"The Core Rule Set": Generic detection of application layer attacks

Ofer Shezaf, OWASP Israel Chapter Leader
Chief Technology Officer, Breach Security
ofers@breach.com
+972-54-4431119

6th OWASP AppSec Conference
Milan - May 2007

The OWASP Foundation
http://www.owasp.org/

About The Speaker

- Community Participation:
  - ModSecurity Core Rule Set Project Leader
  - OWASP Israeli chapter leader
  - Web Application Security Consortium (WASC) Board Member
  - WASC Web Hacking Incidents Database Project Leader
- Day Job:
  - CTO, Breach Security
  - In charge of security research, rules and signatures.

2nd OWASP IL mini conference at the Interdisciplinary Center (IDC) Herzliya, May 21th 2007 [edit]

Following the big success of the 1st one, we are glad to announce the 2nd OWASP IL mini conference at the Interdisciplinary Center (IDC) Herzliya . The mini conference is a non-commercial event focusing on web application security. As you can see in the program below, we have carefully selected the presentations and we hope they are all relevant, informative and most importantly, none commercial. Never the less, we are happy to say that we were able to get very distinguish companies to sponsor the event and make sure that the refreshments would be great.

The meeting will be held on Monday, May 21st, Starting at 13:30 at the Interdisciplinary Center (IDC) Herzliya campus. Participation is free and open to all, but please inform us (e-mail to ofers@breach.com) that you are coming as space is limited. Feel free to spread the word about this meeting to anyone you feel would be interested. You can also register to get the OWASP Israel mailing list and receive updates regarding chapter's meetings. For further details please contact us.

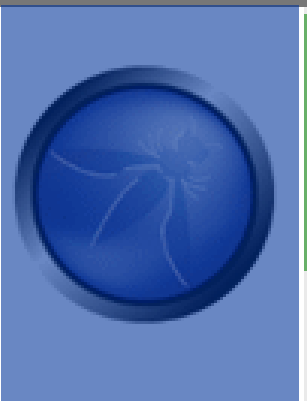
Dr. Anat Bremler-Barr
Program Academic Director, Information Security Program
Efi Arazi School of Computer Science, Interdisciplinary Center (IDC) Herzliya

Ofer Shezaf
CTO, Breach Security
Chapter Leader, OWASP Israel

The meeting is sponsored by Breach Security, Checkpoint, Hacktics, Microsoft, Zend, 2Bsecure, F5 Networks and the Efi Arazi school of Computer Science at the Interdisciplinary Center (IDC) Herzliya.

What are we going to talk about

- What are Application Firewalls
- Application Firewalls vs. IDPS
- The Core Rule Set:
  - Protocol Compliance and Policy.
  - App Layer Signatures detection.
  - Odds and Ends



"The Core Rule Set":
Generic detection of application layer attacks

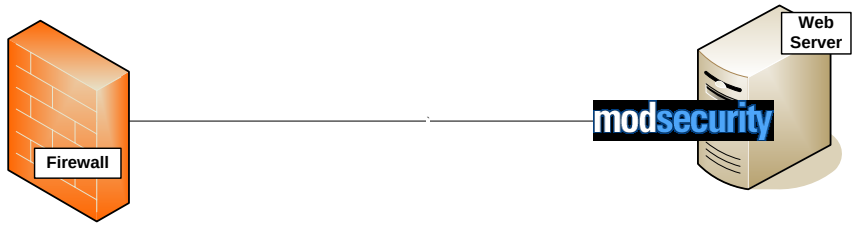
Ofer Shezaf
ofers@breach.com

About Application Firewalls

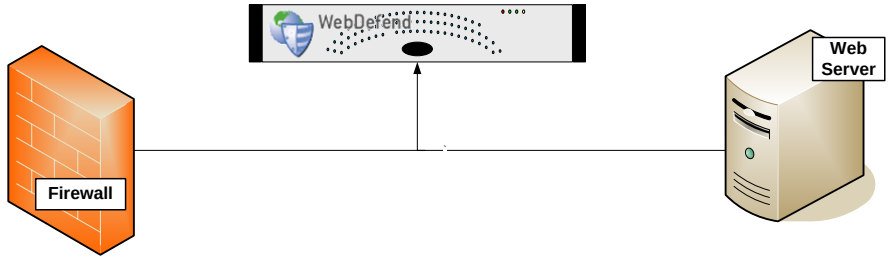
Multiple Deployment Modes

In-Line mode

Embedded mode

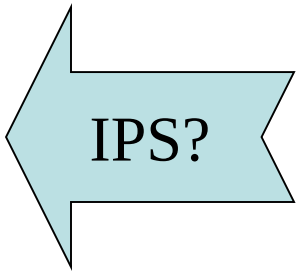


Out of line mode



2 ½ Protection Strategies

- External patching
  - Also known as "just-in-time patching" or "virtual patching".
- Positive security model
  - An independent input validation envelope.
  - Rules must be adjusted to the application.
  - Automated and continuous learning (to adjust for changes) is the key.
- Negative security model
  - Looking for bad stuff,
  - Mostly signatures based.
  - Generic but requires some tweaking for each application.



Virtual Patching

- Testing reveals that the login field is vulnerable to SQL injection.
- Login names cannot include characters beside alphanumerical characters.
- The following rule will help:

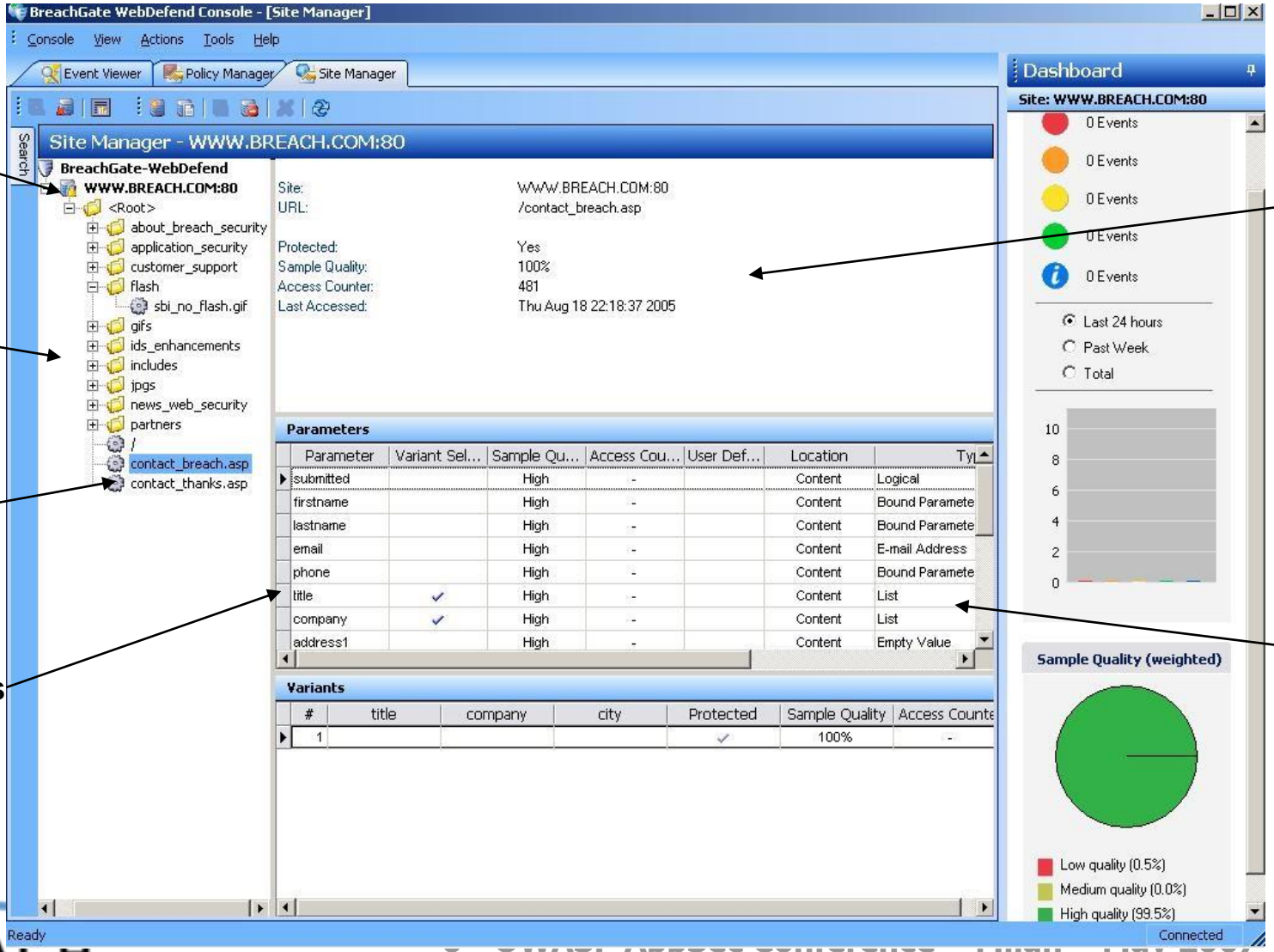
```
<LocationMatch "^/app/login.asp$">
        SecRule ARGS:username "!^\w+$" "deny,log"
</LocationMatch>
```


Positive Security

```
<LocationMatch "^/exchweb/bin/auth/owaauth.dll$">
  SecDefaultAction "log,deny,t:lowercase"
  SecRule REQUEST_METHOD !POST
  SecRule ARGS:destination "URL" "t:urlDecode"
  SecRule ARGS:flags "![0-9]{1,2}"
  SecRule ARGS:username "[0-9a-zA-Z].{256,}"
  SecRule ARGS:password ".{256,}"
  SecRule ARGS:SubmitCreds "!Log.On"
  SecRule ARGS:trusted "!(0|4)"
</LocationMatch>
```

- The same, but for every field in every application
- Very hard to create, requires learning by:
  - Monitoring outbound traffic (match input to web server request). Caveats: JavaScript, Web Services
  - Monitoring inbound traffic (normal behavior). Caveats: Statistics, attacks in learning period.

Positive Security



Site

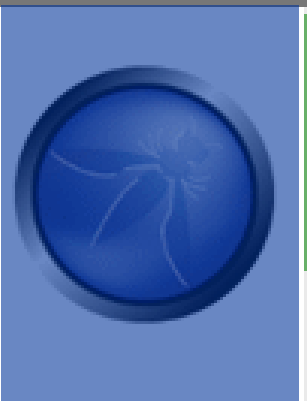
Site Map

URLs

Parameters

Site Status

Parameter Types



"The Core Rule Set":
Generic detection of application layer attacks

Ofer Shezaf
ofers@breach.com

Web Application Firewalls vs. Intrusion Prevention Systems

Negative Security: An IPS, but

- Deep understanding of HTTP and HTML
  - Breaking up to individual fields: headers, parameters, uploaded files.
  - Validation of field attributes such as content, length or count
  - Correct breakup and matching of transactions and sessions.
  - Compensation for protocol caveats and anomalies, for example cookies.
- Robust parsing:
  - Unique parameters syntax
  - XML requests (SOAP, Web Services)
- Anti Evasion features:
  - Decoding
  - Path canonizations
  - Thorough understanding of application layer issues: Apache request line delimiters, PHP parameter names anomalies.
- Rules instead of signatures:
  - Sessions & state management, Logical operators, Control structures.

IDPS signatures vs. WAF Rules

Signatures:

- Simple text strings or regular expression patterns matched against input data.
- Usually detect attack vectors for known vulnerabilities, while web applications are usually custom made.
- Variations on attack vectors are very easy to create

Rules:

- Multiple operators and logical expressions: Is password field length > 8?
- Selectable anti-evasion transformation functions.
- Control structures such as IF:
  - Apply different rules based on transactions.
- Variables, Session & state management:
  - Aggregate events over a sessions.
  - Detect brute force & denial of service.
  - Audit user name for each transaction

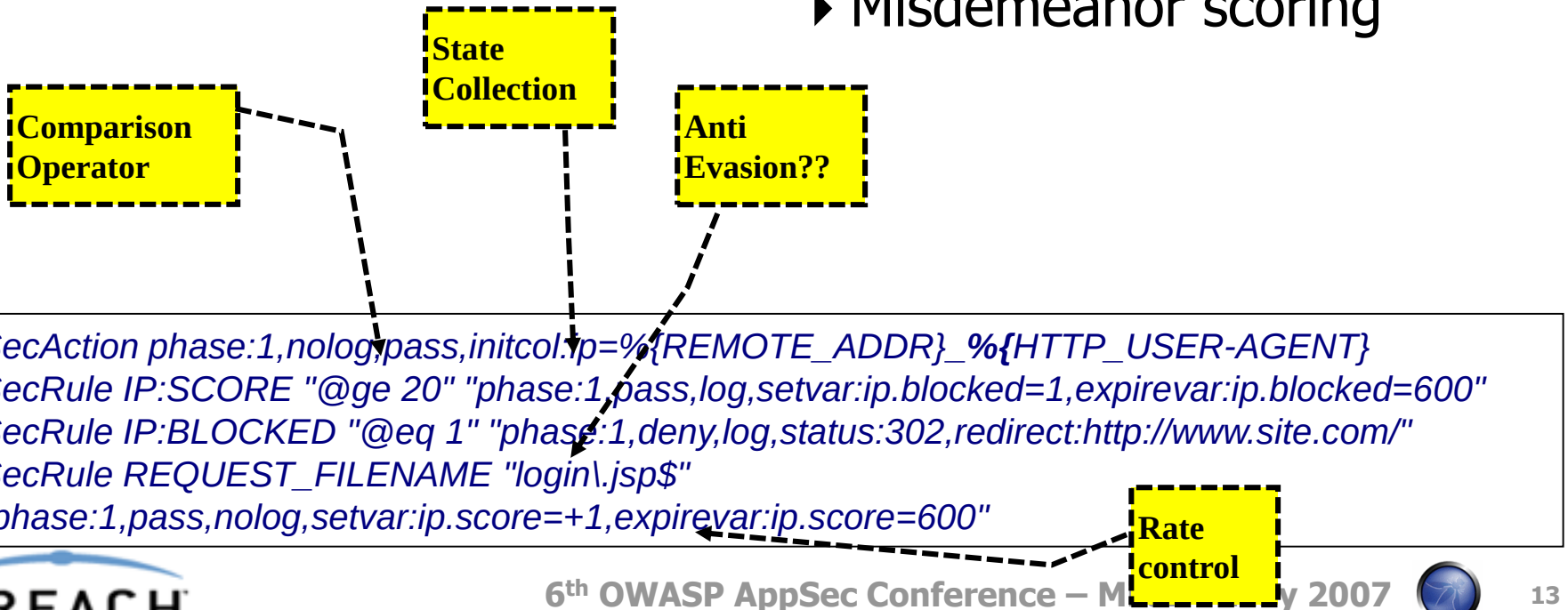
Some Complex Rules:

Monitoring:

- Capturing the user name
- Login failures

Protection

- Brute force detection
- Scanners and automation detection
- Misdemeanor scoring



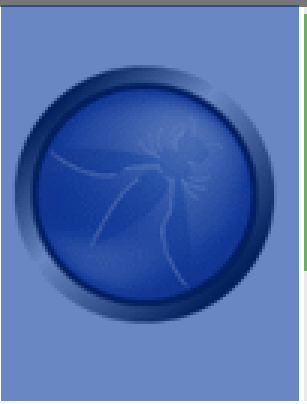
Ofer Shezaf
ofers@breach.com

The Core Rule Set

```
modsecurity-core-rules_2.0-1.1.1 (blocking).zip
modsecurity_crs_10_config.conf
modsecurity_crs_20_protocol_violations.conf
modsecurity_crs_30_http_policy.conf
modsecurity_crs_35_bad_robots.conf
modsecurity_crs_40_generic_attacks.conf
modsecurity_crs_45_trojans.conf
modsecurity_crs_50_outbound.conf
modsecurity_crs_55_marketing.conf
```

Generic detection of app layer attacks

- Core Rule Set available for ModSecurity at:
  - http://www.modsecurity.org/projects/rules/index.html
  - Probably translatable to any App Firewall
- Benefits from ModSecurity features:
  - Anti Evasion
  - Granular Parsing
- Detection Mechanisms:
  - Protocol Validation
  - Generic Attack Signatures
  - Known Vulnerabilities Signatures
  - More…



"The Core Rule Set":
Generic detection of application layer attacks

Ofer Shezaf
ofers@breach.com

Protocol Validation

Protocol Violations

- Protocol vulnerabilities such as Response Splitting, Request Smuggling, Premature URL ending:
  - Content length only for none GET/HEAD methods
  - Non ASCII characters or encoding in headers.
  - Valid use of headers (for example, content length is numerical)
  - Proxy Access
- Attack requests are different due to automation:
  - Missing headers such as Host, Accept, User-Agent.
  - Host is an IP address.

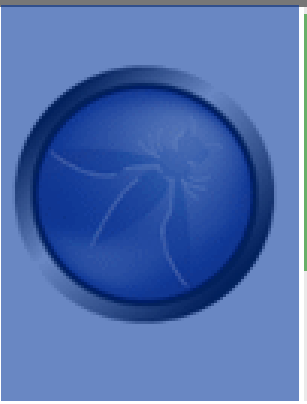
Case study: Full Width Unicode Evasion

- CERT VU#739224, May 14th 2007
  - http://www.kb.cert.org/vuls/id/739224
- Two levels of a solution:
  - ModSecurity: decode right
    - Expected shortly
  - Core Rule Set: block full/half width Unicode
    - ready, available online tommorow

```
SecRule
REQUEST_FILENAME|ARGS|ARGS_NAMES|REQUEST_HEADERS|XML:/*|!RE
QUEST_HEADERS:Referer "\%u[fF]{2}[0-9a-fA-F]{2}" \
  "t:none,deny,log,auditlog,status:400,msg:'Unicode Full/Half Width Abuse Attack
Attempt',,id:'950116',severity:'4'"
```

Protocol Policy

- Policy is usually application specific:
  - Some restrictions can usually be applied generically.
  - White lists can be build for specific environments.
- Items that can be allowed or restricted:
  - Methods - Allow or restrict WebDAV, block abused methods such as CONNECT, TRACE or DEBUG.
  - File extensions – backup files, database files, ini files.
  - Content-Types (and to some extent other headers)
- Limitations on sizes:
  - Request size, Upload size,
  - # of parameters, length of parameter.



"The Core Rule Set":
Generic detection of application layer attacks

Ofer Shezaf
ofers@breach.com

Application Layer Signatures

Case study: 1=1

- Classic example of an SQL injection attacks. Often used as a signature.
- But, can be avoided easily using:
  - Encoding: 1%3D1
  - White Space: 1 =%091
  - Comments 1 /* This is a comment */ = 1
- Actually not required at all by attacker.
  - Any true expression would work: 2 > 1
  - In some cases, a constant would also work. In MS-Access all the following are true: 1, “1”, “a89”, 4-4.
- No simple generic detection

Generic application layer signatures

- Detect attack indicators and not attack vectors:
  - xp_cmdshell,
  - "<", single quote - Single quote is very much needed to type *O'Brien*
  - *select*, *union – which are English words*
- *Aggregate indicators to determine an attack:*
  - Very strong indicators: xp_cmdshell, varchar,
  - Sequence: *union …. select, select … top … 1*
  - Amount**:** *script, cookie* and *document* appear in the same input field.
  - Sequence over multiple requests from the same source.

Snort signature for Bugtraq vulnerability #21799

Exploit:

```
/cacti/cmd.php?1+1111)/**/UNION/**/SELECT/**/2,0,1,1,127
.0.0.1,null,1,null,null,161,500, proc,null,1,300,0, ls -
la > ./rra/suntzu.log,null,null/**/FROM/**/host/*+11111
```

Snort Signature:

Does the application accepts POST requests?

Signature built for specific exploit

```
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS
(
  msg:"BLEEDING-EDGE WEB Cacti cmd.php Remote Arbitrary
  SQL Command Execution Attempt";
  flow:to_server,established;
  uricontent:"/cmd.php?"; nocase;
  uricontent:"UNION"; nocase;
  uricontent:"SELECT"; nocase;
  reference:cve,CVE-2006-6799; reference:bugtraq,21799;
  classtype: web-application-attack; sid:2003334; rev:1;
```

An SQL injection does not have to use SELECT or UNION

UNION and SELECT are common English words. So is SELECTION

Back to Bugtraq vulnerability #21799

The Core Rule Set Generic Detection

Supports any type of parameters, POST , GET or any other

```
SecRule REQUEST_FILENAME|ARGS|ARGS_NAMES|
REQUEST_HEADERS|!REQUEST_HEADERS:Referer \

        "(?:\b(?:(?:s(?:elect\b(?:.{1,100}?\b(?:(?:length|count|top)\b.{1,100
}?\bfrom|from\b.{1,100}?\bwhere)|.*?\b(?:d(?:ump\b.*\bfrom|ata_type)|(?:
to_(?:numbe|cha)|inst)r))|p_(?:(?:addextendedpro|sqlexe)c|(?:oacreat|prep
ar)e|execute(?:sql)?|makewebtask)|ql_(?:… … … \

        “capture,log,deny,t:replaceComments, t:urlDecodeUni,
t:htmlEntityDecode, t:lowercase,msg:'SQL Injection Attack. Matched
signature <%{TX.0}>',id:'950001',severity:'2'“
```

Every SQL injection related keyword is checked

Common evasion techniques are mitigated

SQL comments are compensated for

Back to Bugtraq vulnerability #21799

Virtual Patching

```
<LocationMatch :"/cmd.php$">
        SecRule QUERY_STRING "^[\d\s]*$″ ‶deny,log"
</LocationMatch>
```

Parameters Must Be Numeric

Or

```
SecRule REQUEST_FILENAME :"/cmd.php$" ‶deny,log"
```

Actually script should not be run remotely

Simpler, isn't it?

Odds and Ends

6th OWASP AppSec Conference

Milan - May 2007

Malicious Robots

- Detection of malicious robots:
  - Unique request attributes: User-Agent header, URL, Headers
  - Black list of IP addresses
  - Rate based detection
- Not aimed against targeted attacks, but against general malicious internet activity:
  - Offloads a lot of cyberspace junk & noise
  - Effective against comment spam.
  - Reduce event count.
- In addition:
  - Detection of security scanners
  - Detection of non malicious robots (such as search engines).
  - Confusing security testing software (HTTPrint)

Trojans and Viruses

- Major problem at hosting environments
  - Uploading is allowed.
  - Some sites may be secure while others not.
- Generic detection:
  - Check upload of Viruses.
  - Check upload of Trojans:
    - AV software is not very good at that.
  - Check for access to Trojans:
    - Known signatures (x_key header)
    - Generic file management output (gid, uid, drwx, c:\)

Error conditions

- Last line of defense if all else fails
- Provide feedback to application developers
- Important for customer experience
- Makes life for the hacker harder

Thank You!

Ofer Shezaf
ofers@breach.com

6th OWASP AppSec Conference

Milan - May 2007

The OWASP Foundation

http://www.owasp.org/