



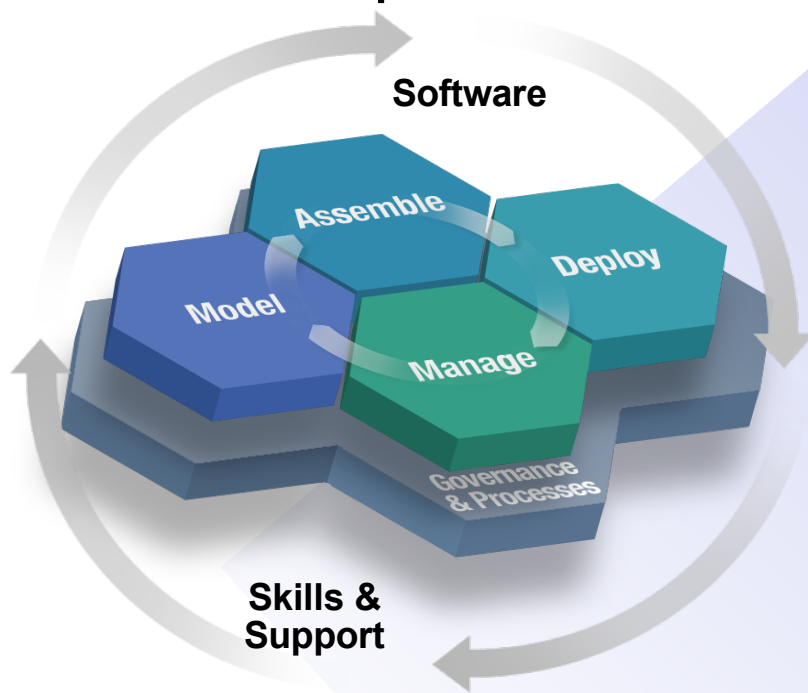
IBM SOA

DataPower SOA Appliances Feature Overview

Presented by Alasdair Nottingham



IBM's acquisition of DataPower



An SOA Appliance...



Creating customer value through extreme SOA performance and security

- **Simplifies** SOA with specialized devices
- **Accelerates** SOA with faster XML throughput
- **Helps secure** SOA XML implementations

WebSphere DataPower SOA Appliances redefine the boundaries of middleware extending the SOA Foundation with **specialized, consumable, dedicated SOA appliances** that combine **superior performance and hardened security** for SOA implementations.

Why an Appliance for SOA

- **Hardened, specialized hardware for helping to integrate, secure & accelerate SOA**
- **Many functions integrated into a single device:**
 - Impact: connectivity will require service level management, routing, policy, transformation
- **Higher levels of security assurance certifications require hardware:**
 - Example: government FIPS Level 3 HSM, Common Criteria
- **Higher performance with hardware acceleration:**
 - Impact: ability to perform more security checks without slow downs
- **Addresses the divergent needs of different groups:**
 - Example: enterprise architects, network operations, security operations, identity management, web services developers
- **Simplified deployment and ongoing management:**
 - Impact: reduces need for in-house SOA skills & accelerates time to SOA benefits

Where to Enforce XML Web service Security

- First level of defense: DataPower XML Security Gateway
 - Performance – at least 10X improvement over software (XML DSig example: 10 tps in software, 1000 tps in hardware)
 - Scalability – Minimize the number of servers
 - Manageability – Fewer enforcement points simplify configuration
 - Simplicity – No need to change applications
 - Security – Removing security from application
 - Availability – XDoS checking protects application and Web servers
 - Interoperability – Translates across multiple transports and standards
 - Monitoring – Simplified audit logging at enforcement point
- Second level of defense: Web Services Application
 - Manageability – Integrate with container-based security
 - Security – Business-specific security embedded within application

Hardware Device for Improved Security



- **Sealed network-resident device:**
 - Optimized hardware, crypto in hardware, firmware, embedded OS
 - Secured by default
 - Hardened, Tamper proof case with intrusion detection
 - Initial access by serial cable only
 - No services or interfaces exposed without administrator's enablement
 - Single signed/encrypted firmware upgrade only, not arbitrary software
 - High assurance, "default off" locked-down configuration
 - Security vulnerabilities minimized (few 3 party components)
 - Hardware storage of encryption keys, locked audit log
 - No drives/USB ports, tamper-proof case
- **Third party certification:**
 - FIPS 140-2 level 3 HSM (option)
 - Under evaluation by Common Criteria EAL4
- **Large financial and government customers**

"The DataPower [XS40]... is the most hardened ... it looks and feels like a datacenter appliance, with no extra ports or buttons exposed and no rotating media. "

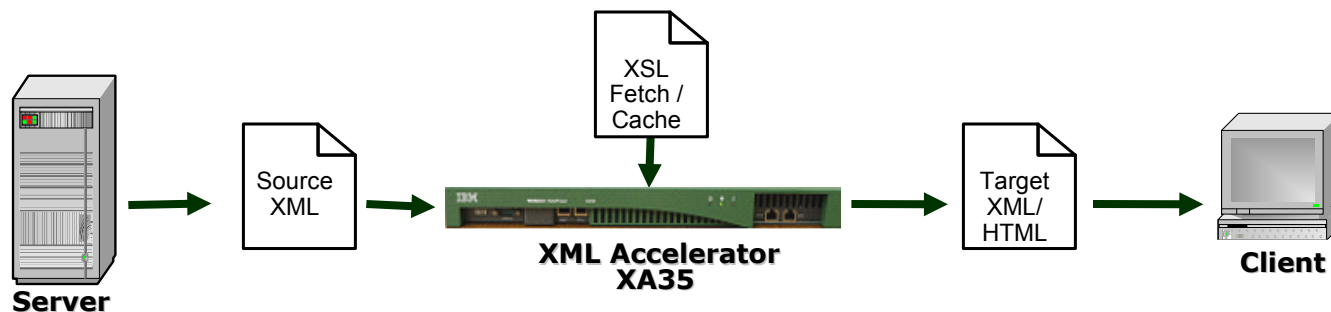
- InfoWorld

High-speed XML Processing

XSLT Transformation, XPath Processing

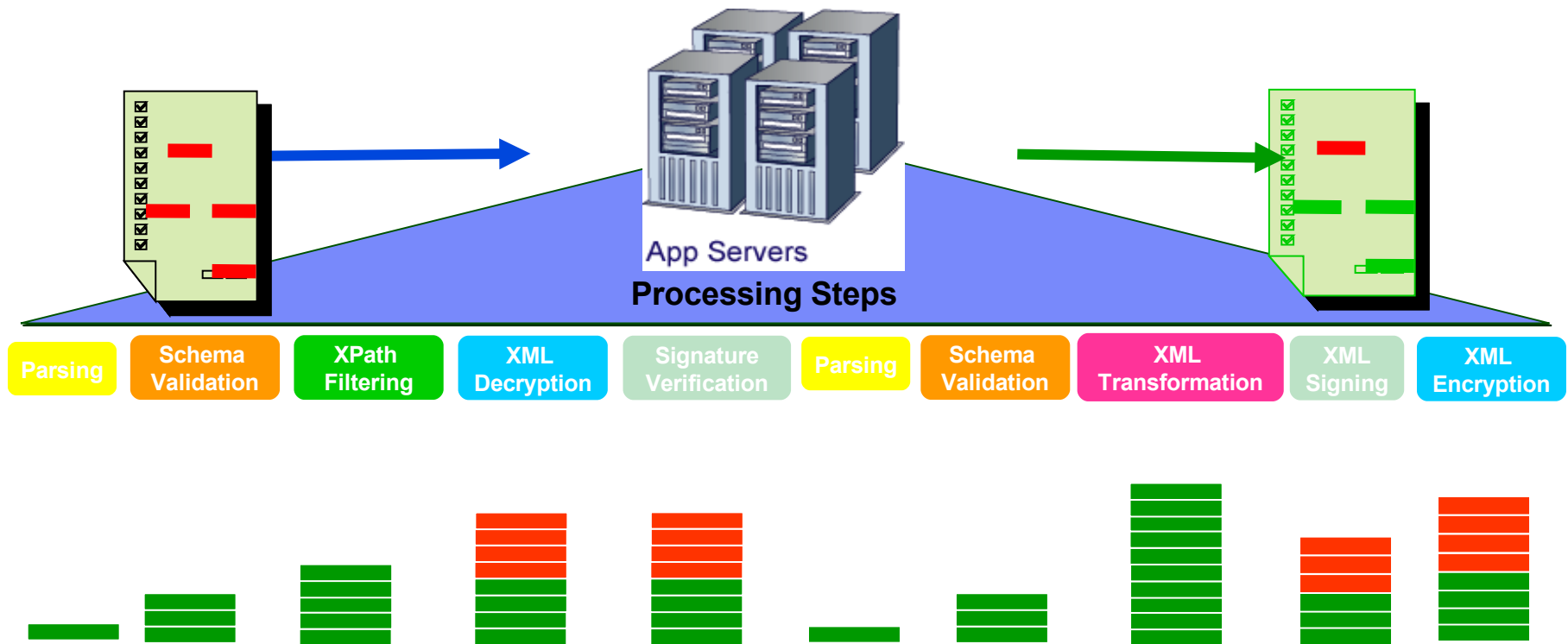


- DataPower's purpose-built message processing engine delivers “wirespeed” performance for both XML to XML and XML to HTML transformations with increased throughput and decreased latency
- Combines the XML processing power of multiple servers in a single network device, off-loading heavy lifting from general purpose servers
- Shares standardized capacity across a number of applications, helping rationalize application infrastructure with a single, standards-compliant, shared XML processing infrastructure



* Performance varies depending on usage and customer scenarios, for example 100-200 Mbps

XML WS Security is XML Processing



- Performance is key to security
 - Each security function requires XML processing
 - Must implement all services without any compromise
 - Need ability to scale as content and user base grows

Take another look back at that chart

- The green bars represent the amount of XML processing at each stage
- The red bars represent cryptographic work required
 - See how little there actually is
 - Even if you use hardware cryptographic accelerators, cryptography is only a tiny part of the actual processing required
- Now the scary part
 - To process WS-Security, you **have** to do each of the stages describe on the previous chart
 - Take a look at how much XML processing you need to do **before** you ever get to actually parsing the incoming message!

IBM DataPower Appliance Platform

Specialized network devices simplify, help secure & accelerate SOA

XML Accelerator XA35



- Accelerates XML processing and transformation
- Increases throughput and reduces latency
- Lowers development costs

XML Security Gateway XS40



- Help secure SOA with XML threat protection and access control
- Combines Web services security, routing and management functions
- Drop-in, centralized policy enforcement
- Easily integrates with exiting infrastructure and processes

Integration Appliance XI50



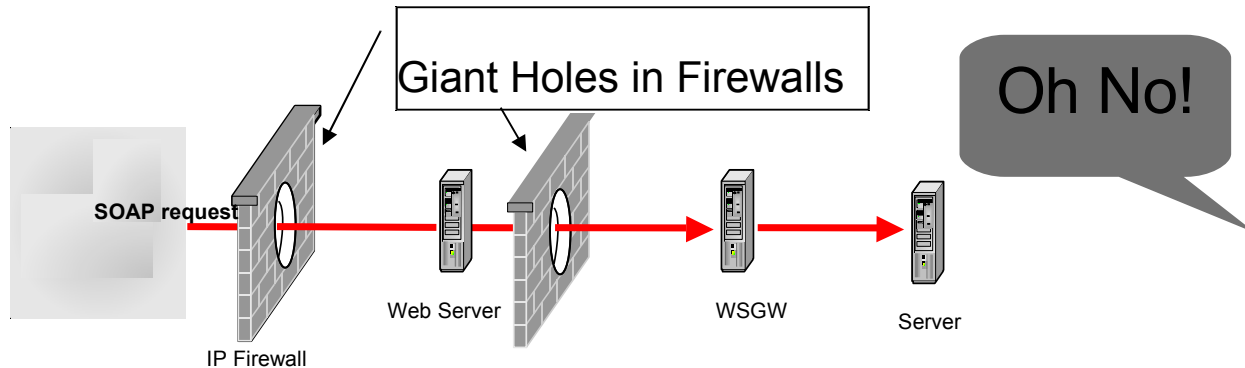
- Transforms messages (Binary to XML, Binary to Binary, XML to Binary)
- Bridges multiple protocols (e.g. MQ, HTTP, JMS)
- Routes messages based on content and policy
- Integrates message-level security and policy functions

XML Threats - The New Frontier

- Traditional targets are no longer as interesting
 - Microsoft has finally gotten serious about security
 - In the hacker world, Windows and other traditional attacks are “oh so yesterday” – been there done that
 - J2EE application servers are no longer the raw, open targets they once were
 - Just look at the improvement in J2EE/WAS security v3.5 to v6.1
- New technologies are always fertile ground for hackers
 - Interesting and immature new platforms
 - Plenty of bugs to exploit
 - Plenty of ‘default’ or non-hardened installations due to lack of training/experience
 - Lots of new APIs, protocols, attack points to exploit
 - SOA architectures built on Web services, SOAP and XML are all the rage
 - Web 2.0, AJAX anyone?
- **Prediction:** An explosion of exploited vulnerabilities related to Web Services/SOA
 - Problem is, they are kept quiet as companies are reluctant to make them public
 - Bad for stockholder/customer confidence!

The New Frontier

- Web Services are based on SOAP/XML/HTTP, very firewall friendly



- Ah....see anything wrong with that picture?
(hint – you bought those firewalls for a reason...)
- These new systems have invited a whole new class of attacks, which we label “XML Threats”
 - And guess what – they pass right on through those friendly firewalls...
- As we will see during the course of this presentation, DataPower provides for a much simpler, more hardened, faster, less expensive solution to the above

XML Threats

Security Risks Growing

- XML Entity Expansion and Recursion Attacks
- XML Document Size Attacks
- XML Document Width Attacks
- XML Document Depth Attacks
- XML Wellformedness-based Parser Attacks
- Jumbo Payloads
- Recursive Elements
- MegaTags – aka Jumbo Tag Names
- Public Key DoS
- XML Flood
- Resource Hijack
- Dictionary Attack
- Message Tampering
- Data Tampering
- Message Snooping
- XPath Injection
- SQL injection
- WSDL Enumeration
- Routing Detour
- Schema Poisoning
- Malicious Morphing
- Malicious Include – also called XML External Entity (XXE) Attack
- Memory Space Breach
- XML Encapsulation
- XML Virus
- Falsified Message
- Replay Attack
- ...others

Types of XML Attacks

- Four broad classifications –
 - **XML Denial of Service (xDOS)** – Slowing down or disabling a Web Service so that valid service requests are hampered or denied
 - **Unauthorized Access** – Gaining unauthorized access to a Web Service or its data
 - **Data Integrity/Confidentiality** - Attacks that strike at data integrity of Web Service responses, requests or underlying databases
 - **System Compromise** – Corrupting the Web Service itself or the servers that host it, attacks that gain control of your systems
- These can be facilitated by tricky/complex XML, virus-laden XML/SOAP attachments, etc

Example Attack – Billion Laughs

- **Coercive parsing/recursive element example**

```
<?xml version="1.0"?>
  <!DOCTYPE billion [
    <!ELEMENT billion (#PCDATA)>
    <!ENTITY laugh0 "ha! ">
    <!ENTITY laugh1 "&laugh0;&laugh0;">
    <!ENTITY laugh2 "&laugh1;&laugh1;">
    ...
    <!ENTITY laugh127 "&laugh126;&laugh125;">
  ]> <billion>&laugh127;</billion>
```

- A completely valid, well-formed XML document
- When submitted to parser, quickly exhausts memory/CPU

```
<billion>
ha! ha! ha! ha! ha! ha! ha! ha! ha! ha! ha! ha! ha! ha! ha! ha! ha! ha! ha! ha! ha!
ha! ha! ha! ha! ha! ha! ha! ha! ha! ha! ha! ha! ha! ha! ha! ha! ha! ha! ha! ha! ha!
ha! ha! ha! ha! ha! ha! ha! ha! ha! ha! ha! ha! ha! ha! ha! ha! ha! ha! ha! ha! ha!
ha! ha! ha! ha! ha! ha! ha! ha! ha! ha! ha! ha! ha! ha! ha! ha! ha! ha! ha! ha! ha!
ha! ha! ha! ha! ha! ha! ha! ha! ha! ha! ha! ha! ha! ha! ha! ha! ha! ha! ha! ha! ha!
.....
</billion>
```


Traditional Systems and Exposure

- Traditional systems don't offer much protection
 - XML validation is typically 'off' for performance reasons
 - Parsers such as WAS/Web Services Gateway don't even look for these types of attacks, or rate-limit traffic
 - By the time they try to parse the message, it's too late
- Traditional architectures, such as that in our earlier topology diagram, allow the requests to flow to the backend, where much harm can be done
 - Can't put Java processes such as WSGW or app server in DMZ, unsafe
 - Traditional Edge devices aren't smart enough to check for these attacks
 - Hackers take advantage of "firewall-safe" SOAP/XML

XML Attack Example – Step 1 <prepare attack...>

- Very simple attack using a single SOAP message. This message is well formed, and can be sent to any Web service:

```
<S:Envelope xmlns:S='http://schemas.xmlsoap.org'>
<S:Body xmlns='http://example.com/'>
<X
  xmlns:X1='http://www.example.com/x1'
  xmlns:X2='http://www.example.com/x2'
.....
  xmlns:X9998='http://www.example.com/x9998'
  xmlns:X9999='http://www.example.com/x9999'
> </X>
</S:Body>
</S:Envelope>
```

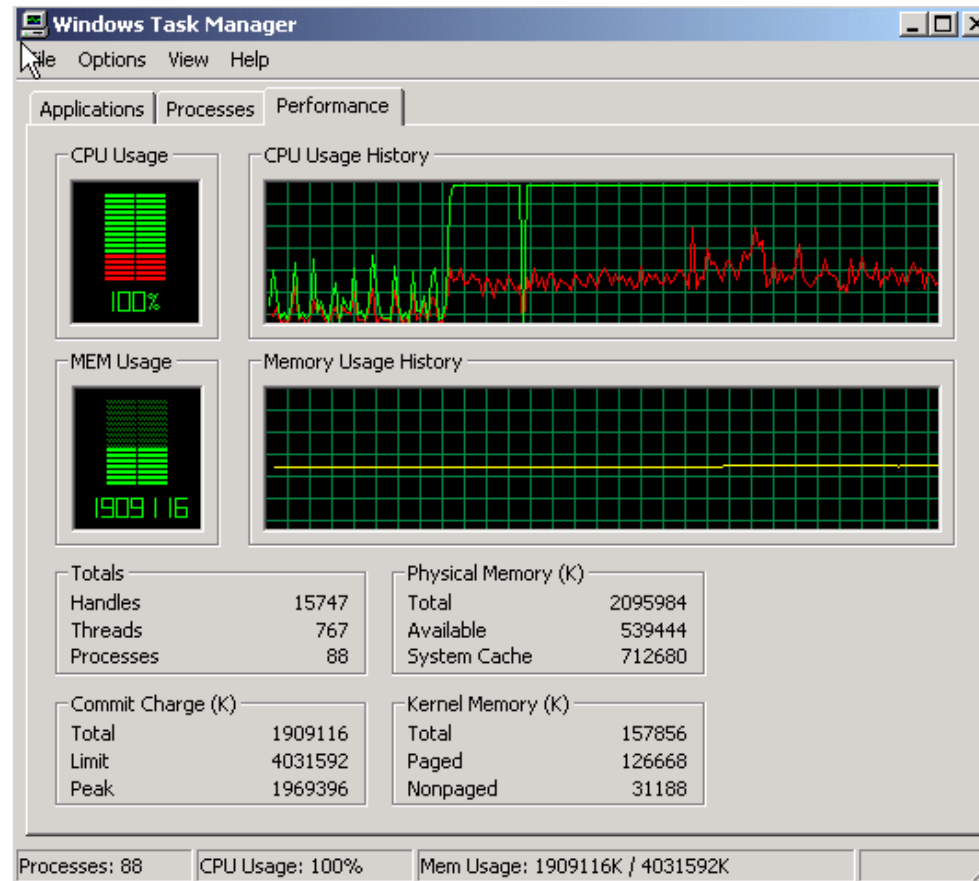
- The message was generated by a small Java program that kindly asks if you want mega elements, namespaces, attributes, etc and builds you the malicious SOAP message for you
 - Note the many namespace declarations
 - Hence, XML attacks do not take a high degree of skill or programming

XML Attack Example – Step 1 <commence attack...>

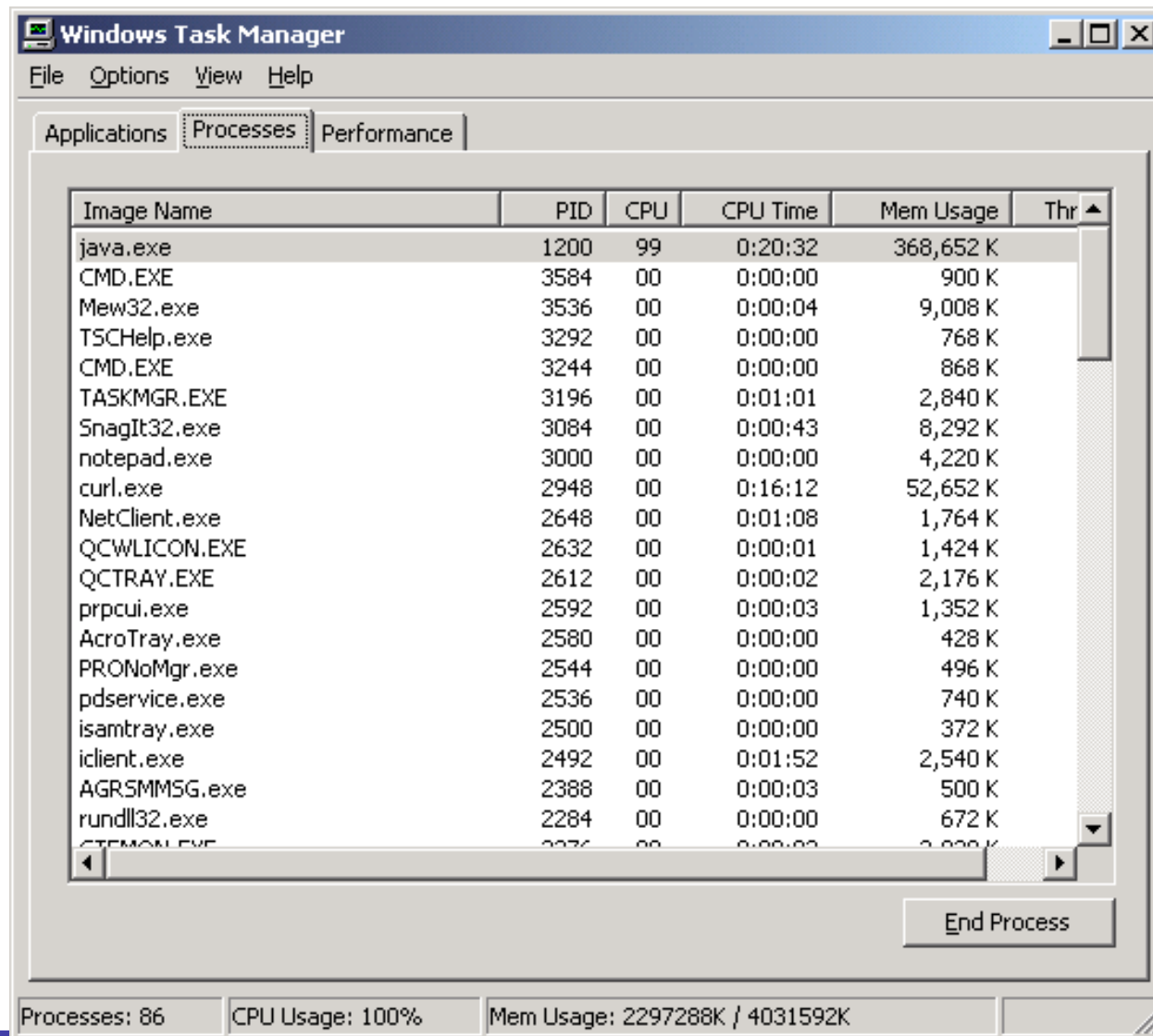
- Take one bog standard J2EE Application Server
 - For marketing reasons the Application Server vendor will remain nameless
 - But bear in mind that this will be true for **any** application server
 - Indeed ISSW have even tested with C based XML parsers with similar results
- A simple Web service was created
 - The service was **very** simple. It took two numbers, added them together and returned the sum.
 - No shortcuts, such as type="xs:any", were used. Despite the fact that many customers make this mistake.
 - In fact, the Web service was written by our very own Keys Botzum. And he was given a brief to write it as sensibly and safely as possible.
- The Web service was deployed to the application server
- The malicious message was sent to the application server

XML Attack Example – Step 2 <the damage begins...>

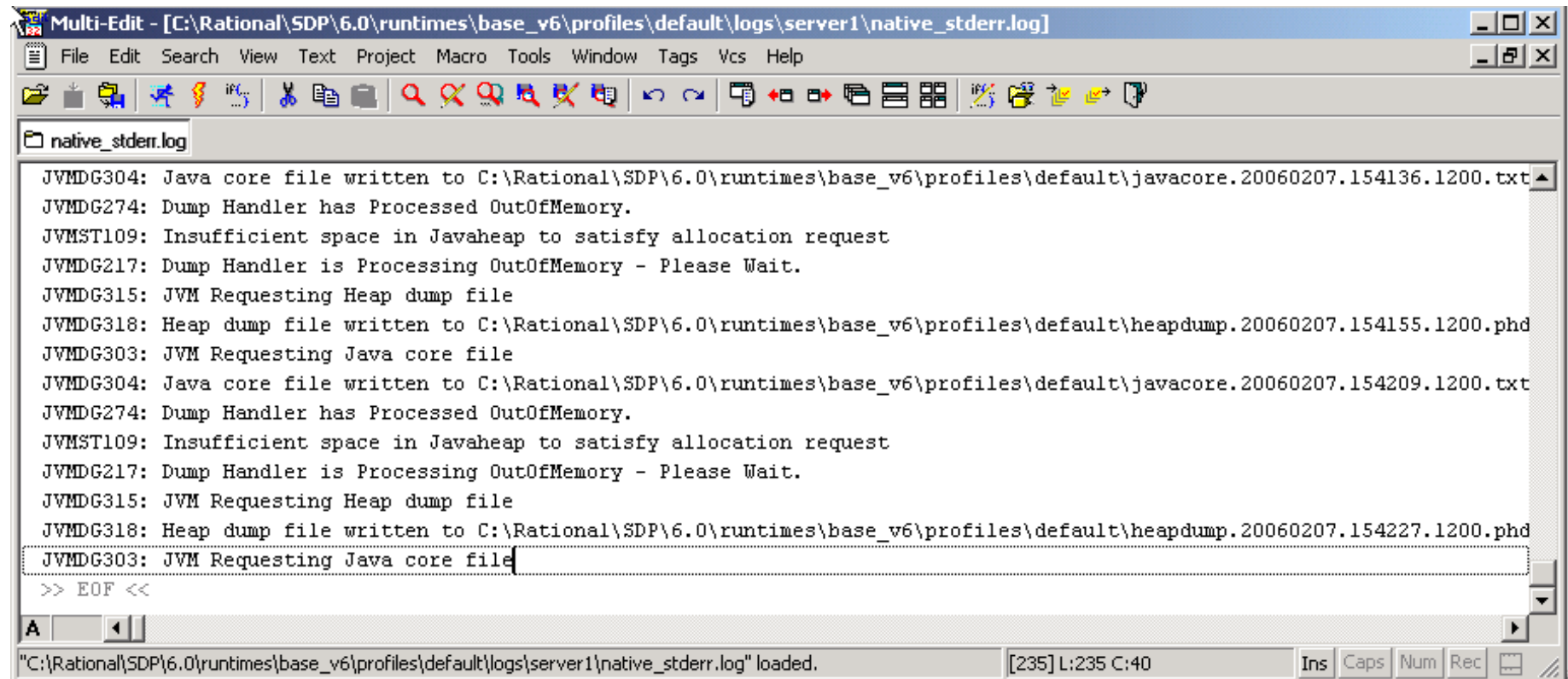
- When the SOAP message is sent, the fun begins...



XML Attack Example – Step 2 <the damage continues...>



XML Attack Example – Step 3 <crash...>



The screenshot shows a 'Multi-Edit' window with the title bar '[C:\Rational\SDP\6.0\runtimes\base_v6\profiles\default\logs\server1\native_stderr.log]'. The window contains a text editor with the following log entries:

```
JVMDG304: Java core file written to C:\Rational\SDP\6.0\runtimes\base_v6\profiles\default\javacore.20060207.154136.1200.txt
JVMDG274: Dump Handler has Processed OutOfMemory.
JVMST109: Insufficient space in Javaheap to satisfy allocation request
JVMDG217: Dump Handler is Processing OutOfMemory - Please Wait.
JVMDG315: JVM Requesting Heap dump file
JVMDG318: Heap dump file written to C:\Rational\SDP\6.0\runtimes\base_v6\profiles\default\heapdump.20060207.154155.1200.phd
JVMDG303: JVM Requesting Java core file
JVMDG304: Java core file written to C:\Rational\SDP\6.0\runtimes\base_v6\profiles\default\javacore.20060207.154209.1200.txt
JVMDG274: Dump Handler has Processed OutOfMemory.
JVMST109: Insufficient space in Javaheap to satisfy allocation request
JVMDG217: Dump Handler is Processing OutOfMemory - Please Wait.
JVMDG315: JVM Requesting Heap dump file
JVMDG318: Heap dump file written to C:\Rational\SDP\6.0\runtimes\base_v6\profiles\default\heapdump.20060207.154227.1200.phd
JVMDG303: JVM Requesting Java core file
>> EOF <<
```

The status bar at the bottom indicates the file path: "C:\Rational\SDP\6.0\runtimes\base_v6\profiles\default\logs\server1\native_stderr.log" loaded. The cursor is at [235] L:235 C:40. The status bar also includes buttons for 'Ins', 'Caps', 'Num', and 'Rec'.

XML Attack Example – Post Mortem

- App server quickly went into a death spiral as the message was attempted to be parsed
 - CPU pegged
 - Memory spiked to JVM max heap
 - After some long time of unresponsiveness, crash with javacore
- A staggering situation
 - A single user can send a simple, legit SOAP message and bring down an entire server after consuming enormous amounts of resources on the machine



Oh No!



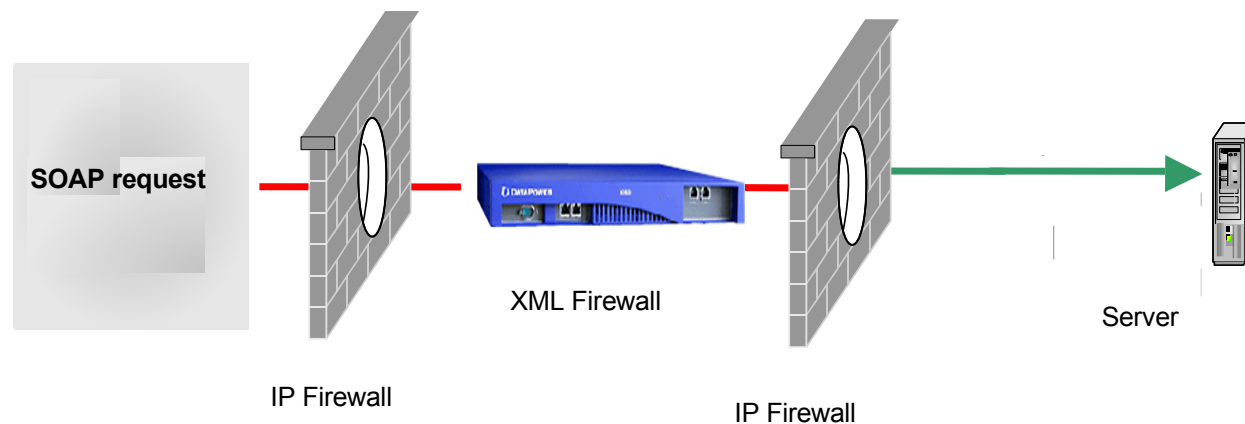
XML Attack Example – DataPower

- Message was now resent to the Web service via the DataPower proxy
 - Note the timestamps, message handled and rejected in milliseconds
 - Attribute limit of 128 per element was exceeded, message REJECTED

17:58:45	multistep	error		<u>37652</u>	9.49.218.77	xpath-eval default implied action Parse input as SOAP, attempt pipeline failed: attribute limit of 128 per element exceeded, aborting at line 135 of http://9.42.117.9:5026/WebServices2Web/services/SimpleServiceBean
17:58:45	xmlparse	error		<u>37652</u>	9.49.218.77	attribute limit of 128 per element exceeded, aborting at line 135 of http://9.42.117.9:5026/WebServices2Web/services/SimpleServiceBean
17:58:45	xmlparse	debug		<u>37652</u>	9.49.218.77	Parsing document 'http://9.42.117.9:5026/WebServices2Web/services/SimpleServiceBean'
17:58:45	xslt	debug		<u>37652</u>	9.49.218.77	xslt Compilation Request: Found in cache (store:///identity.xsl)
17:58:45	xslt	debug		<u>37652</u>	9.49.218.77	xslt Compilation Request: Checking cache for URL store:///identity.xsl
17:58:45	multistep	debug		<u>37652</u>	9.49.218.77	Stylesheet URL to compile is 'store:///identity.xsl'
17:58:45	schema	debug		<u>37652</u>	9.49.218.77	xsd Compilation Request: Found in cache (store:///schemas/soap-envelope.xsd)
17:58:45	schema	debug		<u>37652</u>	9.49.218.77	xsd Compilation Request: Checking cache for URL store:///schemas/soap-envelope.xsd
17:58:45	mpgw	debug	BillH	<u>37652</u>	9.49.218.77	Generating chunked response stream to front
17:58:45	mpgw	debug	BillH	<u>37652</u>	9.49.218.77	Found content length 49778167 HTTP input
17:58:45	mpgw	debug	BillH	<u>37652</u>	9.49.218.77	HTTP Transaction # 1 on this TCP connection,
17:58:45	mpgw	info	BillH	<u>37652</u>	9.49.218.77	Received HTTP/1.1 POST for /WebServices2Web/services/SimpleServiceBean from 9.49.218.77

Security Recommendation Scope

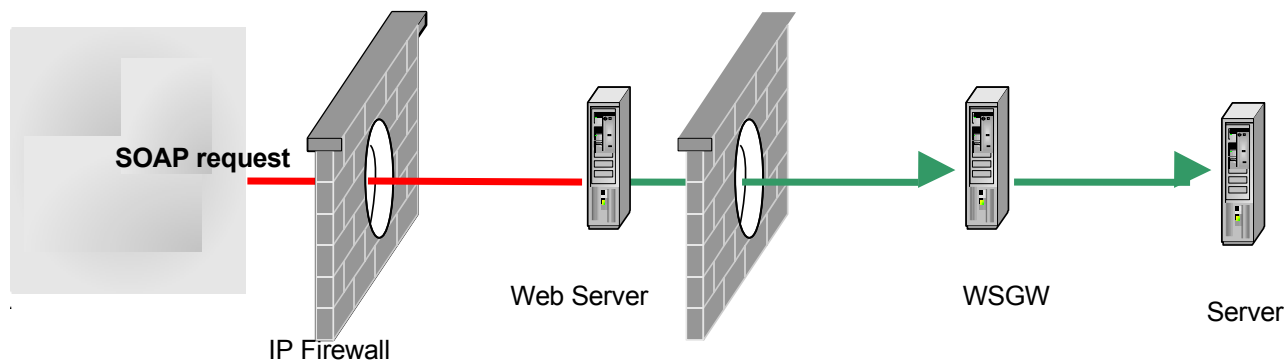
- We will give three recommendations on the next three slides.
- The scope of these recommendations is for Internet facing web services applications where client requests are inbound XML
 - This is where security is of greatest importance
 - This is where there is the largest client need
 - Note: “Internet facing” really means “crossing trust domain”



Recommendation: *DataPower is appropriate for DMZ deployment unlike other current WebSphere Application Server based solutions (e.g., WSGW)*

Appropriate for DMZ Deployment

- Hardened box, built for the DMZ
 - No user visible operating system, No installable software, Tamper proof screws, Secure by Default
- By deploying DataPower in DMZ, can eliminate a hop from current solutions
 - E.g., before DataPower we recommended this:



- DataPower could replace Web server and WSGW

Recommendation: *All Internet facing systems that provide for inbound web services requests should use DataPower as their XML firewall without regard to other considerations.*

- XML-based threats against web services based systems are the “new frontier” for attackers
- There are many different types of attacks
 - Denial of service – flooding small well-formed documents that are difficult to parse
 - Content-based – huge documents, complex documents, large attachments, injection
 - Schema replacement – poison, invalid, or substitute schemas
- DataPower acting as an XML firewall can detect and reject these attacks
- We currently have little protection against these in our existing software-based SOA products
 - WebSphere Application Server, Enterprise Service Bus, Web Services Gateway, Process Server, Message Broker, Service Integration Bus

Recommendation: *DataPower should be used as the Point of Contact proxy for systems that accept inbound web services requests using WS-Security where performance, standards currency, or flexibility is important. Use TAM and TFIM with DataPower as appropriate.*

- DataPower WS-Security support versus native WAS support
 - Faster
 - More flexible – with custom XSL can alter messages that are not “quite right”
 - More complete – includes varying degrees of support for WS-SecureConversation, Kerberos tokens, SAML tokens, etc
- DataPower for central policy enforcement
 - Can natively perform some (but not all) access control and identity mapping function provided today by WSGW with TAM and TFIM
 - Can use TAM and TFIM when appropriate for more complex scenarios

Shameless Plug

- IBM WebSphere DataPower SOA Appliance Handbook
- Published by IBM Press, due late 2008
- By Bill Hines, John Rasmussen, Jaime Ryan, Simon Kapadia, Jim Brennan
- http://www.amazon.com/gp/product/0137148194?ie=UTF8&tag=dph-20&link_code=as3&camp=211189&creative=373489&creativeASIN=0137148194



DataPower Resources

- DataPower Forum, (talk to the Pros!)
 - <http://ibmforums.ibm.com/forums/forum.jspa?forumID=2418>
- DataPower Family Overview
 - <http://www-306.ibm.com/software/integration/datapower/>
- DataPower Library
 - <http://www-306.ibm.com/software/integration/datapower/library/index.html?acss=wdp121007>
- DataPower Support
 - <http://www-306.ibm.com/software/integration/datapower/support/>
- DataPower Firmware and Documentation
 - <https://www14.software.ibm.com/iwm/web/swg-datapower/index.shtml>
- Bill Hines' Comment lines on XML security
 - http://www-128.ibm.com/developerworks/websphere/techjournal/0603_col_hines/0603_col_hines.html

Legal

- **© Copyright IBM Corporation 2008. All rights reserved.**
-
- **IBM, the IBM logo, the e-business logo and other IBM products and services are trademarks or registered trademarks of the International Business Machines Corporation, in the United States, other countries or both. References in this publication to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates.**
-
- **Product release dates and/or capabilities referenced in this publication may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way.**
-
- **Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries or both.**
-
- **Microsoft, Windows, Windows NT and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries or both.**
-
-
- **All other trademarks, company, products or service names may be trademarks, registered trademarks or service marks of others.**