Rogue XML Specifications

[Dethroning Insecurities In Web.xml Schema]

By: Aditya K Sood
Handle : Zeroknock
http://zeroknock.metaeye.org

Abstract :

This article solely relates to the the insecurities that remain in the XML schema.This schema is defined for any web server that relates to peculiar web servicing application.This is actually based on the AJAX framework.The xml specification act as an interface to server objects.The interface which is being provided by the xml schema directly configures the server on the fly which is dependent on the specific service providing servlets.The wrong schema in the web.xml or the index.xml leads to the origin of the web attack base that really disrupts the functioning of the server.As a result of which lot of information gets leveraged..I am going to discuss the schema designing and relative effects and what happen if it is not configured properly.

The Anatomy Of Web.xml File:
First of all we have to understand the exact structure and working relativity of the web.xml file because this file serves as the classic layout to handle the server objects on the fly without having direct relation with the server.This file considered to be as the base for the servlets functioning which provide service to the client applications.This specific file act as an interface between client request and the server.At core before calling any servlet the web server checks the specification provided in this file and what liable properties are set for the generic requests that are requested by the client.This model is based on the AJAX framework where direct access to the server is barred.This is done to have uni standard and platform independent code based on xml.It can be put into practise directly on any platform.This reduces the complexity on the server for the administrators and the coders.This model of hierarchial implementation of xml schema makes the coder to easy traverse their code and help them to really access the server objects through the web,xml schema.This file is of great importance because a simple manipulation in this file generates loophole in the applications.This file provides an ease to the programmers but at the same time have raised the risk of web attacks on the server from top to bottom.Now we are going to understand what actually this file comprise of and the attacks that are associated with it.The web.xml file is a deployment decriptor file.This file is held in the application's WEB-INF directory. It define a number of parameters that are used when the web application is deployed into the Tomcat Servlet/JSP container. As the working stats of this file makes this very clear that the base it serves is of extreme importance in the web transactions and objects association.The major point of this file is providing specification of the servlet and how these are configured.Before getting to the descriptors and the web attacks associated with it , we just going to have brief outlining of the servlets with respect to xml schema defined for them.

The Default Servlet : Anonymity:
There is always a default servlet present on the server which is used to serve static resources on all the web application resources.The server objects are based on static functioning and dynamic functioning.The static objects are already and defaultly present but this is not a case with the dynamic objects as these comes to play during run time execution .These specific servlets are put into practise as the server received a client request and choose the servlet which is best for that by seraching through service calls.The server no doubt look at the web.xml file in the server directory for the working layout of the servlet.The descriptors present in the file throw bottom to top information of every servlet and the security roles being provided in the file.Now we will look into the factor which shows how the default servlet handles the request as soon as the URI is placed into browser.If no welcome descriptor present the underlined condition occurs:

[*] A 404 document appear on the web page.
[*] A file listing will be displayed.

These results when come into practise let us know how the descriptors are configured.Any wrong configuration leads to the formation of web attack base.We will look into attack types and information leveraged through this within spec of time.So you have seen very clearly that the default servlet is of statistical importance.This we are talking about static servlets.You can think how critical is to define the security parameters in the file because the processing is going to happen at the execution time.

The Invoker Servlet : Anonymity
This servlet is always mapped in to the URLpattern /servlet/* but this can be mapped into another pattern as well based on the server properties.If you really want to look into the extra path information then it must start with the standard HTTP based servlet.The HTTP servlet is responsible for the requests to be processed which are initiated by the client.It uses theHTTP class information to handle the requests.

The Processing Servlet : Anonymity
This servlet is responsible for the processing Of SSI directives which are called to be as Server side includes.These SSI tags hold the functioning of server objects to handle major requests.Basically these are included in the HTML pages which then gets subjected to the server container in web directory.If any request is put forward by the client , the SSI directives included in that , must getprocessed first to throw response to the client.The requests are based on the web page specification too.This is used for embedding in web HTML pages so that more generic web applications are created sticking to the base.For processing JSP pages the server uss jsp compiler and execution servlet to serve the request based on JavaServer Pages.As the name suggest this servlet is mapped in to the *.jsp URL pattern.For processing gateway based requests ie CGI the cgi based servlet is used to hold and process the request related to common gateway interface.The servlet is mapped into the *.cgi URL pattern for disseminating the cgi parameters.So at this point of time we come to know about the various kind of servlets that has to be properly configured.The mapping of servlet is always a prime part of the configuration process.The elements are:

A] The Servlet Name.
B] The Mapping.
C] The Initialisation Parameters

There is specific name to every single servlet and it should be mapped properly in the schema so that no false linking encountered.The initalisation parameters are required to have control over the working part of servlet.It includes the generation and processing of servlet.

The Servlet Example:
JSP Servlet

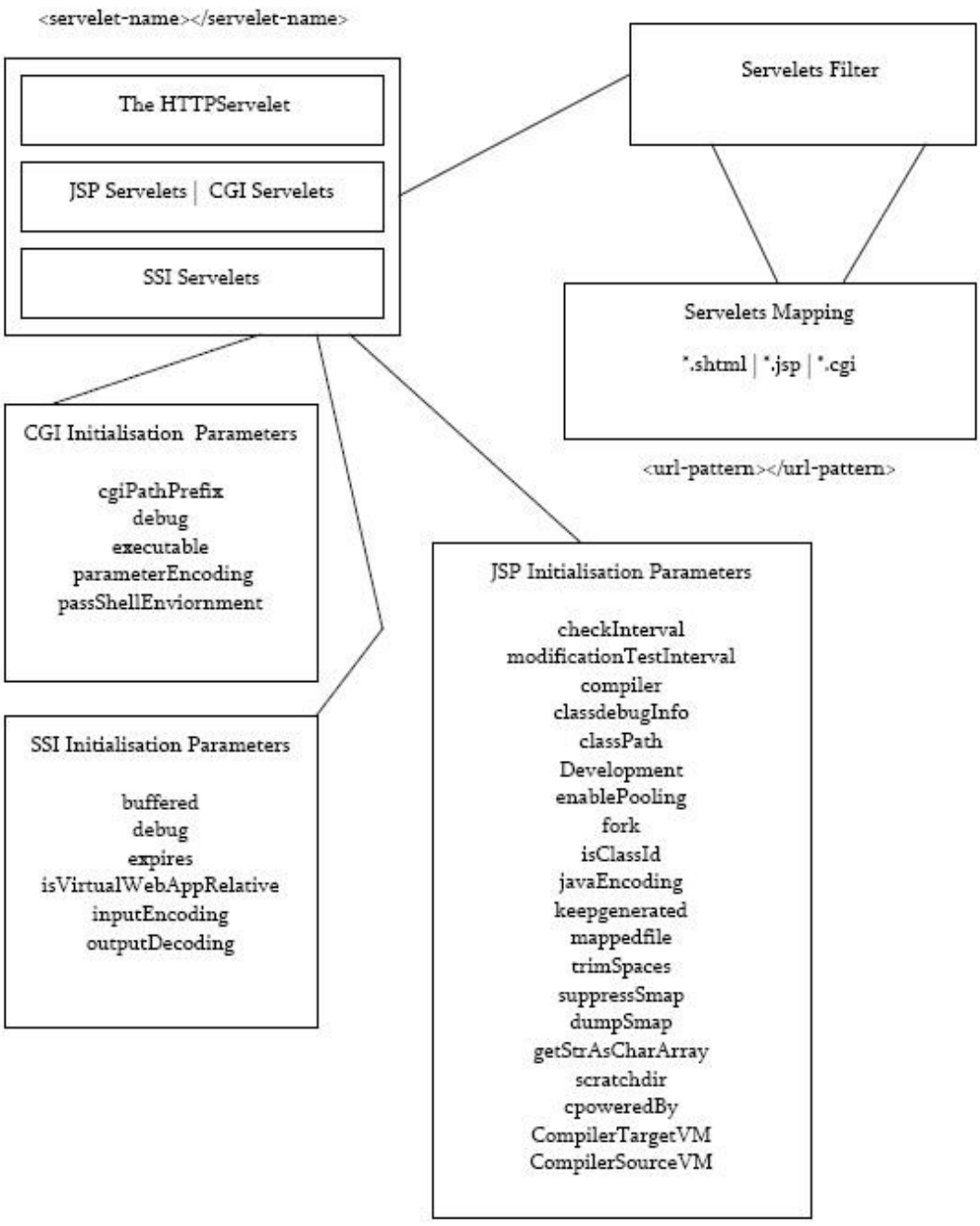
```
<servlet>
<servlet-name>jsp</servlet-name>
<servlet-class>org.apache.jasper.servlet.JspServlet</servlet-class>
<init-param>
<param-name>fork</param-name>
<param-value>false</param-value>
</init-param>
```

A schematic view of web.xml.The core functioning with respect to servlets have been show in the underlined mapping.

The Schematic View : Servelet in Web.xml

<servelet-mapping></servelet-mapping>

The servelet , initialisation parameters and mappings in a relative model.



This is very definite schematic model to understand how the servlets are arranged in the web server container.

[Rogue XML Specifications]

The very first question comes to mind is , how web attacks through url find their base to leverage information.This is clear in the context as web server first process the web.xml file to undertake the required descriptors specified for particular servlet.Lets see which descriptors let these type of attack to happen.We try to fuse the techniques with xml specification and analyse how the descriptors are deployed.This encompass the positive and negative elements of the insecurity.

False Handling in Starting File :
This is very good practise to provide the welcome list of files.In this the default link is provided to the the file which is to be displayed in the web browser to the user when ever no transaction request is undertaken.This is also helpful in the cases when the server is unable to handle the input.It automatically redirects the page to main page.During my penetration sessions i have observed that sometimes the welcome file link is given but not redirected internally through which information leakage is accomplished.This method is put into practise by the following descriptor as:

```
<welcome-file-list>
        <welcome-file></welcome-file>
        Example:
        <welcome-file-list>
        <welcome-file>index.html</welcome-file>
        <welcome-file>index.htm</welcome-file>
        <welcome-file>index.jsp</welcome-file>
</welcome-file-list>
```

This type of working standard is very useful because i have seen many times if these descriptors are not deployed in a generic manner a lot of information is extracted because server is not able to redirect it to the page what a user is requested and hence again the information is revoked.

Fused Behavior Of Error Links:
The error links and error management is always a point of concern in the web management and application development.There are defined descriptors provided for managing errors which are to be applied very craftly in the web.xml file.If these descriptors are not configured properly the information disclosure gets relevant and internal mirror gets clear.So there must be a point to redirect when ever error is generated to a definitive errorpage.The descriptors used are as

A] Error Code Specification.
B] Exception Type.
C] Error Page

When ever an error is generated from the client , the server possess an error handling mechanism.The mechanism for an error handling works on the configuration defined in the xml file.The error page holds the error code and the type of exception occurs.I think you have usually seen whenever SQL injection occurs , the error code and the exception is displayed on the screen.Thats how the error is subjugated when ever an exception occurs.On the other part hackers use this information to launch further attacks.So if the error handling is not configured in a definite way lot of information will be retrieved from the server.

:**<error-page>**
<error-code></error-code>
<exception-type></exceptiontype><location>path/to/resource</location>
Example:
<error-code>theErrorCodeNumber</error-code>
<exceptiontype>java.exception.TypeOfException</exceptiontype>
<location>path/to/resource</location>

Session Specification:
The session management is also very crucial because it sets the time limit for a particular session to be established with the server .The filters applied check the session initiation parameter ie the client that initiates the session is authorised or not with the server.This comes handy as it automatically break the session with the defined client.The entities used in this are:

A] The Session Configuration Parameter.
B] The Session Timeout.

The session configuration parameter sets the session for a particular user when he is trying to access specific resource.The session timeout limitisize the time period by applying time constraint for specific session.

<session-config></session-config>
<session-timeout></session-timeout>
Example: <session-timeout>2</session-timeout>

Mismanagement In Application Security Roles:
The mismanagement in the specification of application roles can lead to application role attacks.These attacks are generated when proper security roles are not applied to the user.This means if the role of a user is not configured well , the user can access any of the resource placed on the web server.This happens when some modules are active for definite users.But when roles are not configured according to the usage of users the calling of undesired module becomes possible.See the configuration of security roles are really crucial because it sets the demarcation between the normal users and the admin users.In this constraints are added so that simple user cannot transact the request if the authorisation parameter gets failed.This holds definitive importance because the query for transactions in business model executes directly as server go after the descriptors placed in the web.xml file.The application role configuration is divided into following entities:

A] Web Resource Collection
    A.1] URL Pattern.
    A.2] HTTP Method.
B] Authorisation Constraint.
C] User-Data Constraint.

The first entity sets layout for the type of URL and HTTP method to be used for accessing specific resource.The URL pattern shows the accessing link to that resource and HTTP method shows whether GET or POST request is issued to that resource.Now the basic point is how the constraints applied on

the resources.The authorisation constraint comes after the application of authorisation method that sets the limiting parameter on the authorisation.The user data constraint tell how the user access the data on the web server.So all these three entities when gets integrated comprise the application role configuration in the web.xml schema file.Now lets have look on the configuration of these descriptors in the schema file:

Lets look into it:

```
<security-role>
<description>The description</description>
<role-name></role-name>
</security-role>
```

The constraints based on security descriptors which sets the role functioning for a user as a developer.These constraints are of high importance for security reasons.

```
<security-constraint>
<web-resource-collection>.
<web-resource-name></web-resource name>
<description></description>
<url-pattern>/url-pattern>
<http-method>/http-method>
<auth-constraint>
<description</description>
<role-name></role-name>
<user-data-constraint>
<description></description>
<transport-guarantee></transport-guarantee>
</security-constraint>
```

The constraints are applied in this way.

Weakness In Login Configuration Parameters:
The weakness in login configuration parameters lead to authentication bypassing attacks.These are defined against any web application.You open awebsite there is a login page appears , which require username and password to login into web page.If properchecks are not applied in the web.xml then a rogue input willl leverage lot of information.The login configuration of xml schema deployment descriptors are divided on three basic elements which are explained as:

A] Authentication Method.
B] Realm Name
C] Form Login Configuration.
    C.1] Form Login Page
    C.2] Form Error Page.

The very first element states that the type of authentication method used by the server container to let

the users login.This is very specific configuration as because it sets the security base for login entities.Moreover if this parameter is missing then the authenication will be occurred in clear text which in itself is a major security breach.The authentication methods can be browser authentication , USER HTML-Form authentication or Certificate based. The second element describes the realm in which the login configuration is defined The system security realm is a collection of security information that is checked when certain operations are performed on the server. The servlet security realm is a different collection of security information that is checked when a page is accessed and basic authentication is used. The third element describes the login form configuration..This also gives an importance to response that is generated from the user mistake and what output is to be shown on the screen.If this is not configured well then internal information can be easily extracted.Thats what hacker do , they try to generate errors and analyse the response from that.So the login schema is of importance with respect to dethrone the authentication bypassing attacks.
The Parameters:

```
<login-config>
<auth-method> </auth-method>
<realm-name> </realm-name>
<form-login-config>
<form-login-page> </form-login-page>
<form-error-page> </form-error-page>
</form-login-config>
```

If any of the descriptor gets misconfigured the attack base will get solidify.Let see the exemplary layout.
Example:

```
<login-config>
<auth-method>Basic</auth-method>
</login-config>
```

Weakness In Reference Specification:
This is also one of the major factor in web application attacks because this also leverage lot of information when misconfigured.The reference element sets the external link for the unique resource present in the web server.It also includes the type of resource and the authentication method used to link to that external entity.This allows the servlet code to look up a resource by a "virtual" name that is mapped to the actual location at deployment time. As a result of which inter-relation is set between two different realms and inter exchange of data is possible.It lay stress on the fact that how to have limiting access and generic configuration for a particular resource so that it functions well.The descriptors are:

A] Resource Reference Name.
B] Resource Type.
C] Resource Authentication.
D] Description.

The first entity sets the reference for an external resource.This means the name to be used as reference in the xml schema.The second entity sets the type of resource whether java based or asp based is used and further the authentication method ie how the web server access the external resource from the

specific link.The description tag is used for configuration knowledge in the file.Lets see how the reference parameters are set:

<description></description>
<res-ref-name></res-ref-name>
<res-type></res-type>
<res-auth></res-auth>
<res-sharing-scope></res-sharing-scope>

The Example:

<Description>Java Resource</description>
<res-ref-name>Java Servlet</res-ref-name>
<res-type>JSP</res-type>
<res-auth>Application</res-auth>

So you can look very clearly how the entities are assembled together to perform definite functioning.

Servlets Mapping Parameters:

The servlet mapping is very necessary component of every XML framework because these are the objects that are called by the server container for every request that a client make.The servlet-mapping element defines a mapping between a servlet and a URL pattern.Actually this is done to ensure reliablity in the working pattern of the servlets which are used by web server container to process the requests.The mapping defines a way thorough which when ever designated URL is called from the browser with the required extension the specific servlet invoke which perform further functioning.The entities involved in this are:

A] Naming Of Servlet.
B] URL Pattern

The naming of servlet means , which type of servlet whether cgi ,jsp,asp etc is used. The URL pattern means where the servlet will reside on the web server directory.So this sets the servlet layout in view point of mapping.If this configuration gets wrong and mapping is misconfigured the internal information can be leveraged very easily.The wrong mapping will set the wrong vector of working. Lets look into it:

<servlet-mapping>
<servlet-name></servlet-name>
<url-pattern></url-pattern>

Example:

<servlet-mapping>
<servlet-name>cgi</servlet-name>
<url-pattern>/cgi-bin/*</url-pattern>
</servlet-mapping><servlet-mapping>

The wrong mapping will result in the disastrous consequences related to web application.

Enviornment Entry Specifications:
The enviornment entry actually describes the environment entry for an application.The enviornment here refers to the parameters of the application.Its a kind of setting property for an application but based on the specification of that application.This is also crucial because environmental settings play an important role in working approach of the application.Lets look at the entities of environmnet entry:

A] Environment Entry Name.
B] Enviornment Entry Value.
C] Enviornment Entry Type.

The very basic descriptor sets the enviornment entry name for the application that must be associated with web server.Then it describes the valu and type of the enviornment entry for the application.The java type can be:

java.lang.Boolean
java.lang.String
java.lang.Integer
java.lang.Double
java.lang.Float

So you can look the type define the vraible type.Often a lot of errors are generated from this kind of base.so if the parameters are not configured properly for the enviornment the errors will be generated and hence favours the web application attacks.Lets look into the configuration layout:

<env-entry-name> <env-entry-name>
<env-entry-value><env-entry-value>
<env-entry-type> <env-entry-type>

The descriptors define the entry point in clear context.

Specification Of Context Parameters:
It sets the Web Application servlet context initialization parameter.The intialisation of servlet parameter define the context in which application is deployed.The various descriptors used in this are

A] Parameter Name
B] Parameter Value

The context is elaborated on the basic parameters , termed to be as name and value.Tha name entity holds the specific name of the application servlet and value holds the servlet definitive value for the context in which it is applied.

<description> <description>
<param-value> <param-value>
<param-name> <param-name>

Specification of context is very crucial for generic working of the application servlet.A bit of false configuration can throw lot of information.So this is also considered to be as the crucial configuration parameter in web.xml schema.

Servlet Filtering:
The filter element defines a filter class and its initialization parameters .The filters are defined for servlet classes and limitsize the working capability of the servlets to work in a best possible way.The filtering removes the unwanted specifcations that are adhered to the application servlet or classes.The entities are:

The Filter Parameters:
A] Filter Class Specification.
B] Initialisation Parameter.

The Filter Mapping Parameters:
A] Naming Of Filters.
B] URL Pattern Specification.

The filter is set on the servlet class having definitive layout for filter class that is defined against for web application servlet.The intialisation parameters are needed for this.The mapping is set with filter mapping parameters.The URL pattern that is to be filtered is specified in that.The improper filtering can cause lot of information leakage.The artistic hackers know how to manipulate the wrong parameters.Let see the descriptors:

<filter>
<display-name> <display-name>
<description> <description>
<filter-class> <filter-class>
<init-param> <init-param>
</filter>
<filter mapping>
<filter-name>
<url-pattern>
<filter mapping>

Thats how the required filtering is done.

Specifications Of Referential Enterprise Java Beans:
There is also definite provisions for setting references to EJB as local entity or remote entity.This is done to undertake the deployment of EJB servlets as a remote resource which can be accessed very easily.The references hold a great importance because this reduce the complexity in servlet layout as direct inclusion from the remote location is possible.The descriptors define in the web.xml file define the in control functioning of the EJB servlet.The entities are:

A] Enterprise Java Bean Referential Name.
B] Enterprise Java Bean Referential Type.
C] The Home Interface.
D] The Remote Interface.
E] The Enterprise Java Bean Linking.
F] Specification For Security Role.

The various descriptors defined in EJB refrencial tag provides very generic servlet functioning as the

references hold the linking and to the address of the remote locations where the servlet is accessed.This is all defined clearly with EJB parameters.The home interface sets the local interface and remote tag the destination interface.The servlet is linked between this.If misconfigurations happen , it not only effects the local working approach of servlet but also remote point.The hackers are very sophistacated in finding links but the main aim is find the internal links which leverage lot of information.So the EJB references have to be set in best possible way.Lets look at the descriptors:

<ejb-ref-name> <ejb-ref-name>
<ejb-ref-type> <ejb-ref-type>
<home> <home>
<remote> <remote>
<ejb-link> <ejb-link>
<run-as> run-as>

The EJB reference is mapped to the actual location of the EJB at deployment time by defining the mapping in the WebLogic-specific deployment descriptor file, weblogic.xml. Use a separate <ejb-ref> element to define each reference EJB name.This is done to set reliability factor in the functionality.

So at this point of time we have understood the attack base from where the web applications attacks are launched and the working approach of the schema elements.The main point of rogue specification is to look at the weakness and mis configured parameters that let the exploitation to trigger. Now we have analyse the rogue specification of web.xml file , now we will have a look on the class of web application attacks that arise due to result in the misconfiguration of these descriptors.

The Possible Attacks Based On XML Schema : Base Web.xml
This class of attacks is mainly constraint to web ie it adheres to URL specifically and some related malformed input.The basic structure deals with this is that the wrong configuration in the web.xml file leads to bypassing of security and transaction procedures for a definite user which puts a request to the server container.Theattackers are very sophisticated thinkers when generic attacks are laid.A very tiny mistake in the configuration can lead to auto updation of the database or queries on the server.These

descriptors which define for a particular servlet and its dependent mappings play a crucial role in making application secure and robust working stats.Lets get deep into it and we will look what kind of attacks can be possible when servlets are not managed properly that dethrone the web application security.

Information Leveraging Attacks :
An attacker can leverage lot of information by executing information leveraging attacks.This encompass the false positives as providing very crafty input when not handled by the server results in lot of information disclosure.The information is leveraged with very different techniques but i will specify only those that point to our discussion base .The techniques involved in this are:False Input Validation Attacks.Parameter Stripping Attacks.Index Tracing On Servers.Parameter False Argumenting.First of all we will look into the type of techniques that results in negative layout one by one because it is necessary to check the attack base and its characteristics and how these attacks happen and the web is getting exploited by it.

False Input Validation Attacks:
These type of attacks are the prime type of attacks because the hackers believe in generating errors to leverage lot of information from the servers.This actually more false coding prone as if coding is not properly done there must remain a loop hole which cannot control the wrong input and hence gets in misconfigured state which further results in false application working.The application is not able to handle the false inputs correctly thereby displaying a lot of internal information.This what attackers want to happen because with no set of intrusion a lot of information is generated.

Index Tracing On Servers:
This is also a very peculiar class of web attacks.These attacks are also a result of misconfiguration is web server files which check the access control over the directories that are present on the web servers.If you see clearly in the URL present in the address bar of web browser number of directories are being displayed as definitive URL consisting of a chain which holds the specific web resource you have requested.Soeme times if misconfiguration is present in the XML files , the tracing can become very easily because if directory is traversed by the attacker and not proper rights or credentials are set for the web user the other files can be indexed on the server and displayed on the web spaghettis is also a generic attack on the web that throw lot of information .

Parameter Stripping Attacks:
These kind of attacks adhere to the URL of the website..Normally what happen , the websites which are based on three tier concept ie database at the back end , a lot of parameters are to be passed through the URL which serve as the database elements to let the web transactions occurring flaw occur in this will directly hits the database server.Now when the attacker use to strip the URL , the parameters are get trespassed ie the database is tested with different URI which is being encountered by stripping the URL parameters.As a result sometimes database cannot hold the different parameters because the query constructed is not full and information is leveraged.So these attacks are getting very high now a days because it hits the last tier of the three tier web applications.The attack base has now been shifted to the way that not to involve in the penetration much but still get the max information from the web.

These are the very specific attacks that can occur as the result of false XML specifcations.We have to understand and apply the knowledge in best possible way to combat the attacks.

[Rogue XML Specifications]

Conclusion:

The class of attacks that are undertaken is very context driven in which these are applied.The xml specifications are of great importance in making web applications secure and robust transactions.The panorma of security and breaching always go on.The human malfeasance is always been there that generates loophole.This is a process which always move on.