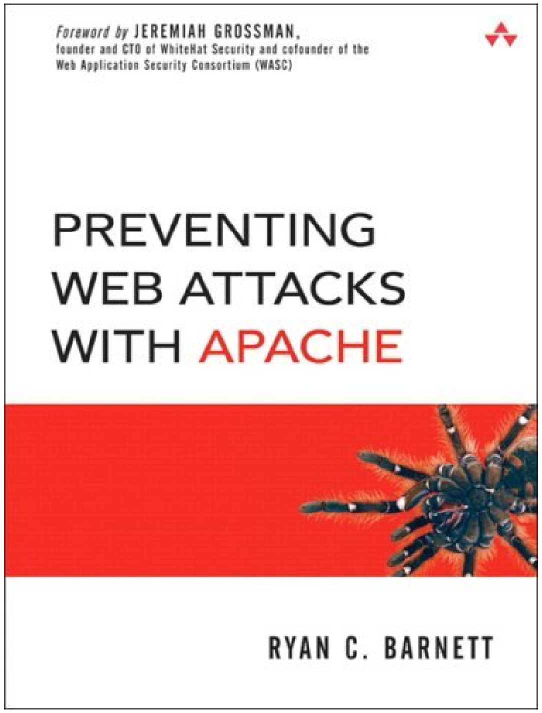
BREACH™

ModSecurity 2.0 Webcast:

Answers To Common Questions

Who am I?

- Breach Security
  - ModSecurity Community Manager
  - Director of Application Security Training
- Developing ModSecurity Courses and a Certification (Estimated Q1 2007 availability)
  - Deployment and Management
  - Rules Writing Workshop
  - Breach Certified ModSecurity Expert (BCME)
- Courseware Developer/Instructor for the SANS Institute
- Center for Internet Security's Apache Benchmark Project Team Leader
- Web Application Security Consortium (WASC) Member
- Author of Preventing Web Attacks with Apache (Addison/Wesley)


Foreword by JEREMIAH GROSSMAN,
founder and CTO of WhiteHat Security and cofounder of the Web Application Security Consortium (WASC)
PREVENTING WEB ATTACKS WITH APACHE
RYAN C. BARNETT

Version Poll Question

- Who is not currently using ModSecurity?
  - Assuming you are interested in deploying a WAF but need more information
- Who is using ModSecurity 1.X?
  - Assuming you are considering switching to 2.0 but need more information
- Who is using ModSecurity 2.0?
  - Assuming you have already switched or are testing 2.0, but you also need more information on its usage
- Who is using both ModSecurity 1.X and 2.0?
  - Assuming you have both Apache 1.X and 2.0 servers

Common Question:

What’s new in ModSecurity 2.0?

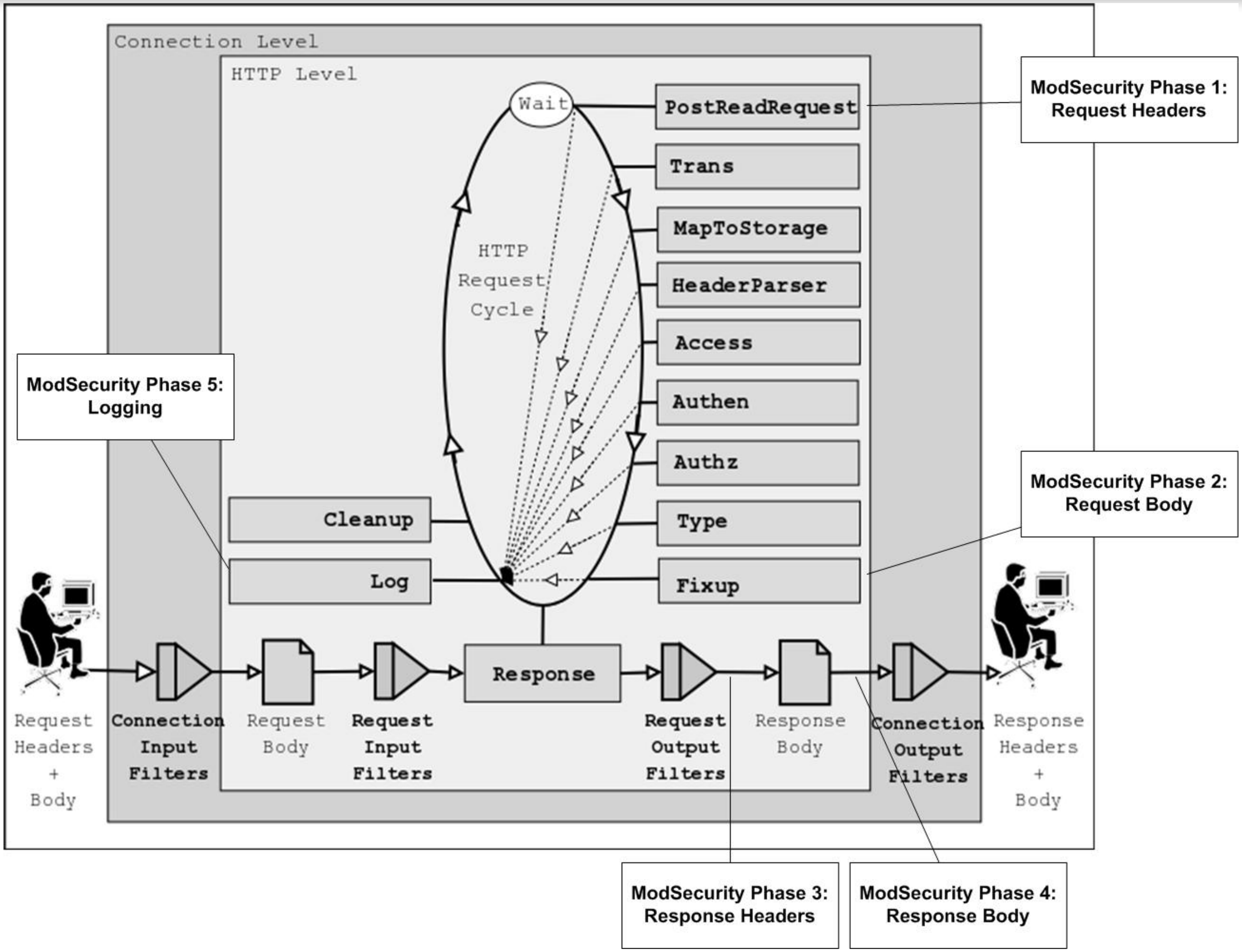
New ModSecurity 2.0 Features

- Five processing phases (where there were only two in 1.9.x). These are: request headers, request body, response headers, response body, and logging. Those users who wanted to do things at the earliest possible moment can do them now.
- Per-rule transformation options (previously normalization was implicit and hard-coded). Many new transformation functions were added.
- Transaction variables. This can be used to store pieces of data, create a transaction anomaly score, and so on.
- Data persistence (can be configured any way you want although most people will want to use this feature to track IP addresses, application sessions, and application users).
- Support for anomaly scoring and basic event correlation (counters can be automatically decreased over time; variables can be expired).
- Support for web applications and session IDs.
- Regular Expression back-references (allows one to create custom variables using transaction content).
- There are now many functions that can be applied to the variables (where previously one could only use regular expressions).
- XML support (parsing, validation, XPath).

New Features – Processing Phases

- Five processing phases (where there were only two in 1.9.x)
  - Request headers
  - Request body
  - Response headers
  - Response body
  - Logging
- Those users who wanted to do things at the earliest possible moment can do them now
- This many phases allow you to decide what you want to happen at key points of transaction processing

ModSecurity 2.0 Processing Phases



ModSecurity Phase 1:
Request Headers

ModSecurity Phase 2:
Request Body

ModSecurity Phase 3:
Response Headers

ModSecurity Phase 4:
Response Body

ModSecurity Phase 5:
Logging

New Features - Transformation Functions (1)

- Transformation functions will automatically convert data before matching
- Previously, normalization was implicit and hard-coded
- Many new functions were added

lowercase

replaceNulls

compressWhitespace

replaceComments

urlDecode

urlDecodeUni

base64Encode

base64Decode

hexDecode

hexEncode

htmlEntityDecode

escapeSeqDecode

normalisePath

normalisePathWin

md5

sha1

New Features - Transformation Functions (2)

- The following is performed by default (and in this order):
  - **lowercase**
  - **replaceNulls**
  - **compressWhitespace**
- But you can change the default setting for all subsequent rules:

```
SecDefaultAction log,deny,status:500,\
t:replaceNulls,t:compressWhitespace
```

- Or, just for one rule:

```
SecRule ARG:base64 ABC t:base64decode
```

- **Be aware of the combined effect of altering these settings**
  - May not be what you expect!
  - Best to turn the SecDebugLogLevel to 9 then test to verify

BREACH

New Features – Data Persistence/Collections

- Can now track multiple requests!
- Can be configured any way you want although most people will want to use this feature to track IP addresses, application sessions, and application users
- **initcol** – persistent collection based on source IP:

```
SecAction initcol:ip=%{REMOTE_ADDR},nolog,pass
```

- **setsid** – session storage based on app session ID:

```
SecRule REQUEST_COOKIES:PHPSESSID !^$
    chain,nolog,pass
SecAction setsid:%{REQUEST_COOKIES.PHPSESSID}
```

The Advantage of Variables

- Variables allow you to move from the "all-or-nothing" type of rules to a more sensible anomaly-based approach.
- The all-or-nothing approach works well when you want to prevent exploitation of known problems or enforce positive security, but it does not work equally well for anomaly detection or multiple request issues (DoS).
- For the latter it is much better to establish a per-transaction anomaly score and have a multitude of rules that will contribute to it.
- Then, at the end of your rule set, you can simply test the anomaly score and decide what to do with the transaction: reject it if the score is too large or just issue a warning for a significant but not too large value.

Variable Actions

- Working with variables:

```
# set the IP collection score to 10
setvar:ip.score=10
# increase the IP collection score by 5
setvar:ip.score=+5
# remove the IP collection score
setvar:!ip.score
# decrease the IP collection score by 60 points every hour
deprecatevar:ip.score=60/3600
# expire a blocked IP collection after an hour
expirevar:ip.blocked=3600
```

Full Example – Initcol/Variables

```
#Specify the local directory for collection storage
SecDataDir /path/to/apache/logs/state

# Initiate a collection based on the source IP address
SecAction initcol:ip=%{REMOTE_ADDR},nolog,pass

# Increase the IP collection score based on filter hits
SecRule REQUEST_FILENAME “/cgi-bin/phf" pass,setvar:ip.score=+10
SecRule REQUEST_FILENAME “cmd.exe” pass,setvar:ip.score=+10
SecRule REQUEST_METHOD “TRACE” pass,setvar:ip.score=+5

# Evaluate the overall IP collection score
SecRule IP:SCORE "@ge 30"
```

BREACH

Common Question:

What type of security models does ModSecurity 2.0 support?

ModSecurity Protection Models (1)

1. **Negative security model**
   - Looking for bad stuff/known attack signatures
   - Core Rules – modsecurity_crs_40_generic_attacks.conf
2. **Positive security model**
   - Verifying input is correct.
   - Core Rules
     - modsecurity_crs_20_protocol_violations.conf.
     - modsecurity_crs_30_http_policy.conf
3. **Anomaly-based Model**
   - Must be able to identify abnormal requests/responses
   - Rules can be created to increase anomaly score based on 4XX/5XX level status codes

ModSecurity Protection Models (2)

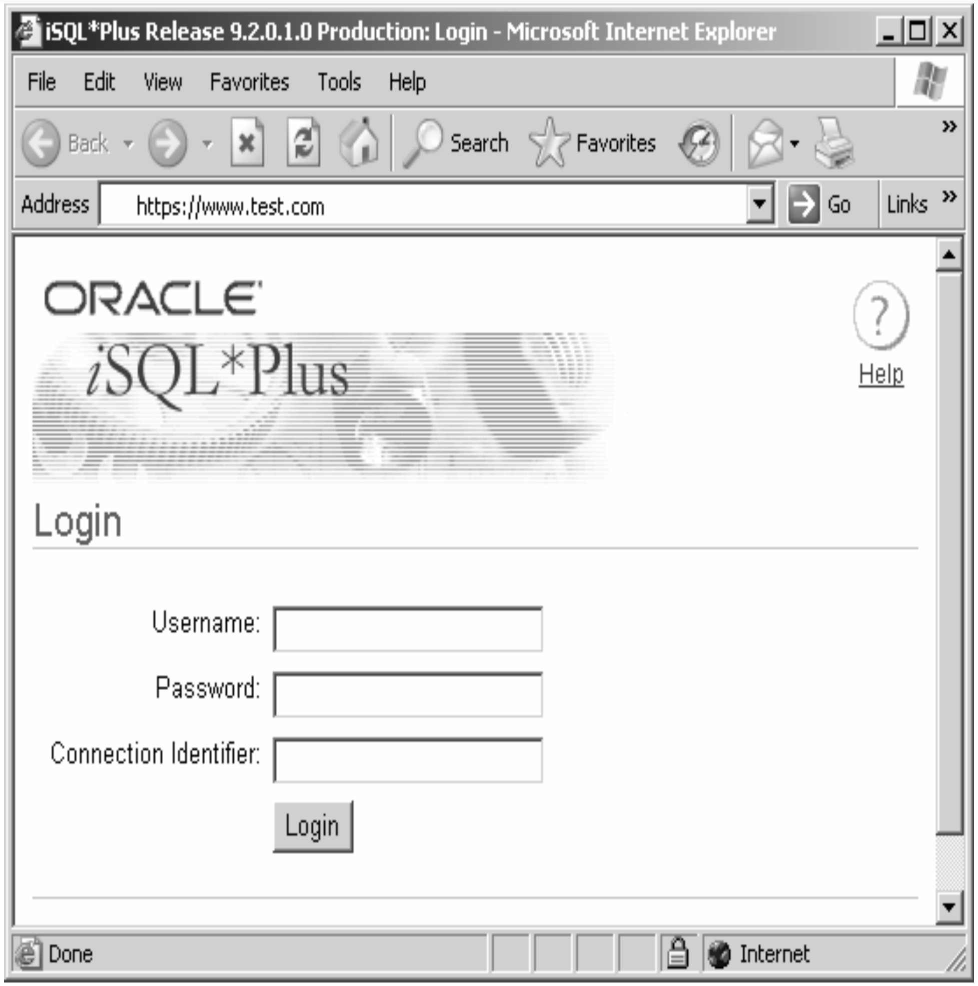
4. **External patching**
   - Also known as "just-in-time patching" or "virtual patching“
   - Provides immediate protection from identified vulnerabilities
5. **Extrusion Detection Model**
   - Monitoring outbound data to ensure sensitive information does not leave your network (i.e. – Information Leakage)
   - Core Rules – modsecurity_crs_50_outbound.conf
6. **Heuristic-based Model**
   - Statistical calculation which correlates various detection methods – signature match + RBL status
   - ModSecurity now has this capability with the following new features:
     - Data persistence
     - Transactional/Session scoring
     - Blocking can be based on an overall score

Common Question:

Can you show me some examples of ModSecurity 2.0 fixing specific issues?

Virtual Patch Example - Oracle iSQL*Plus buffer overflow

- **CVE 2002-1264** - Buffer overflow in Oracle iSQL*Plus web application of the Oracle 9 database server allows remote attackers to execute arbitrary code via a long USERID parameter in the isqlplus URL.
- **Oracle Response -** There is no workaround to address the potential security vulnerability identified above. Some patches took 2 months
- **With ModSecurity**, a translated Snort signature could have been used to implement an **Immediate Patch**


iSQL*Plus Release 9.2.0.1.0 Production: Login - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Address
Go
Links
ORACLE
iSQL*Plus
Help
Done
Internet

https://www.test.com

Login

Username:

Password:

Connection Identifier:

Login

Convert Snort Signature to ModSecurity Format

- Snort Signature

```
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS
  (msg:"WEB-MISC Oracle iSQLPlus login.uix username
  overflow attempt"; flow:to_server,established;
  uricontent:"/login.uix"; nocase;
  pcre:"/username=[^&\x3b\r\n]{250}/smi";
  reference:bugtraq,10871;
  reference:url,www.nextgenss.com/advisories/ora-
  isqlplus.txt; classtype:web-application-attack; sid:2703;
  rev:1;)
```

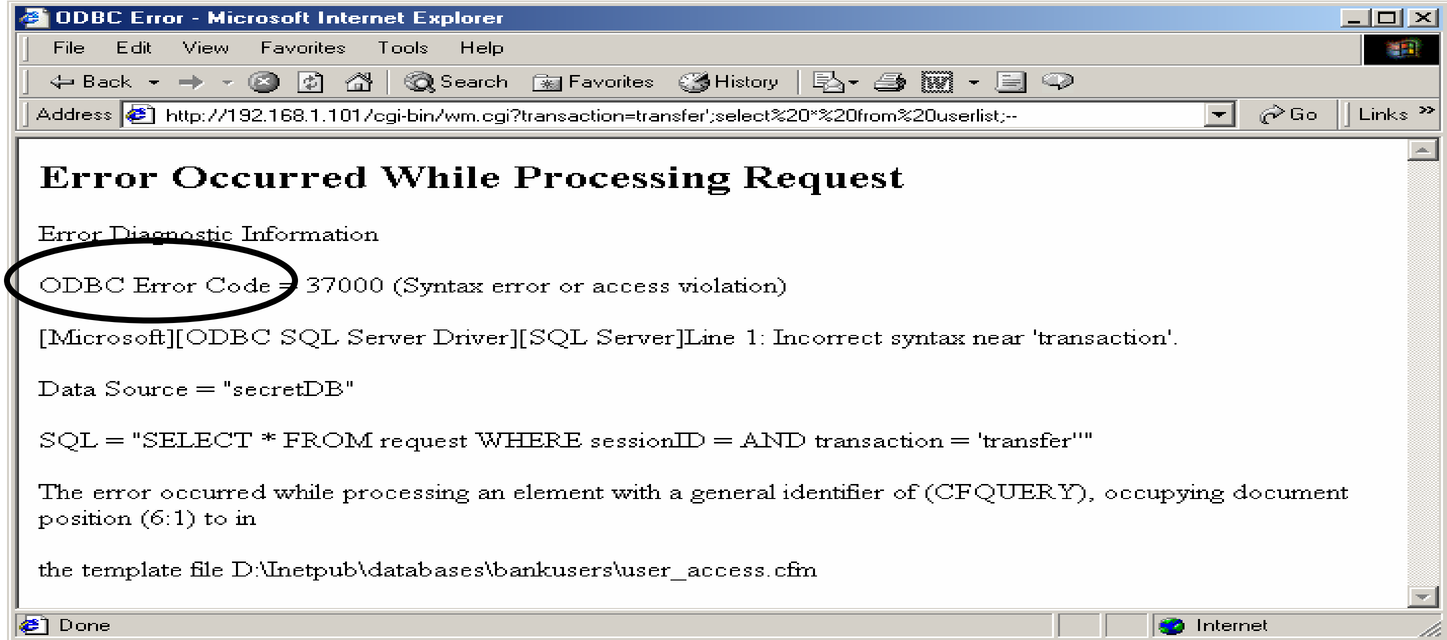
- ModSecurity Rule

```
SecRule REQUEST_URI “login.uix$” “chain,deny,status:403,\
phase:2,msg:’Oracle iSQLPlus login.uix username overflow
  attempt’”
SecRule REQUEST_BODY “username=[^&\x3b\r\n]{250}”
```

Extrusion Detection – DB Errors

- ModSecurity can identify and block outbound data such as error messages from back-end databases

```
SecRule RESPONSE_BODY “ODBC Error Code”
“deny,log,status:503,phase:4,msg:’Database
Error Message Detected’”
```


ODBC Error - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address
Go
Links
Done
Internet

http://192.168.1.101/cgi-bin/wm.cgi?transaction=transfer';select%20*%20from%20userlist;--

Error Occurred While Processing Request

Error Diagnostic Information

ODBC Error Code = 37000 (Syntax error or access violation)

[Microsoft][ODBC SQL Server Driver][SQL Server]Line 1: Incorrect syntax near 'transaction'.

Data Source = "secretDB"

SQL = "SELECT * FROM request WHERE sessionID = AND transaction = 'transfer''"

The error occurred while processing an element with a general identifier of (CFQUERY), occupying document position (6:1) to in

the template file D:\Inetpub\databases\bankusers\user_access.cfm

Denial of Service Protections

- Apache module - Mod_Evasive
  - Mod_Evasive does not use shared memory and can only identify multiple requests on the current httpd process
  - Evasion possibilities if client does not use Keep-Alives and forces the web server to spawn new processes for each request
- ModSecurity - SecGuardianLog directive + httpd-guardian perl script
  - Can identify and react to both DoS and Brute Force Attacks by monitoring the speed of requests
- Initcol – Use the “@ge” operator to evaluate the IP.UPDATE_RATE built-in collection variable.
  - SecRule IP.UPDATE_RATE “@ge 100”

Common Question:

How do I install ModSecurity 2.0?

ModSecurity 2.0 Installation

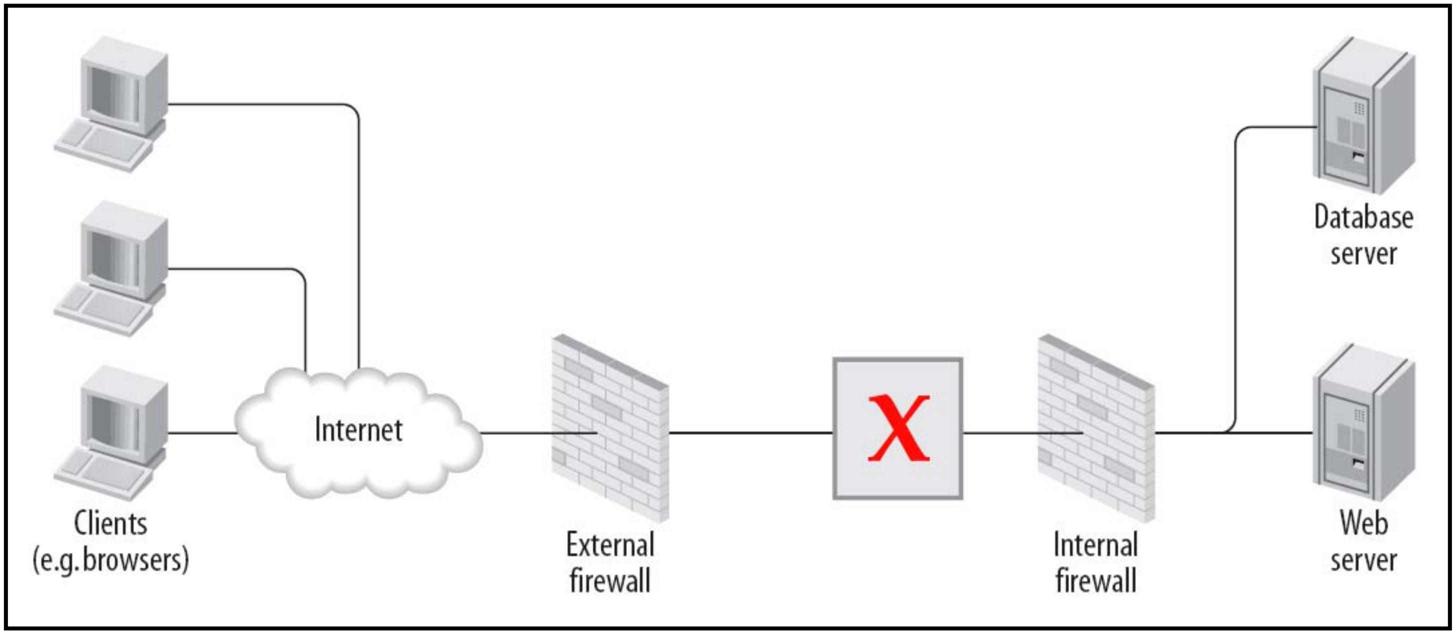
- Currently, ModSecurity 2.0 can only be installed as a DSO module in Apache
  - There are plans to update the configuration/compilation code to allow for static installations with Apache
- Although it is installed as a DSO, do not use apxs directly
  - ModSecurity 2.0 has a standard Makefile for compilation
  - It does use apxs behind the scenes
- Update the Makefile “top_dir” setting with the correct path to your Apache ServerRoot directory
- Use make and make install

Common Question:

How can I deploy ModSecurity?

Reverse Proxy Deployment

- Open Source ModSecurity users can deploy software on an Apache server acting as a Reverse Proxy server
- Breach ModSecurity Pro M1000 appliance



Reverse Proxy Deployment Pros/Cons

Pros

- Single point of access – choke points for applying security settings and makes management easier
- Increased performance – if SSL accelerators/caching used
- Network isolation – divides web into multiple tiers
- Network topology hidden from the outside world
- You can implement “Virtual Patches” to protect vulnerable web apps that either don’t have a patch available or where the code can not be changed

Cons

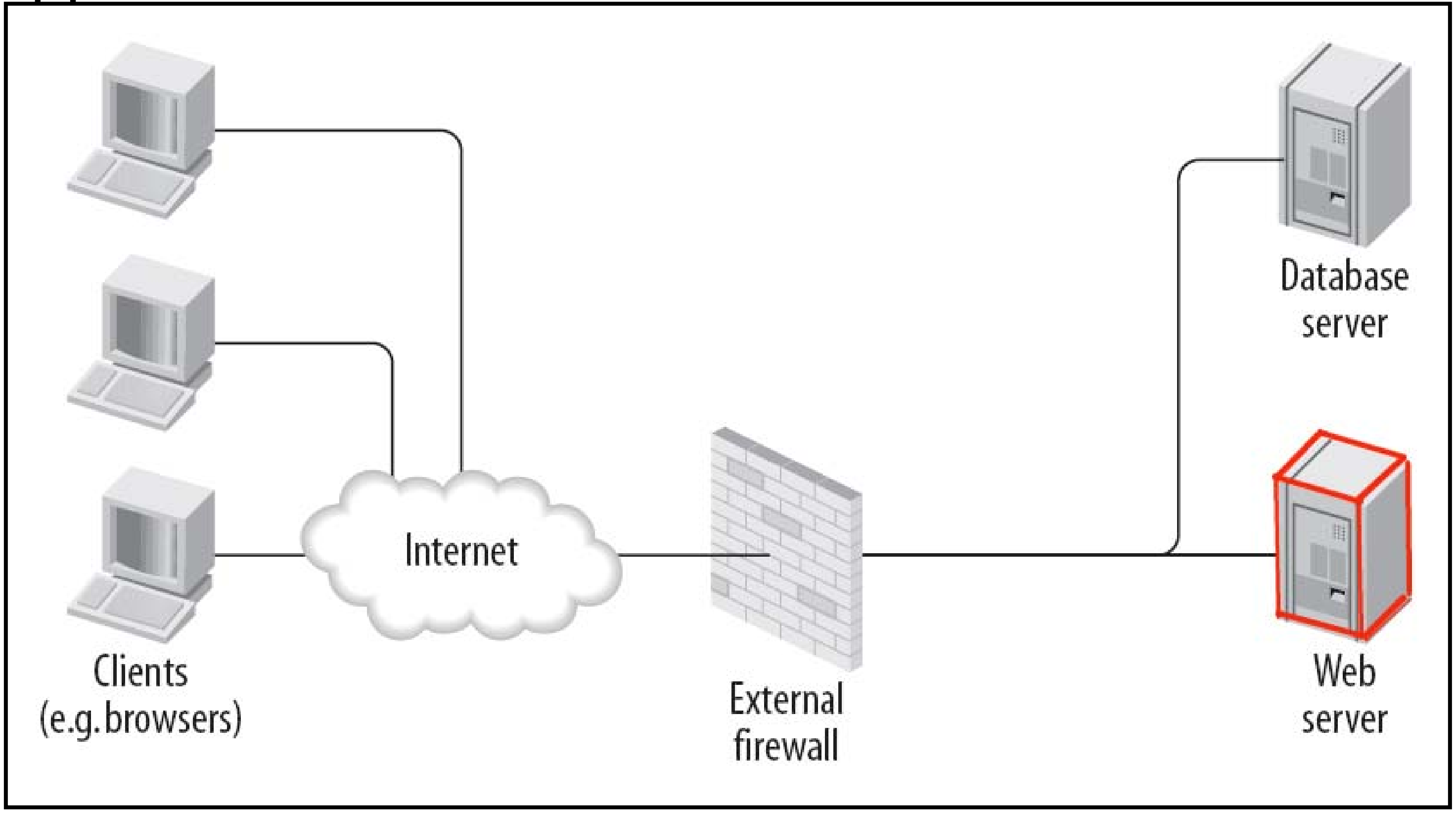
- A potential bottleneck
- Point of failure
- Requires changes to network (unless it's a transparent reverse proxy)
- Must terminate SSL (can be a problem if application needs to access client certificate data)

M1000 vs. Building Your Own Appliance

- Building a ModSecurity reverse proxy appliance is non-trivial. Need specific skill sets - expert in Apache, web application security, and ModSecurity.
- The M1000 is a Breach certified appliance, hardened for security, and updated with new releases for bug fixes and optimizations.
- M1000 includes certified rule sets guaranteed to be accurate and efficient. Includes regulator/application rule sets (PCI and OWA).
- The M1000 includes a graphical user interface with comprehensive alert management, configuration, and reporting facilities (HTML, PDF) that lower the total cost of ownership of the solution.
- The M1000 includes first year support and maintenance at no additional charge. In addition to leveraging the ModSecurity community, customers of the M1000 have access to the core developers and the world's experts in ModSecurity.

Embedded Mode Deployment

- Open Source ModSecurity users can deploy software on an Apache server to protect the local server and web application



Embedded Mode Deployment

Pros

- Easy to add
  - No network changes required
  - Little to no cost as no new hardware is required
- Not a point of failure.
  - Only protecting the local web server

Cons

- Only protects the local server
- Uses web server resources
- Management of configurations and log files is more difficult as you have multiple installs

Deployment Poll Question

- What deployment modes are you currently using?
  - Embedded mode
  - Reverse Proxy
  - Mixed/Both

Common Question:

How do I migrate from 1.X to 2.0?

Migration Consideration/Issues

- ModSecurity 2.0 only works with the Apache 2.X branch (no current code/support for Apache 1.X version)
  - If you are running Apache 2.X, then there is no problem.
  - If you have Apache 1.X hosts, then you can still use ModSecurity 2.0, but you would have to use it in a Reverse Proxy front-end.
- Rules Migration Issues
  - Must change SecFilter/SecFilterSelective to SecRule equivalents
  - Need to specify the correct processing phases/locations/transformation functions
- Integrating Custom Rules with Core Rules
  - Should specify your custom rules in a separate file
  - Call them up after the modsecurity_crs_10_config.conf file, but before the http protocol rules

Common Question:

Why are these new rules killing my server performance?

Performance Considerations – Writing Rules

- The number of rules used will impact performance, so be wary of implementing too many/unneeded negative filter rules
  - "GotRoot-Effect"
  - Converted Snort rules
- Joining/Grouping multiple rules into one RegEx line increases performance (Perl "Or'ing feature with the "|" character)
  - Instead of individual lines –
    - SecRule REQUEST_URI "file1\.cgi"
    - SecRule REQUEST_URI "script2\.cgi"
  - Combine/Group them based on LOCATION
    - SecRule REQUEST_URI "(file1|script2)\.cgi"
- Optimized RegEx rules (such as using (?:xxx) can cut the validation time by 50%
  - SecRule REQUEST_URI "(?:(script[1-3]|file[1-3])\.cgi)"

BREACH

Common Question:

How do I manage my ModSecurity Installations?

Log Management Poll Question

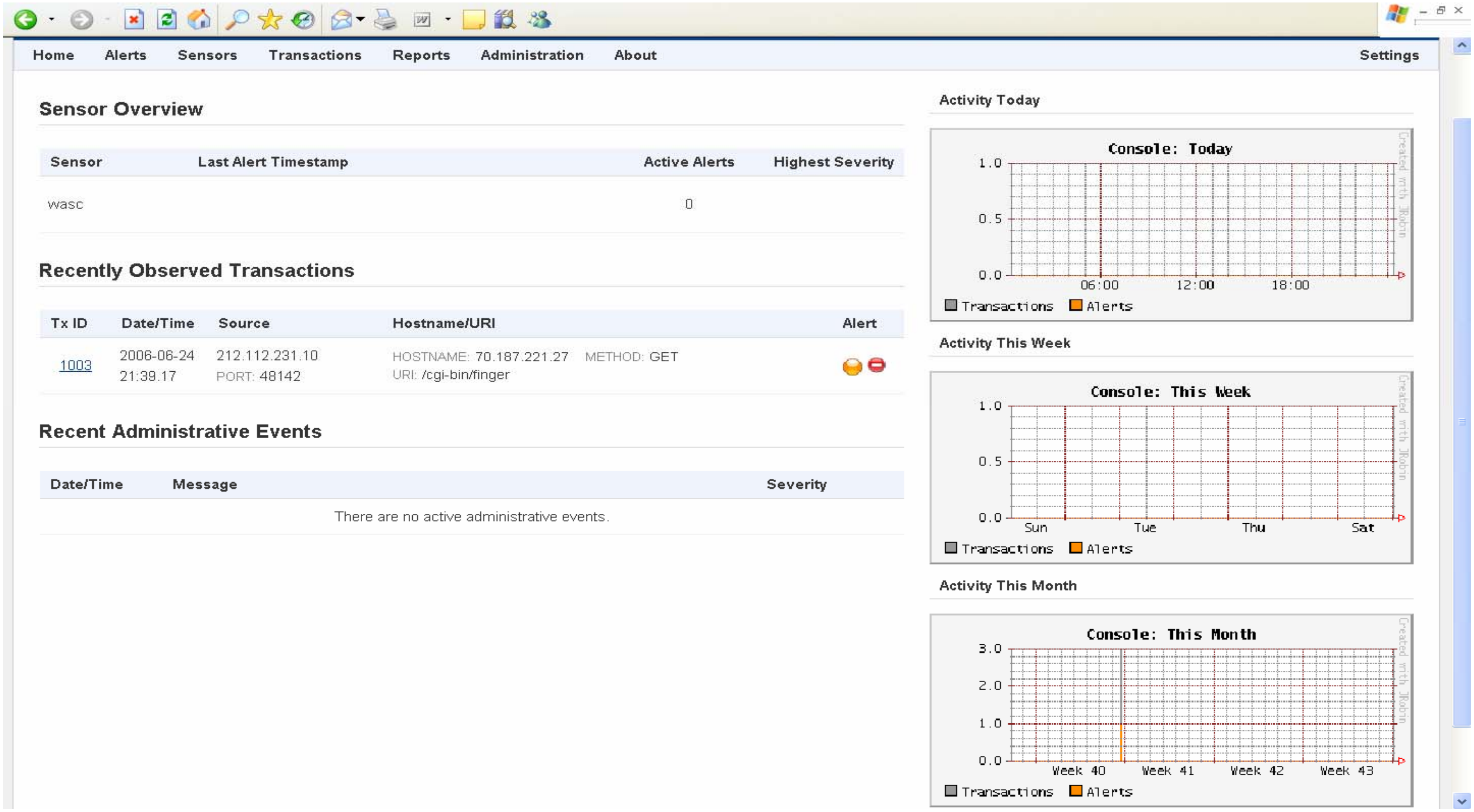
- How are you managing your ModSecurity logs?
  - Keeping all logs local on the web server/proxy
  - Sending Apache/ModSecurity logs to a remote host via Syslog
  - Sending data to a SIM host (LogLogic/Intellitactics, etc…)
  - Using the ModSecurity Console
  - Would like to use the ModSecurity Console, however the sensor limit is hindering me (I have a large ModSecurity deployment)

ModSecurity Console (1)

- Log & alert centralization solution, can capture alerts or entire traffic streams.
- Daemon with a GUI (web application).
- Single package (comes with its own web server and database).
- Runs on all platforms that support Java 1.4 or better.
- Can received logs form a limited number or remote sensors
- Does not provide Command and Control of remote sensor configs
- M1000 Appliance also has Command and Control of the configs
- Will evolve into **Enterprise Manager Console**.

ModSecurity Console

Console Home Page



Sensor Overview

Sensor	Last Alert Timestamp	Active Alerts	Highest Severity
wasc		0	

Recently Observed Transactions

Tx ID	Date/Time	Source	Hostname/URI	Alert
1003	2006-06-24 21:39.17	212.112.231.10 PORT: 48142	HOSTNAME: 70.187.221.27 METHOD: GET URI: /cgi-bin/finger	

Recent Administrative Events

Date/Time	Message	Severity
	There are no active administrative events.	

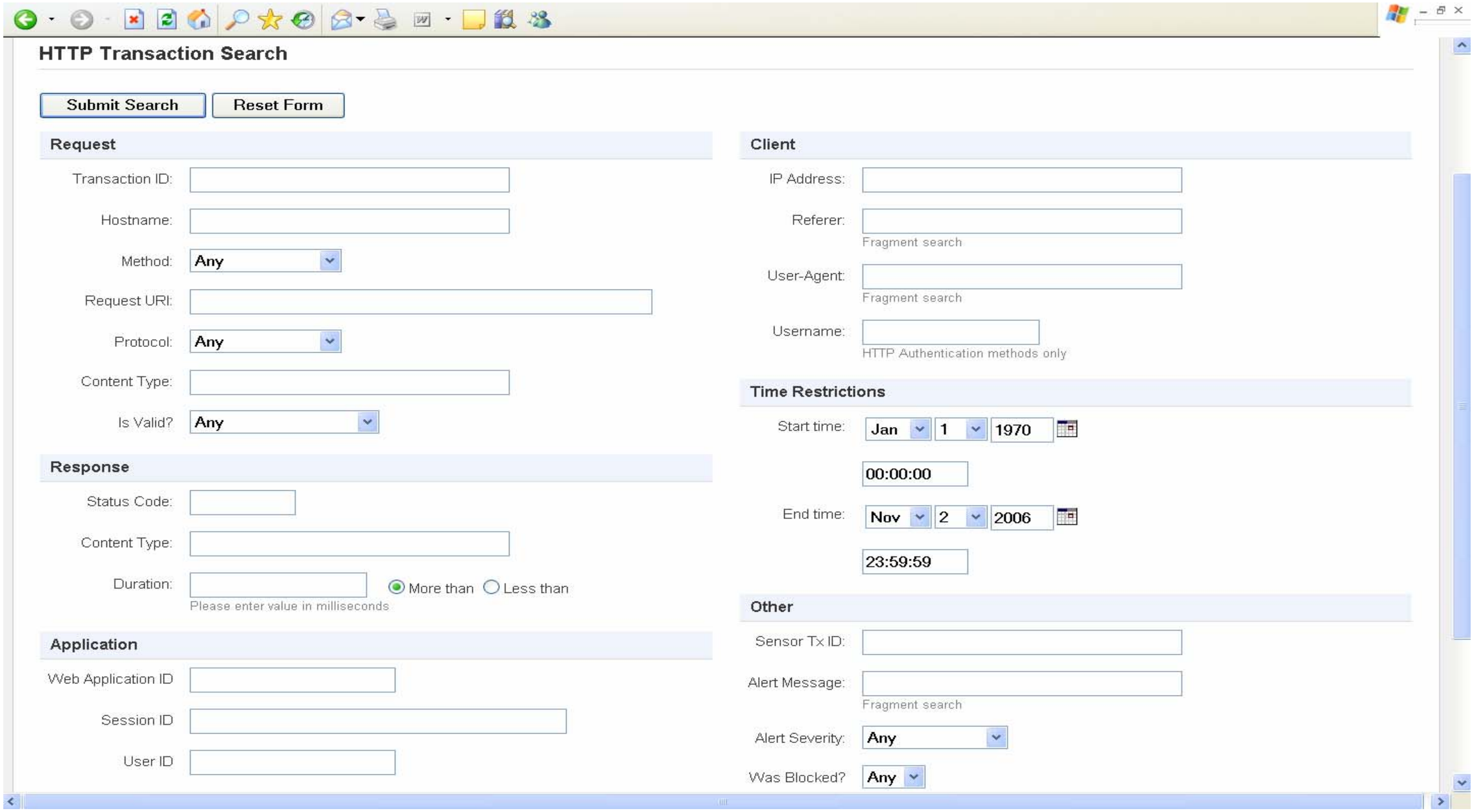
Activity Today

Activity This Week

Activity This Month

BREACH

Transaction Search Interface



HTTP Transaction Search

Request

Transaction ID:

Hostname:

Method: Any

Request URI:

Protocol: Any

Content Type:

Is Valid? Any

Response

Status Code:

Content Type:

Duration: ● More than ○ Less than

Please enter value in milliseconds

Application

Web Application ID

Session ID

User ID

Client

IP Address:

Referer:

Fragment search

User-Agent:

Fragment search

Username:

HTTP Authentication methods only

Time Restrictions

Start time: Jan 1 1970

00:00:00

End time: Nov 2 2006

23:59:59

Other

Sensor Tx ID:

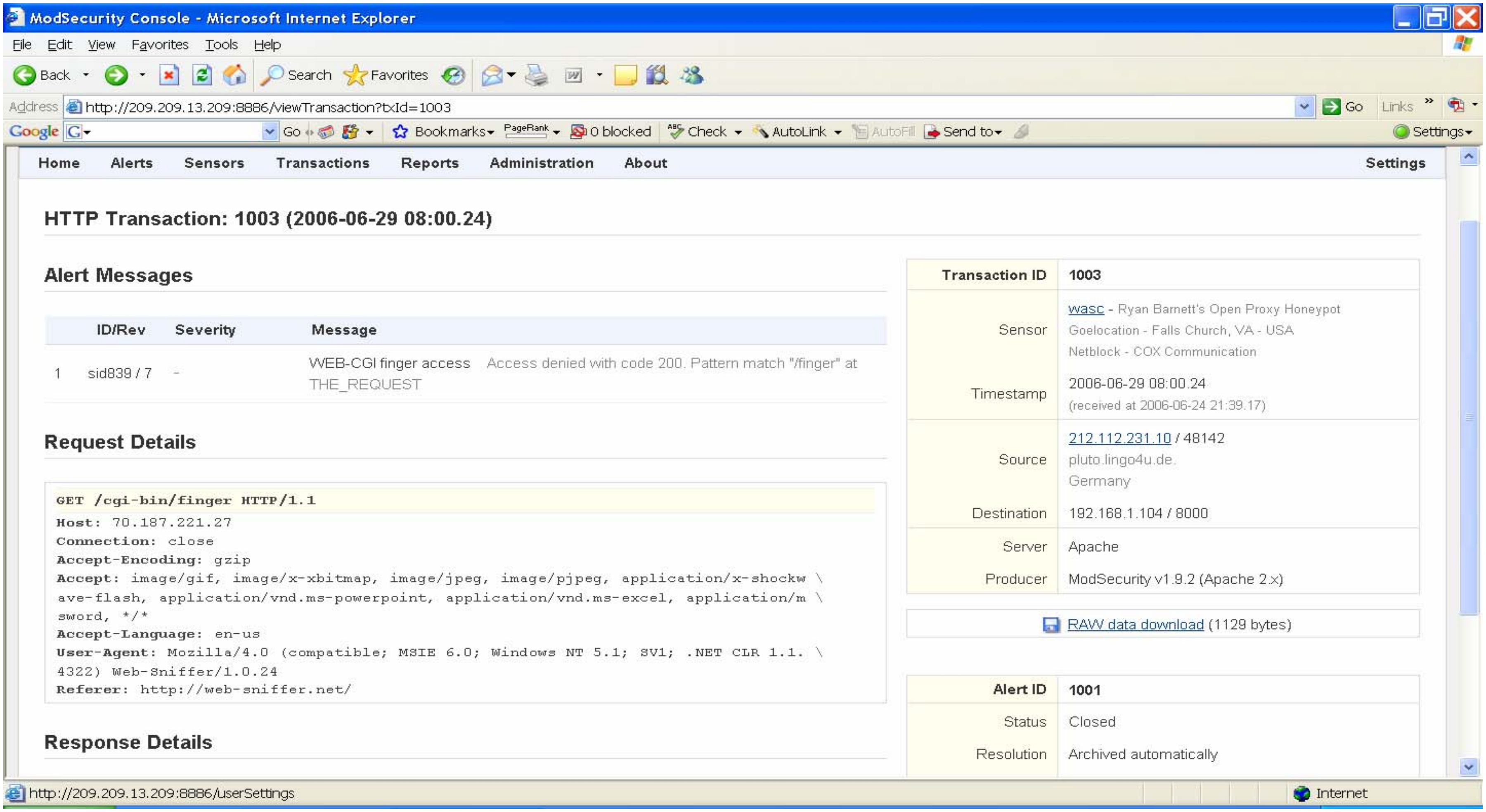
Alert Message:

Fragment search

Alert Severity: Any

Was Blocked? Any

Alert Details Interface


ModSecurity Console - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Address http://209.209.13.209:8886/viewTransaction?txId=1003
Go
Links
Google
Go
Bookmarks
PageRank
0 blocked
Check
AutoLink
AutoFill
Send to
Settings
http://209.209.13.209:8886/userSettings
Internet

Home Alerts Sensors Transactions Reports Administration About Settings

HTTP Transaction: 1003 (2006-06-29 08:00.24)

Alert Messages

	ID/Rev	Severity	Message	
1	sid839 / 7	-	WEB-CGI finger access	Access denied with code 200. Pattern match "/finger" at THE_REQUEST

Request Details

```
GET /cgi-bin/finger HTTP/1.1
Host: 70.187.221.27
Connection: close
Accept-Encoding: gzip
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/x-shockw \
ave-flash, application/vnd.ms-powerpoint, application/vnd.ms-excel, application/m \
sword, */*
Accept-Language: en-us
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 1.1. \
4322) Web-Sniffer/1.0.24
Referer: http://web-sniffer.net/
```

Response Details

Transaction ID	**1003**
Sensor	wasc - Ryan Barnett's Open Proxy Honeypot Goelocation - Falls Church, VA - USA Netblock - COX Communication
Timestamp	2006-06-29 08:00.24 (received at 2006-06-24 21:39.17)
Source	212.112.231.10 / 48142 pluto.lingo4u.de. Germany
Destination	192.168.1.104 / 8000
Server	Apache
Producer	ModSecurity v1.9.2 (Apache 2.x)

RAW data download (1129 bytes)

Alert ID	**1001**
Status	Closed
Resolution	Archived automatically

BREACH

Wrap-Up

- Hope this information was useful.
- Let’s initiate Q&A on the modsecurity-user-mail list.
- Let me know what other Webcast Topics you would like to see.
- Please contact me directly with any commercial support questions - Ryan.Barnett@breach.com
- Thanks for your time.