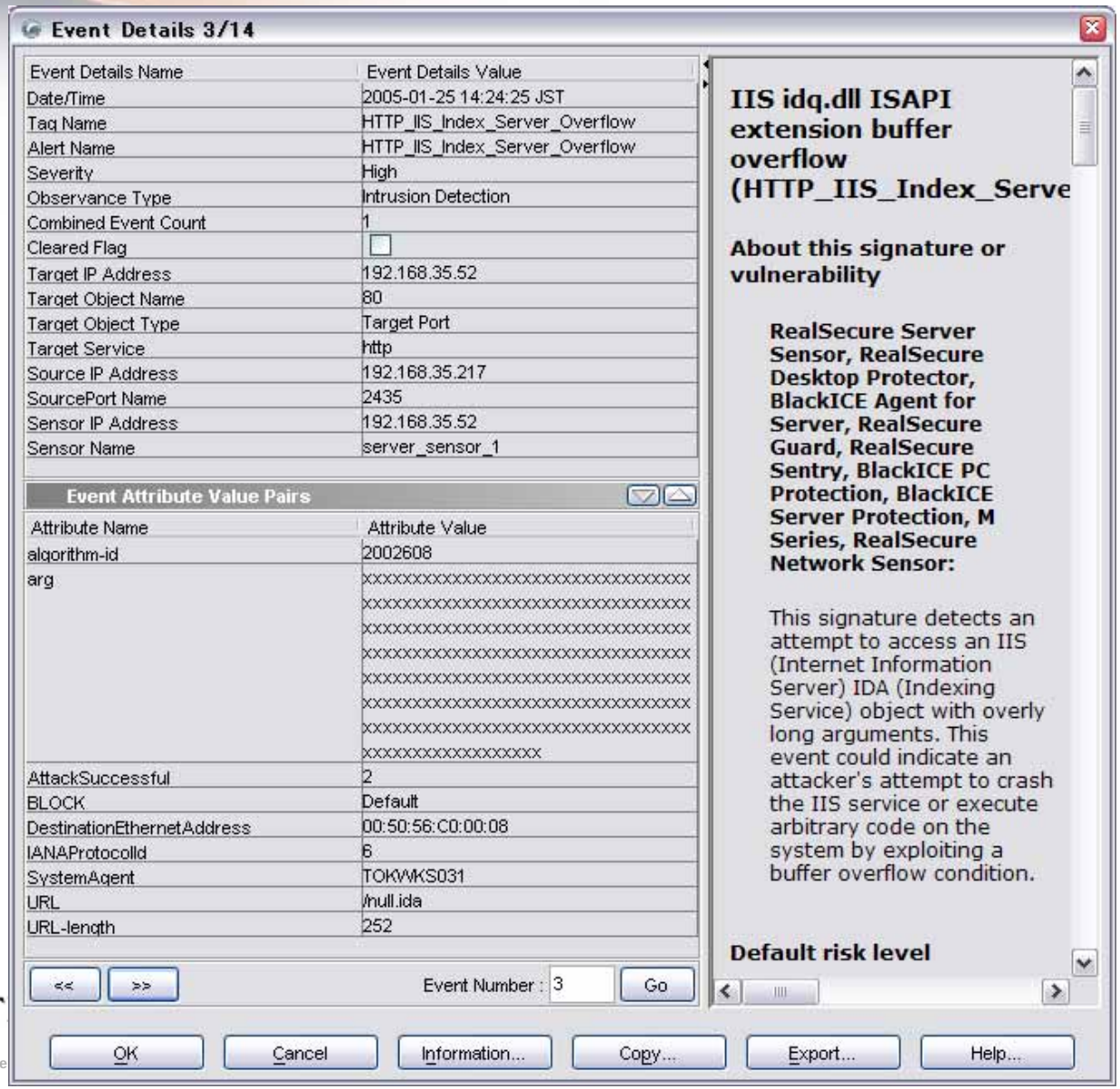
- **IIS idq.dll ISAPI extension buffer overflow**

2005-01-25 05:24:22 192.168.35.52 GET /null.ida
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
=x 80 - 192.168.35.217 - 404 0 2 1800 286

2005-01-25 05:24:22 192.168.35.52 GET /null.ida
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
x=x 80 - 192.168.35.217 - 404 0 2 1800 287

2005-01-25 05:24:22 192.168.35.52 GET /null.ida
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX=x 80 -
192.168.35.217 - 404 0 2 1800 338

IPSによるidq.dllオーバーフロー検知



INTERNET SECURITY SYST

クロスサイト・スクリプティング

IISのログ

- **<script>・・・</script>がPOSTで渡されていることが分からない**

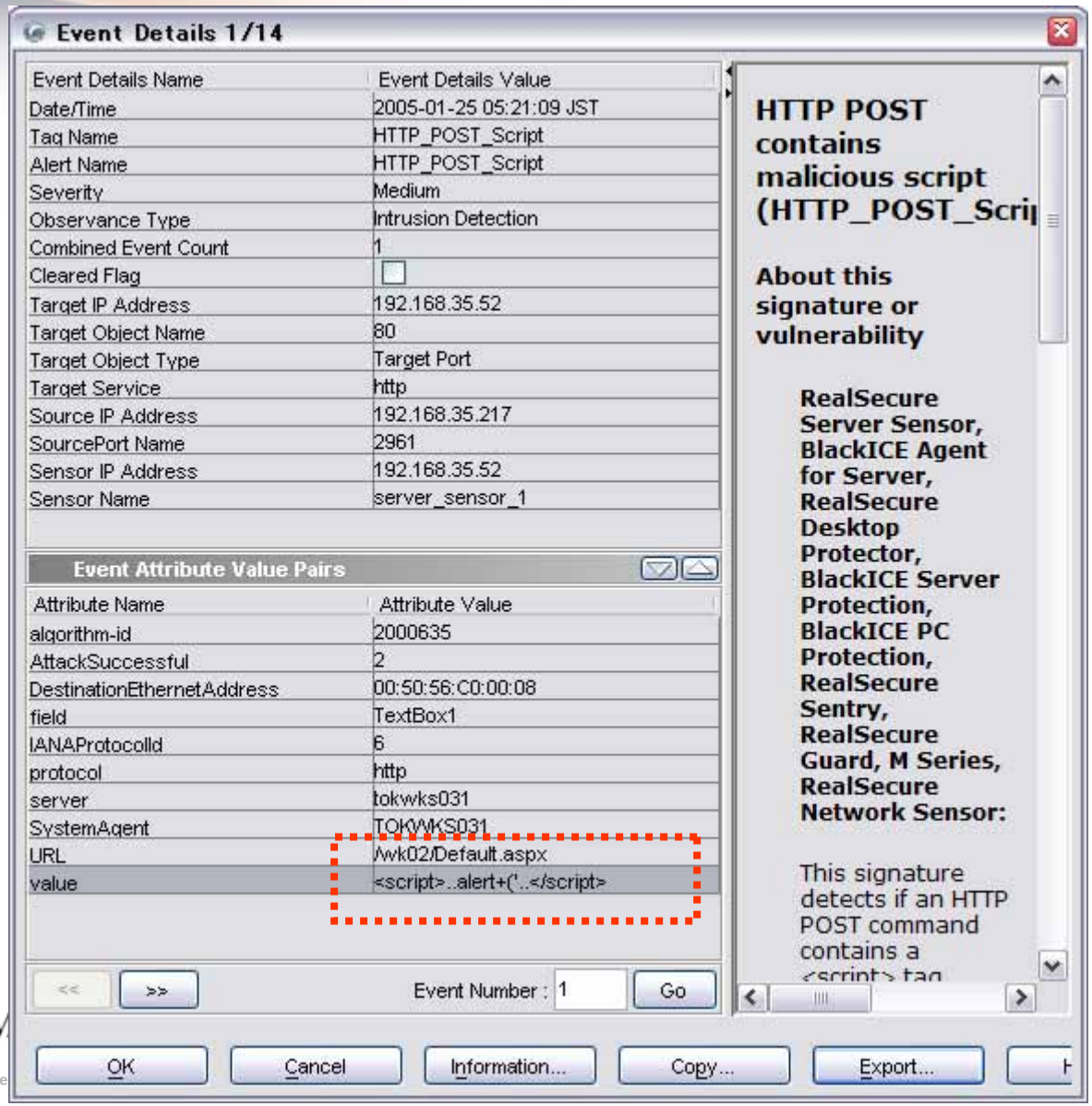
  #Software: Microsoft Internet Information Services 6.0
  #Version: 1.0
  #Date: 2005-01-24 20:20:43
  #Fields: date time s-ip cs-method cs-uri-stem cs-uri-query s-port cs-username c-ip cs(Cookie) sc-status sc-substatus sc-win32-status sc-bytes cs-bytes
  2005-01-24 20:20:43 192.168.35.52 GET /wk02/Default.aspx - 80 - 192.168.35.217 CookieLoginAttempts=4 200 0 0 994 405
  2005-01-24 20:21:08 192.168.35.52 POST /wk02/Default.aspx - 80 - 192.168.35.217 CookieLoginAttempts=4 200 0 0 1132 820

- **Webアプリケーションの構成により、攻撃自体がログに残らない**

- **注意：W3C形式でログを保存した場合、時刻はGMT**
  - Microsoftサポートオンライン：拡張ログファイル形式では時刻が常にGMTで記録される
  - http://support.microsoft.com/default.aspx?scid=kb;ja;194699

IPSであれば・・・

- POSTメソッドによるクロスサイト・スクリプティングを検知可能
  - IISのログでは確認できない内容を補うことが可能



INTERNET SECURITY SYSTEM

IPSでの検知時の情報

- Date/Time : 2005-01-25 05:21:09 JST
- Tag Name : HTTP_POST_Script
- Alert Name : HTTP_POST_Script
- Severity : Medium
- Tag Brief Description :
- Observance Type : Intrusion Detection
- Combined Event Count : 1
- Cleared Flag : No
- Target DNS Name :
- Target IP Address : 192.168.35.52
- Target Object Name : 80
- Target Object Type : Target Port
- Target Service : http
- Source DNS Name :
- Source IP Address : 192.168.35.217
- SourcePort Name : 2961
- Sensor DNS Name :
- Sensor IP Address : 192.168.35.52
- Sensor Name : server_sensor_1
- Attribute Value Pairs for Event Number : 1
- Attribute Name : algorithm-id
- Attribute Value : 2000635
- Attribute Name : AttackSuccessful
- Attribute Value : 2
- Attribute Name : DestinationEthernetAddress
- Attribute Value : 00:50:56:C0:00:08
- Attribute Name : field
- Attribute Value : TextBox1
- Attribute Name : IANAProtocolId
- Attribute Value : 6
- Attribute Name : protocol
- Attribute Value : http
- Attribute Name : server
- Attribute Value : tokwks031
- Attribute Name : SystemAgent
- Attribute Value : TOKWKS031
- Attribute Name : URL
- Attribute Value : /wk02/Default.aspx
- Attribute Name : value
- Attribute Value : <script>..alert+('..</script>

SQLインジェクション

- **サンプルWebサイトに対してSQLインジェクションによるログオン認証回避**
  - **ユーザ認証を以下のようなSQLで実現している場合**

    SELECT UserID FROM UserTbl

    WHERE UserName = txtUserName AND Passwrd = txtPassword

  - **ユーザ名として**’ OR 1=1 **を与えると・・・**

    SELECT UserID FROM UserTbl

    WHERE UserName = txtUserName AND Passwrd = txtPassword OR 1=1
  - **常に認証に成功**
    - **正規のユーザ名とパスワードを知っている必要がない**

SQLインジェクションの際のログ

- **不正なSQL文が渡されていることが分からない**

#Software: Microsoft Internet Information Services 6.0

#Version: 1.0

#Date: 2005-01-24 21:00:37

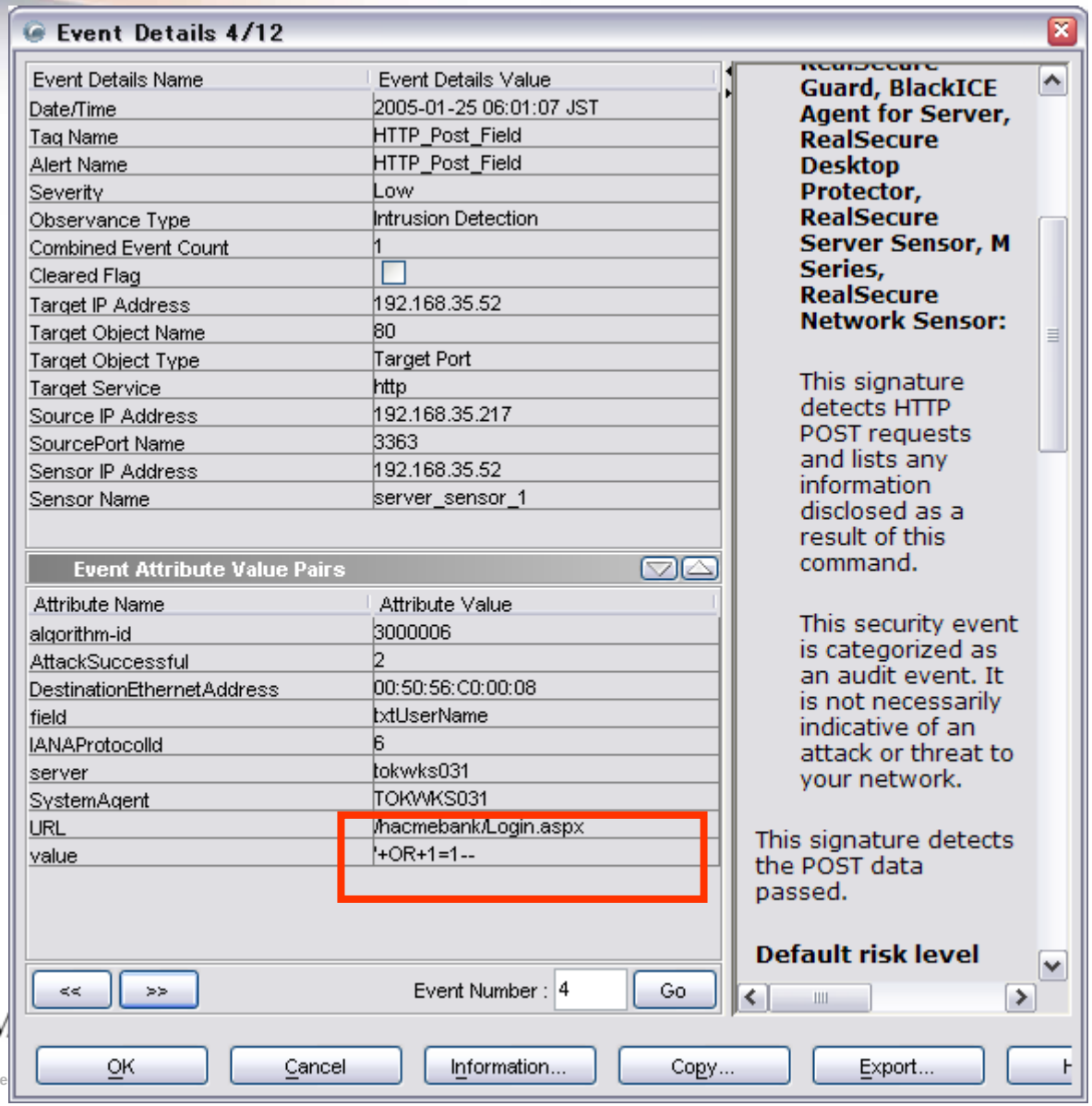
#Fields: date time s-ip cs-method cs-uri-stem cs-uri-query s-port cs-username c-ip cs(Cookie) sc-status sc-substatus sc-win32-status sc-bytes cs-bytes

2005-01-24 21:01:11 192.168.35.52 POST /hacmebank/Login.aspx - 80 - 192.168.35.217 CookieLoginAttempts=5;+ASP.NET_SessionId=4nhskaykui2jor45oqjzpm55 302 0 0 539 753

2005-01-24 21:01:11 192.168.35.52 GET /hacmebank/welcome.aspx - 80 - 192.168.35.217 CookieLoginAttempts=4;+ASP.NET_SessionId=4nhskaykui2jor45oqjzpm55 200 0 0 6657 529

IPSであれば・・・

- POSTメソッドにより渡されている文字列を正確に把握可能



IISログはどこまで有効か

- **有効な点**
  - **オーバーフローを狙った長大なリクエストの記録**
    - WebDAVに対する長大な要求など
    - ただし本当にバッファオーバーフローが発生したかどうかは判断不可能
  - **ディレクトリ・トラバーサルを悪用した攻撃**
    - 主にNimdaやCodeRedなどのワーム
- **無効な点**
  - Webアプリケーションの脆弱点に対する攻撃は記録できない
    - クロスサイト・スクリプティング
    - SQLインジェクションなど

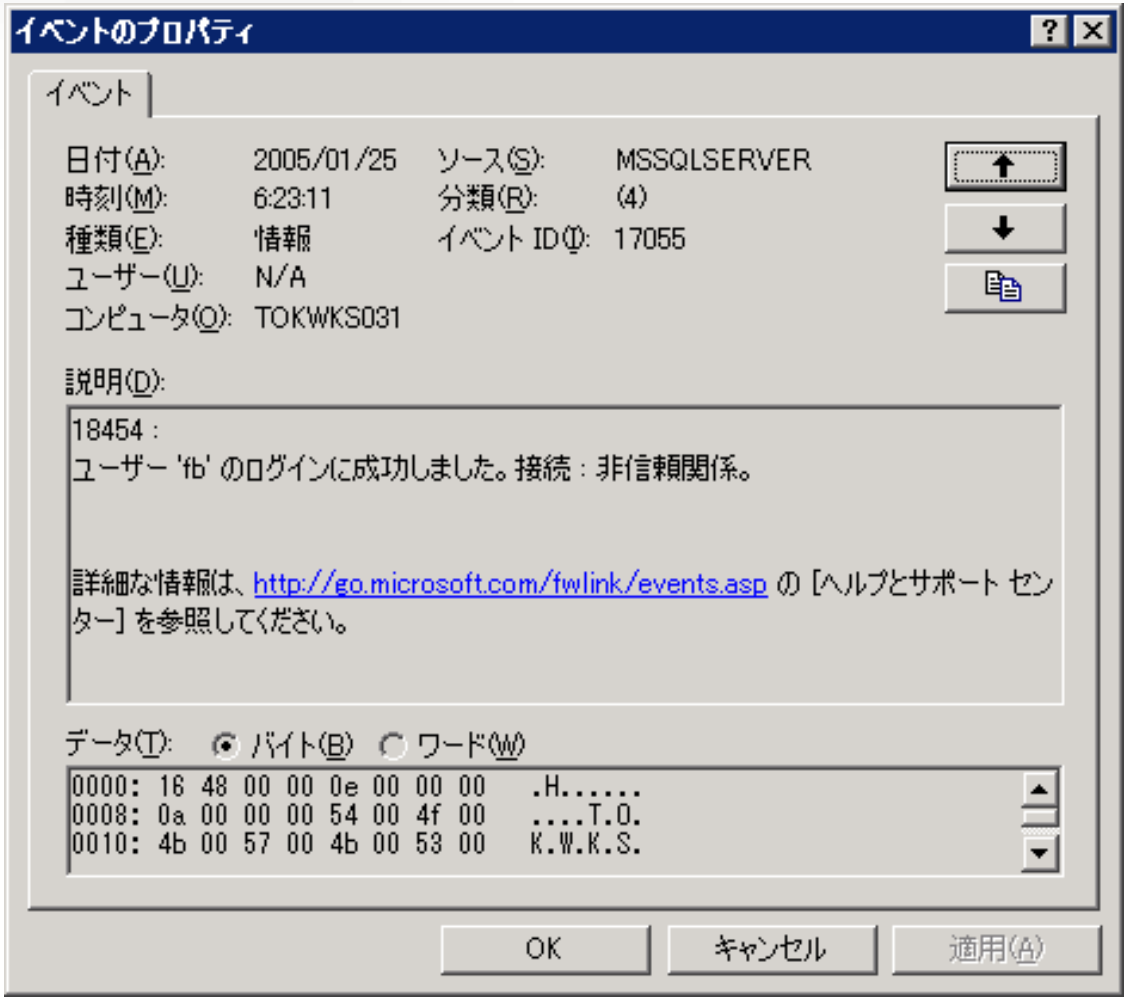
SQL Serverのログ

INTERNET | SECURITY | SYSTEMS®

SQL Serverのログ解析

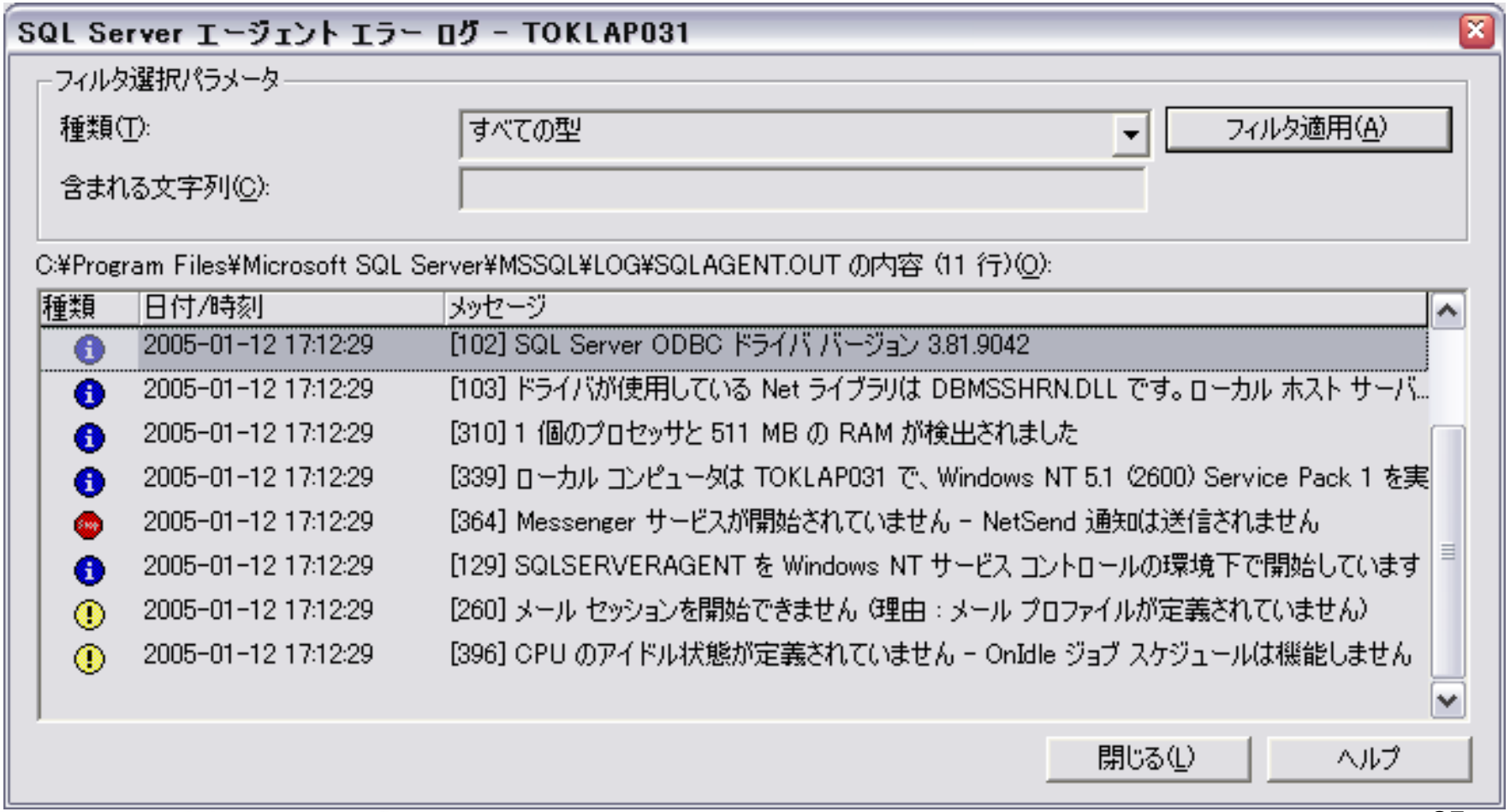
- **ログの種類**
  - **監査ログ**
    - SQL Serverユーザのログオン状況を監査
      - 失敗、成功、すべて
  - SQL Serverエラーログ
    - SQL Serverのステータスを保存
      - 例：SQLServer は 192.168.35.217: 1433 で受信を待っています。等
  - SQL Server Agentエラーログ
    - SQL Server Agentのエラーおよび警告を記録
  - トレース
    - SQL Serverに対するあらゆる操作を記録可能
    - 別途実施する必要あり
      - SQLプロファイラ
      - システムストアドプロシージャの呼び出し

SQL Server監査ログ

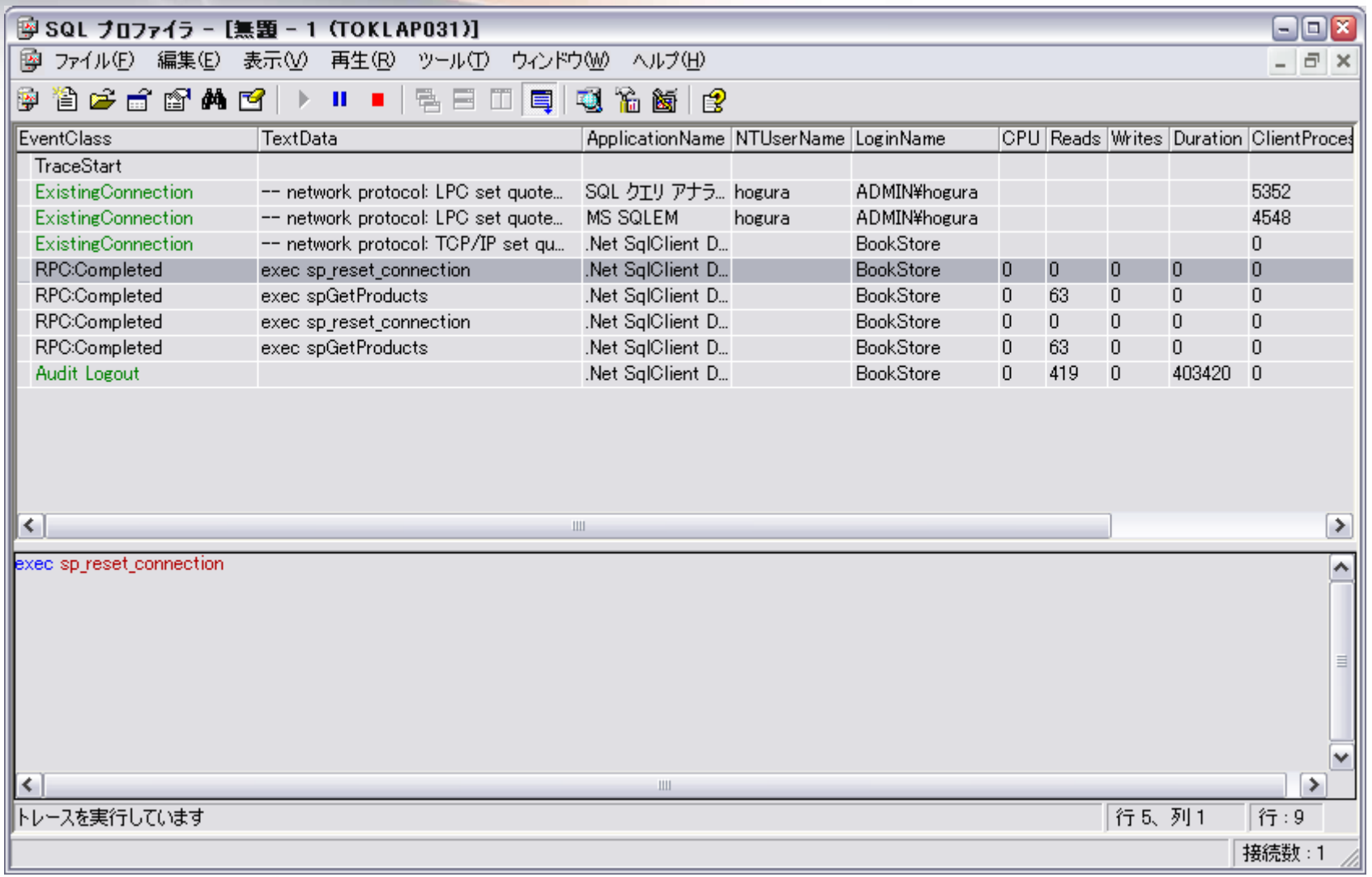


SQL Serverエラーログ

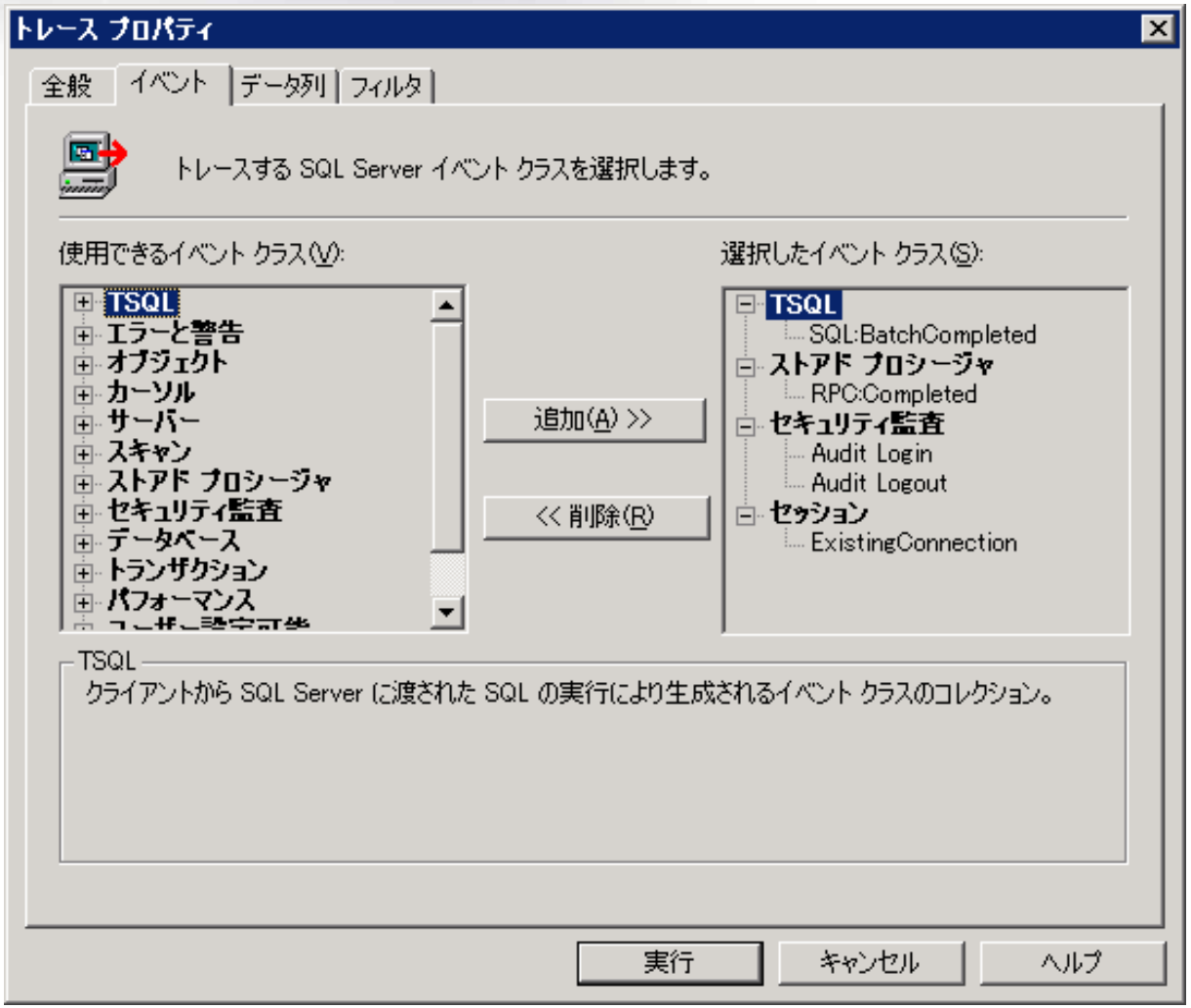
SQL Server Agentエラーログ



SQLプロファイラ



SQLプロファイラトレース対象



SQLプロファイラの目的

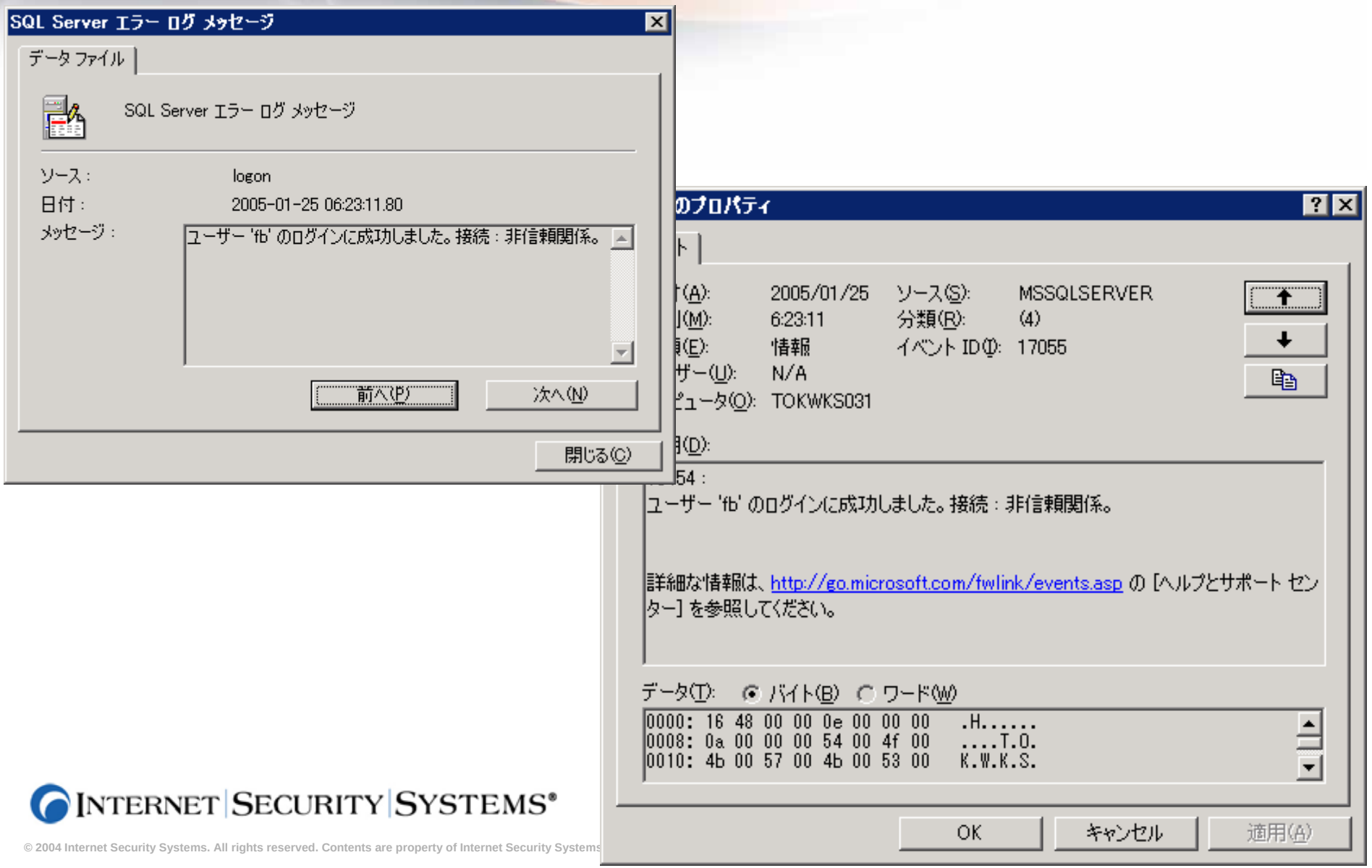
- SQL Serverのインスタンスのパフォーマンスの監視
- Transact-SQLステートメントおよびストアドプロシージャのデバッグ
- 実行に時間のかかるクエリの特定
- ステートメントを1ステップずつ実行してコードが期待どおりに機能するかどうかを確認する、プロジェクトの開発段階でのSQLステートメントおよびストアドプロシージャのテスト
- 稼動システムのイベントをキャプチャし、テストシステムでこれらのイベントを再生することにより、SQL Serverで行う問題のトラブルシューティング
- SQL Serverのインスタンスで発生した利用状況の監査と検討。セキュリティ管理者は、ログインの成否、ステートメントやオブジェクトへのアクセスで使用した権限の成否などを含めて、すべての監査イベントを検討できます。

※SQLプロファイラのヘルプより抜粋

SQLインジェクションの発見

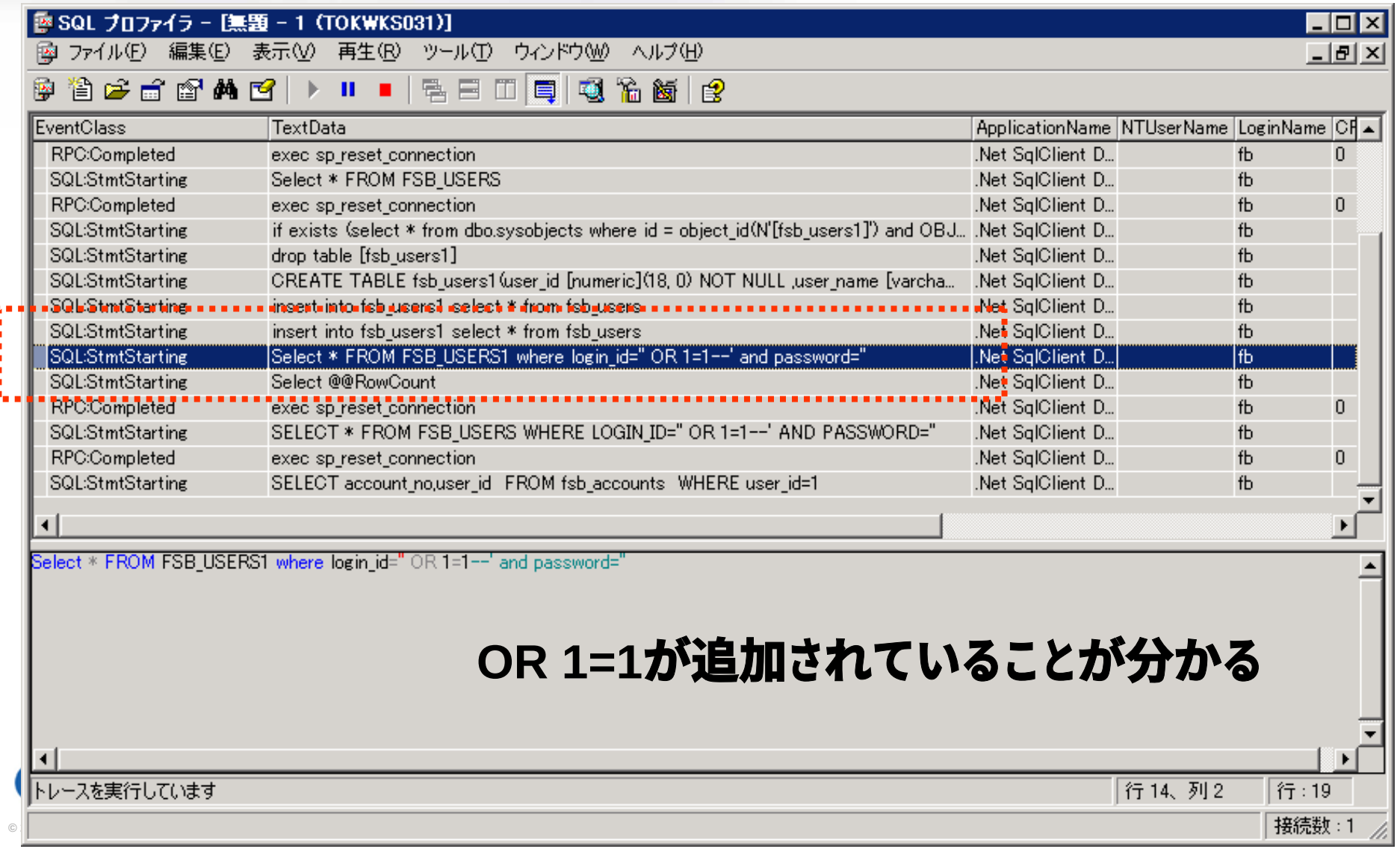
- SQLインジェクションを実施するのは正規アカウント
  - WebアプリケーションがSQL Serverに対してSQLインジェクションされたSQL文を発行 → 正規アカウントが利用される
  - イベントログやSQL Serverエラーログからは発見不可能
- 発行されるSQL文そのものの監査が必要
  - トレースの実施

ログインしか分からない・・・



INTERNET | SECURITY | SYSTEMS®

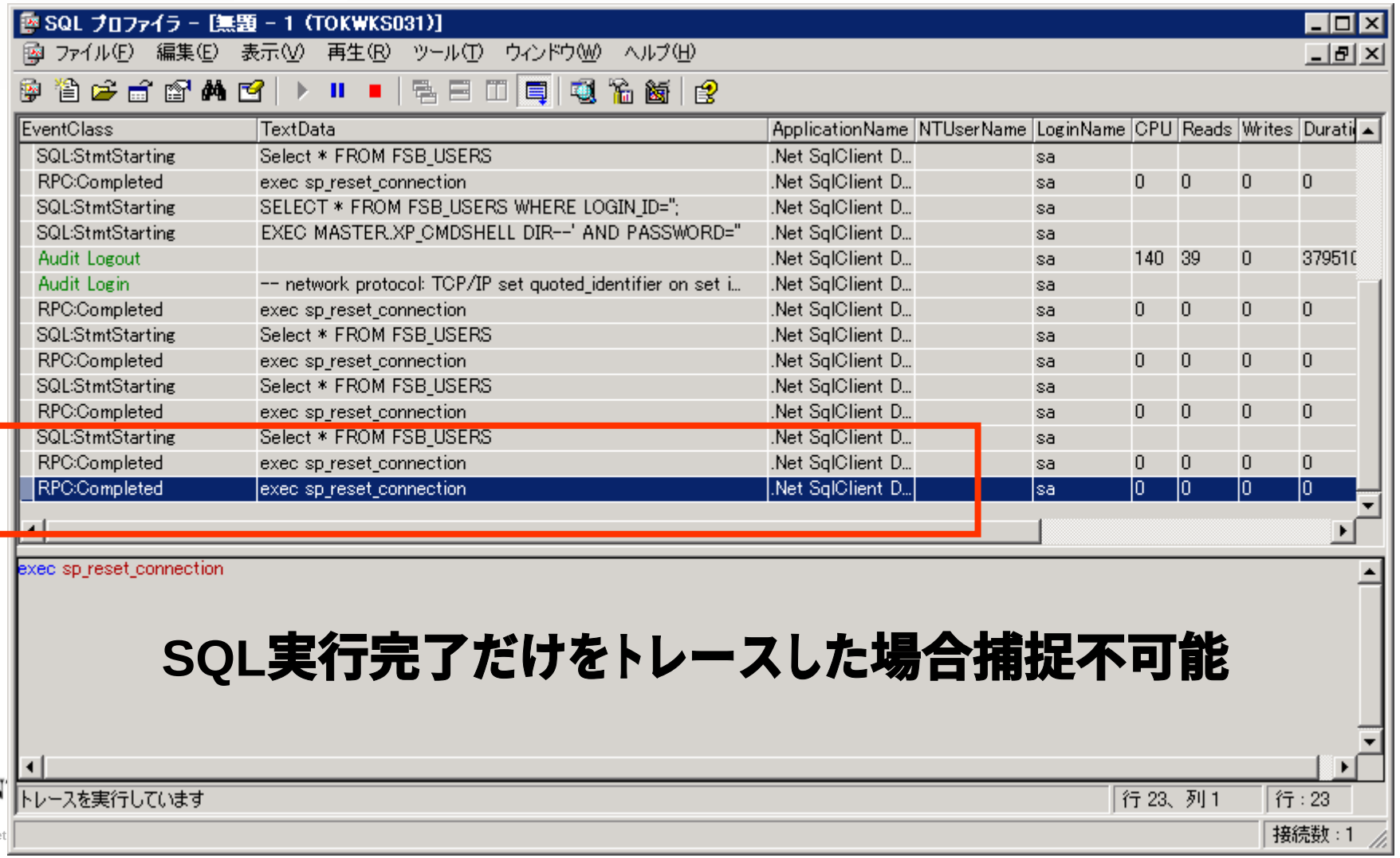
SQLプロファイラによるトレース例



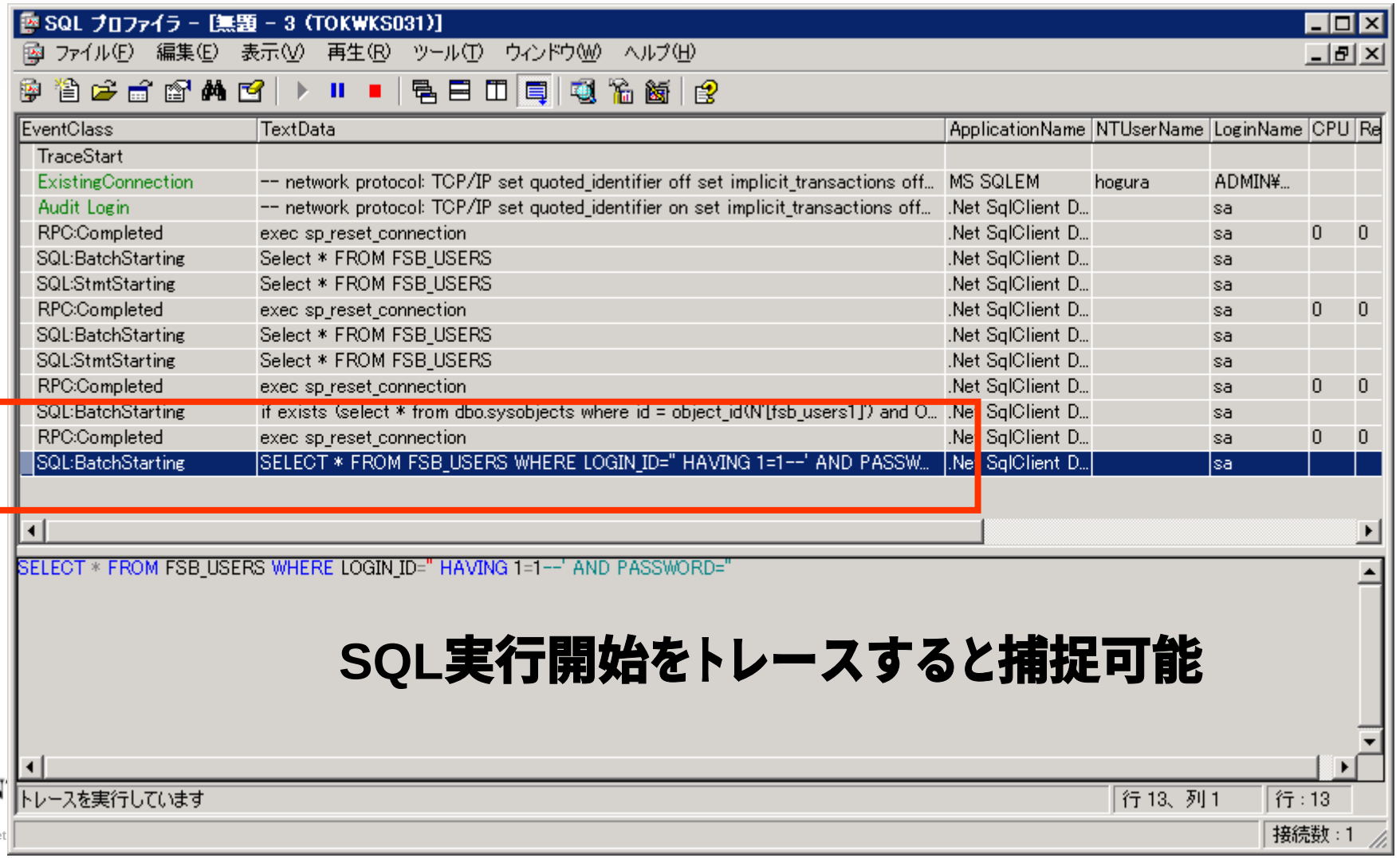
SQLインジェクションで起こしたエラーから情報を収集

- ‘ HAVING 1=1--をインジェクション
  - エラーメッセージ例
    - 列 ‘FSB_USERS.user_id’ が集計関数に含まれていない場合および GROUP BY 句がない場合は、選択リスト内では無効です。列 ‘FSB_USERS.user_name’ が集計関数に含まれていない場合および GROUP BY 句がない場合は、選択リスト内では無効です。列 ‘FSB_USERS.login_id’ が集計関数に含まれていない場合および GROUP BY 句がない場合は、選択リスト内では無効です。列 ‘FSB_USERS.password’ が集計関数に含まれていない場合および GROUP BY 句がない場合は、選択リスト内では無効です。列 ‘FSB_USERS.creation_date’ が集計関数に含まれていない場合および GROUP BY 句がない場合は、選択リスト内では無効です。
- FSB_USERSテーブルが存在
- user_id, user_name, login_id, password, creation_date列が存在

エラーによる情報取得の捕捉



エラーによる情報取得の捕捉

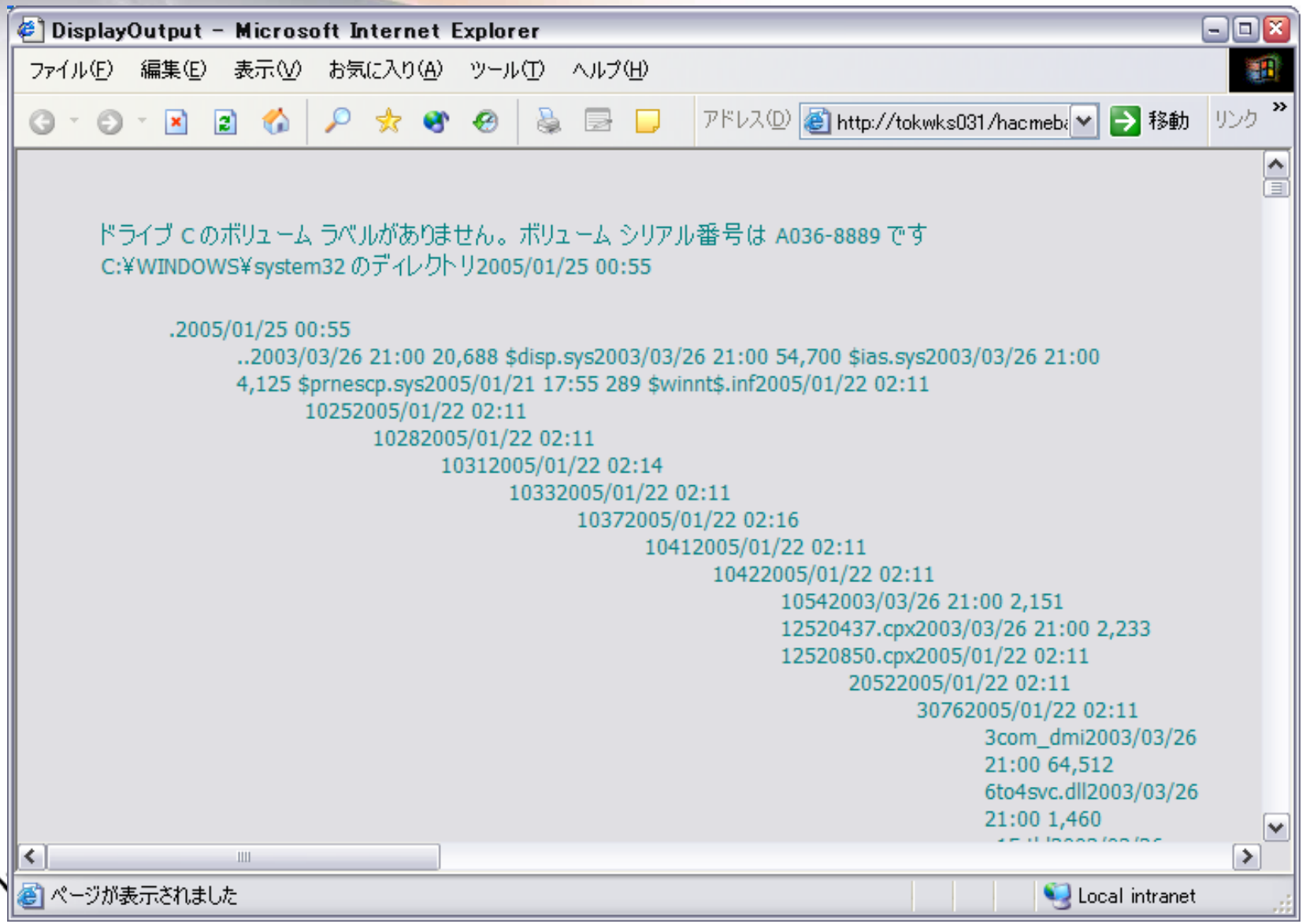


SQL実行開始をトレースすると捕捉可能

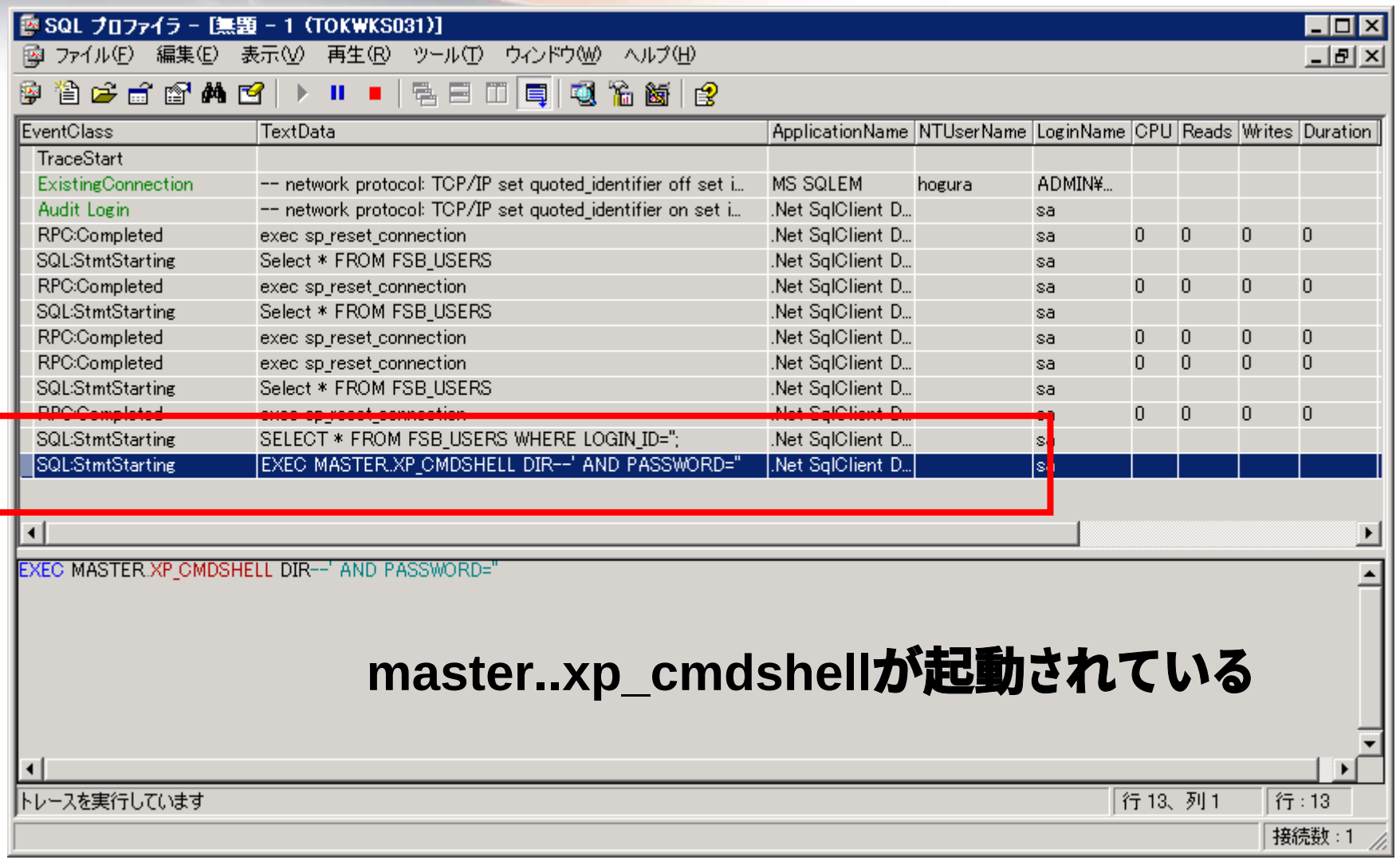
SQLインジェクションによる任意のコマンド実行

- **master..xp_cmdshell拡張ストアドプロシージャを悪用**
  - ‘; exec master..xp_cmdshell dir--
  - **;を先頭に入れることで、1行に複数のSQL文を入れることが可能**
  - xp_cmdshellを通じてdirコマンドを実行
    - **任意のコマンドを実行可能**

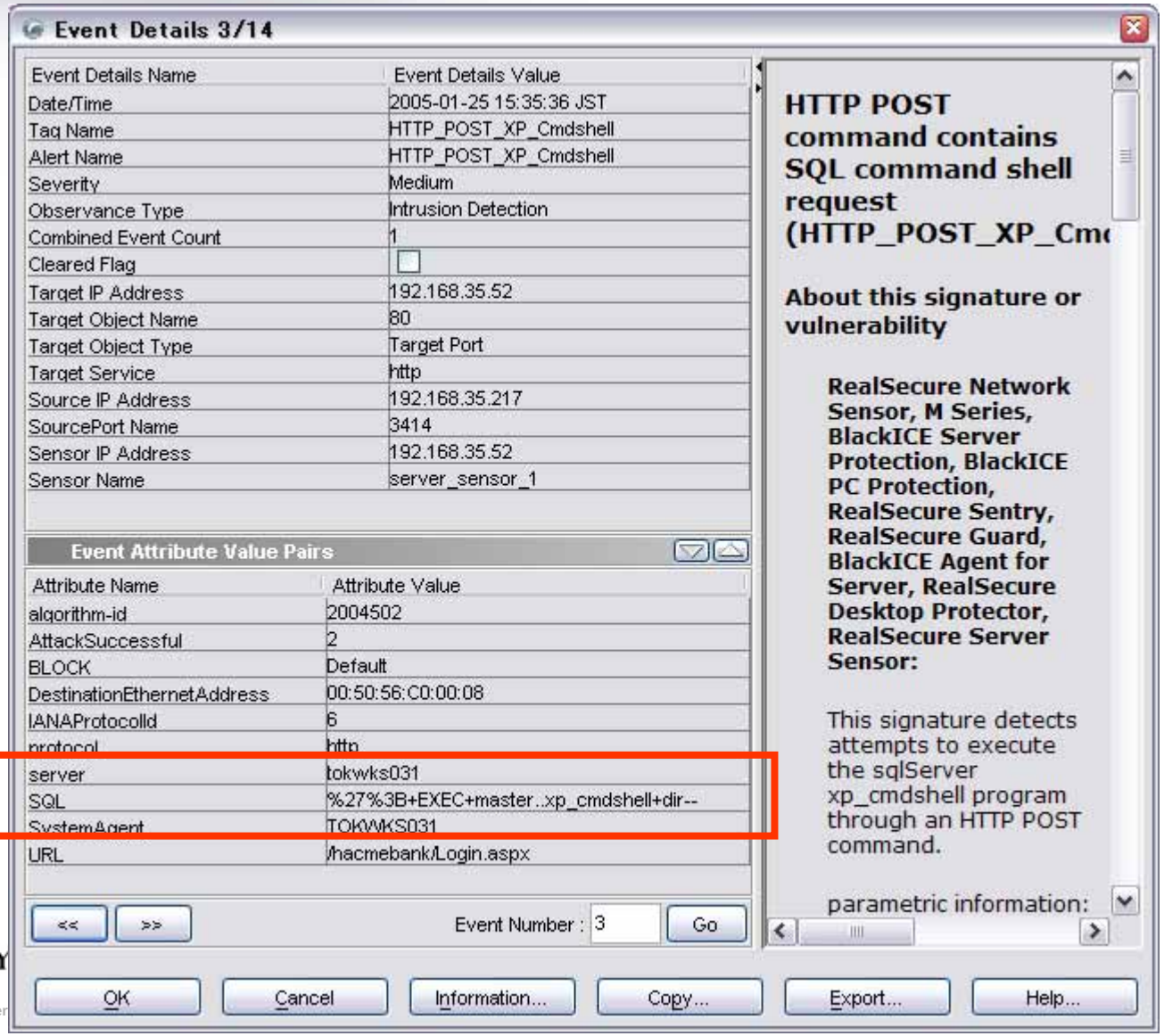
SQLインジェクションによるdir



トレースによるxp_cmdshellの実行記録



SQLインジェクションによる xp_cmdshellプロシージャの実行検知



INTERNET SECURITY SY

トレースの問題

- **SQLプロファイラによるトレース**
  - SQLプロファイラはSQL Serverのクライアントプログラムの一種
    - 常時接続クライアントが追加される
    - SQLプロファイラ自体はSQL Serverとは異なるホストで実行可能
  - SQL Server自体のパフォーマンス問題につながる
  - トレースの設定によっては捕捉できない状況が発生
- **システムストアドプロシージャの利用**
  - SQL文を実行する前にシステムストアドプロシージャにより部分的にトレースを実施
  - 部分的に実施するためSQLプロファイラを実行しておく場合よりもSQL Serverに対する影響は僅か
  - アプリケーション開発者が意図的にシステムストアドプロシージャを利用しなければならない

参考

- **Webハッキングトレーニングアプリケーション**
  - FoundstoneのHacme Bank™
  - http://www.foundstone.com/resources/proddesc/hacmebank.htm
  - **簡単にSQLインジェクションやクロスサイト・スクリプティングをテスト可能**
  - **実行後のIISやSQL Serverのログの確認に利用可能**
  - **環境**
    - Microsoft .NET Framework 1.1
    - IIS
    - MSDE 2000もしくはSQL Server 2000
    - Microsoft ASP.NET Web Matrixがあればなお良い