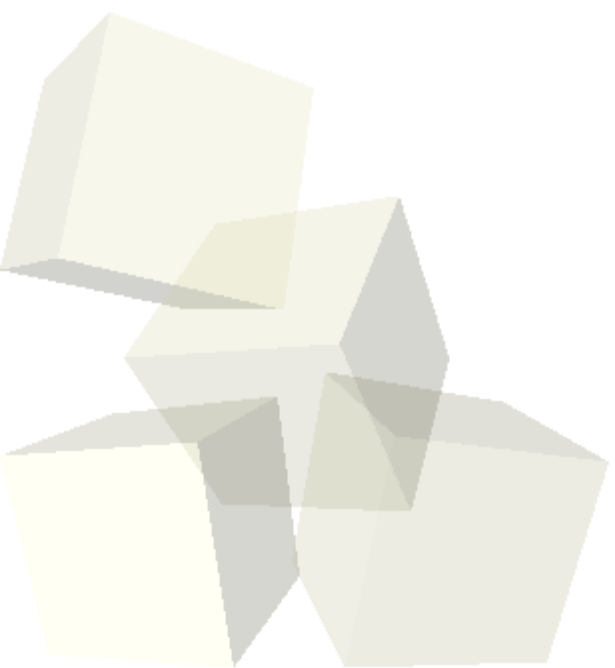




# Metacoretex-NG

**Fist Conference  
Octubre 2005**

Edge-Security





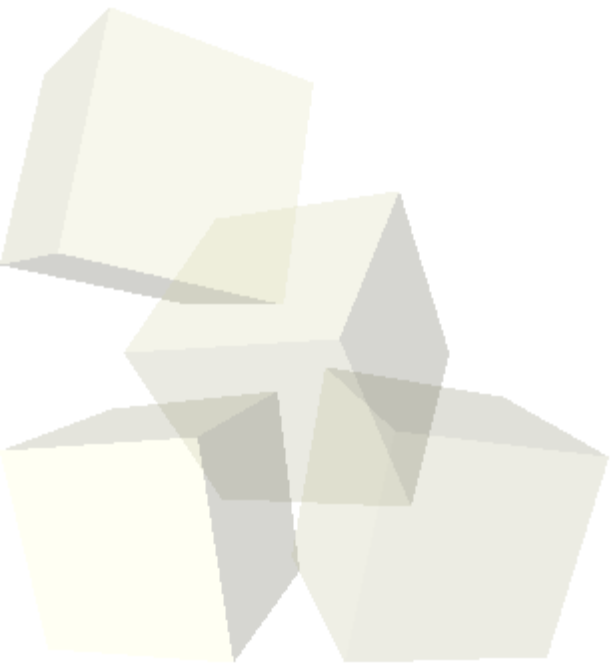
## ■ Índice de la presentación

- ♦ 0. ¿Quiénes somos?
- ♦ 1. Introducción
  - 1.1. ¿Qué es Metacoretex?
  - 1.2. Historia
  - 1.3. ¿Qué es un framework?
  - 1.4. Otros productos
- ♦ 2. Metacoretex-NG
  - 2.1. ¿Porqué Metacoretex-NG?
  - 2.2. Objetivos
  - 2.3. Implementación
  - 2.4. Características
    - 2.4.1. Kb
    - 2.4.2. Dependencias
    - 2.4.3. Probes
- ♦ 3. Demostración
- ♦ 4. Ruegos y preguntas



# Objetivos de la presentación

- El objetivo principal es dar a conocer el proyecto Metacoretex-NG
- Mostrar las capacidades de este software
- Y enseñar las nuevas funcionalidades en las cuales se está trabajando





# 1.1 ¿Qué es Metacoretex?

- Metacoretex es un **framework** para realizar escaneos de vulnerabilidades, auditorías de seguridad y tests de intrusión en **Bases de datos**.
- Está basado en probes (pruebas). Éstos son programas independientes en java que realizan algún tipo de chequeo, escaneo, prueba, consulta, obtención de datos, etc.





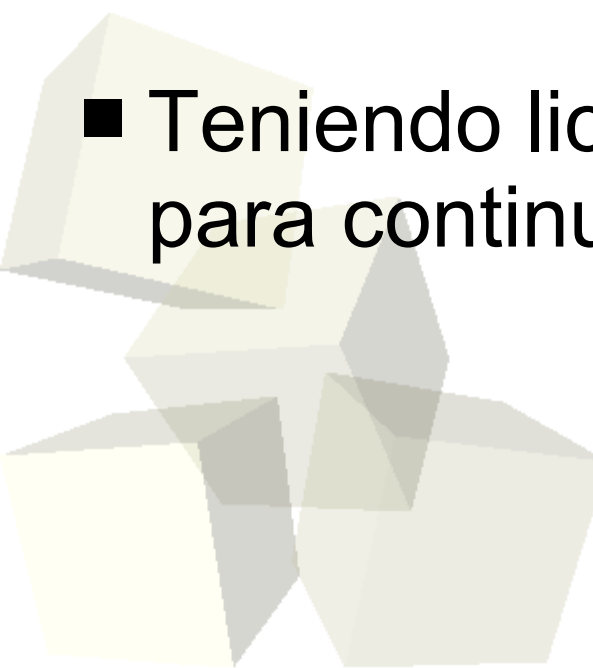
# 1.1 ¿Qué es Metacoretex?

- La posibilidad de ampliar/añadir pruebas hacen que sea una herramienta muy flexible.
- Está desarrollado totalmente en Java.
- Soporta:
  - ◆ Mysql
  - ◆ Oracle
  - ◆ MSSQL
- ¿ De dónde viene el nombre ? ;)



## 1.2 Historia

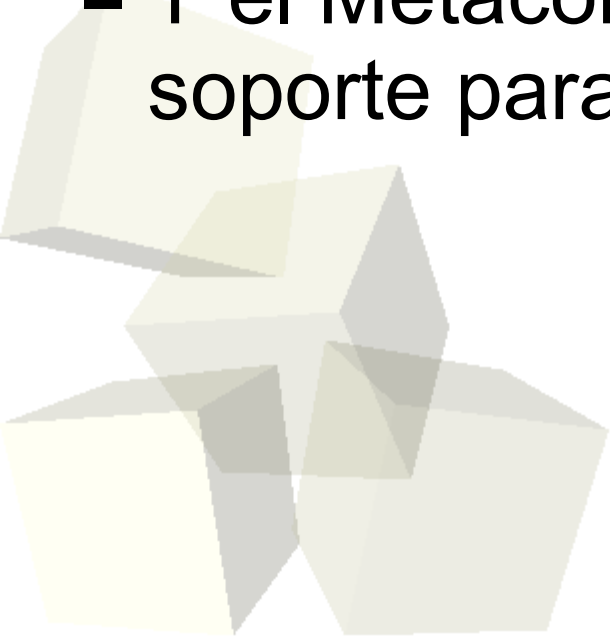
- El proyecto se hizo público en octubre del 2003.
- A partir de esa fecha se publicaron 4 probes nuevos y luego el proyecto quedó congelado.
- En repetidas ocasiones se intentó contactar al autor, pero nunca se obtuvo respuesta.
- Teniendo licencia BSD, decidimos crear un fork para continuar el trabajo.





## 1.3 ¿Qué es un Framework?

- Un framework es una estructura o esqueleto de soporte, utilizado como base para algo que esté siendo construido.
- En nuestro caso, ese “algo” es un escáner de vulnerabilidades de base de datos.
- Y el Metacoretex es el esqueleto que nos da soporte para lograrlo.





## 1.4 Otros productos

- Se pueden encontrar pocos productos con el mismo objetivo que el Metacoretex, ninguno de ellos es open source y gratuito.
  
- ◆ **Appdetective** ([www.appsecinc.com](http://www.appsecinc.com))
  - Soporta:
    - Mysql
    - Mssql
    - Oracle
    - DB2
    - Sybase
  - Os: Windows
  - No permite desarrollo personalizado



## 1.4 Otros productos

- ♦ **Ngssquirrel** ([www.ngsssoftware.com](http://www.ngsssoftware.com)): este producto se vende por separado para cada Base de datos.
  - Soporta:
    - Mysql
    - Oracle
    - Mssql
    - DB2
    - Sybase
  - OS: Windows
  - No permite desarrollo personalizado



## 2.1 ¿Porqué Metacoretex-NG?

- Metacoretex-NG es un fork del Metacoretex, que esta siendo desarrollado para llenar el vacío de estas aplicaciones en el mundo del software libre.
- En lugar de crear una nueva aplicacion se decidió retomar el proyecto y aprovechar el framework ya existente.
- La idea es ir más allá de lo que aporta el proyecto original: poner el proyecto al día, crear una comunidad y mantener la base de pruebas actualizada.



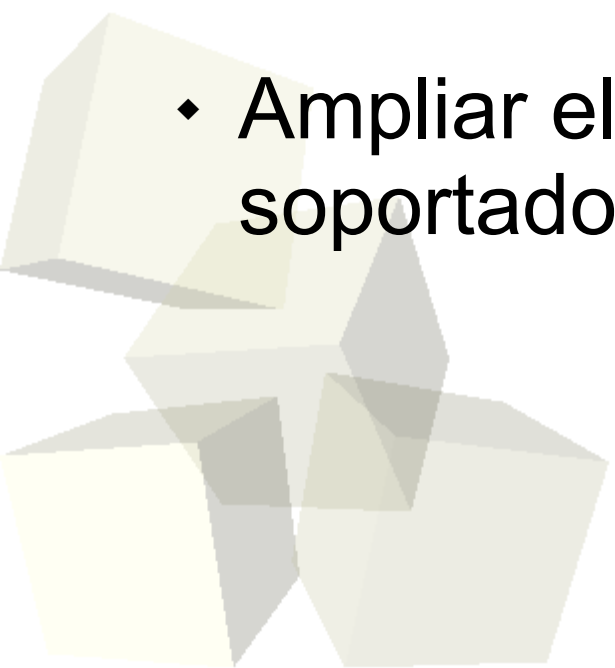
## 2.2 Objetivos

- Los objetivos del nuevo proyecto son:
  - ♦ Refrescar y optimizar la interface del usuario, mejorar la usabilidad.
  - ♦ Actualizar los probes para estar al nivel de los comerciales.
  - ♦ Mejorar el sistema de informes, permitir la personalización y que tengan un aspecto profesional.



## 2.2 Objetivos

- ♦ Centralizar las distintas herramientas existentes en una sola aplicación.
- ♦ Nuevos modos de funcionamiento (Pen-Test)
- ♦ Solución de problemas y Workarounds.
- ♦ Ampliar el número de bases de datos soportados. (DB2, Sybase, Postgresql)





## 2.3 Implementación

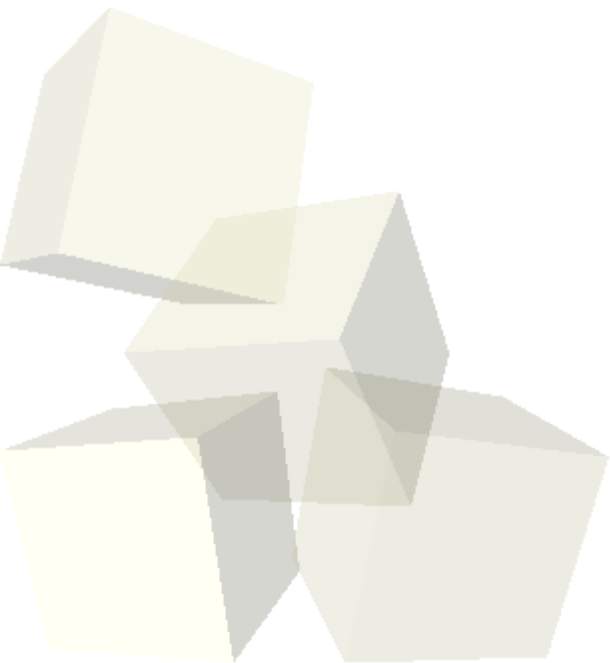
- La implementación es 100% Java.
- La utilización de los drivers JDBC es una gran ventaja, proporcionando versatilidad y buen rendimiento en múltiples bases de datos.
- Podemos considerarlo como un RAD.
- Java es completamente multiplataforma.





## 2.3 Implementación

- Se pensaron en otros lenguajes para migrar el Metacoretex, pero la principal limitación fueron los drivers de las bases de datos.
- Python + GTK era un buen candidato, pero el nivel de madurez de algunos drivers nos hicieron rechazar la idea.





## 2.4 Características

- Algunas de las características más destacadas:
  - ♦ KB (Knowledge Base)
  - ♦ Dependencias de probes
  - ♦ Probes
  - ♦ Wizard





## 2.4.1 Características

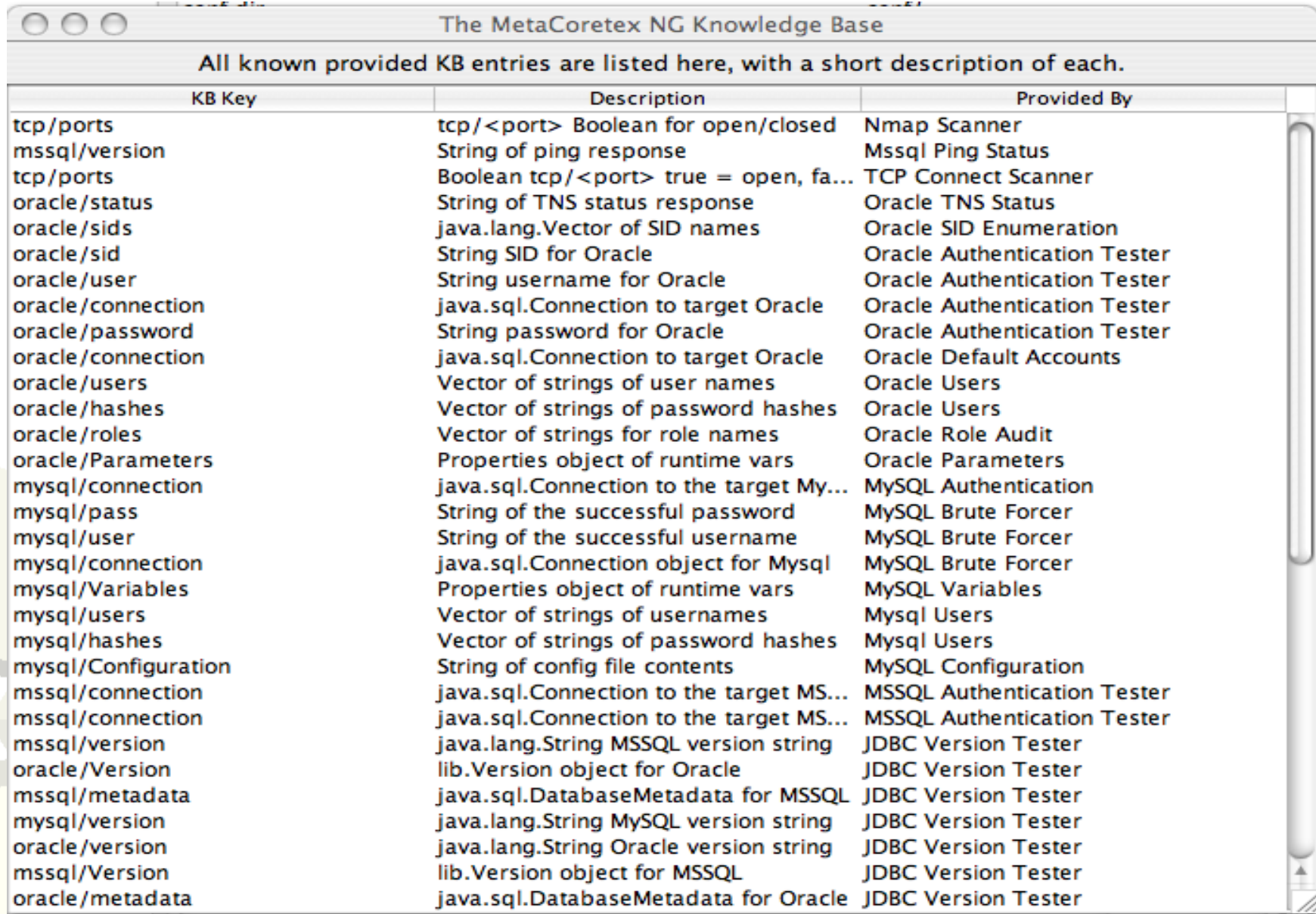
### ■ Knowledge Base (KB)

- ♦ Es una estructura de datos interna donde los probes comparten información.
- ♦ Una de las características más importantes de este framework es que la KB almacena todo tipo de objetos (no sólo strings). Permite, por ejemplo, reutilizar conexiones entre probes.



# 2.4.1 Características

## ■ Ejemplo KB



The MetaCoretex NG Knowledge Base

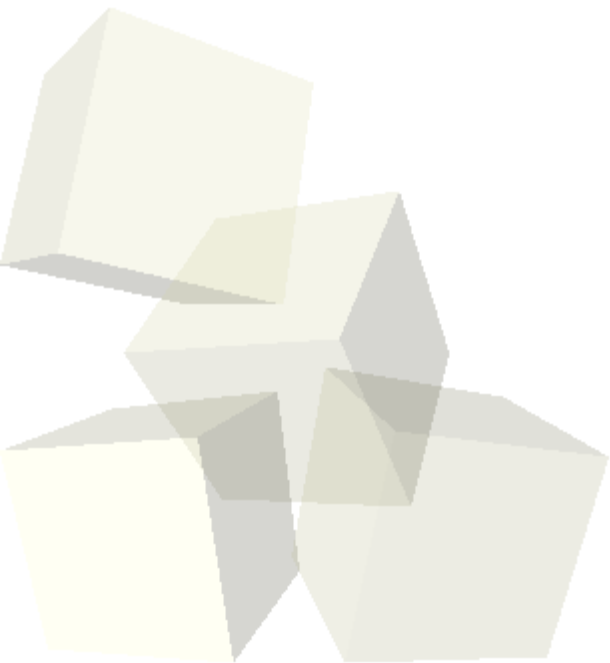
All known provided KB entries are listed here, with a short description of each.

KB Key	Description	Provided By
tcp/ports	tcp/<port> Boolean for open/closed	Nmap Scanner
mssql/version	String of ping response	Mssql Ping Status
tcp/ports	Boolean tcp/<port> true = open, fa...	TCP Connect Scanner
oracle/status	String of TNS status response	Oracle TNS Status
oracle/sids	java.lang.Vector of SID names	Oracle SID Enumeration
oracle/sid	String SID for Oracle	Oracle Authentication Tester
oracle/user	String username for Oracle	Oracle Authentication Tester
oracle/connection	java.sql.Connection to target Oracle	Oracle Authentication Tester
oracle/password	String password for Oracle	Oracle Authentication Tester
oracle/connection	java.sql.Connection to target Oracle	Oracle Default Accounts
oracle/users	Vector of strings of user names	Oracle Users
oracle/hashtes	Vector of strings of password hashes	Oracle Users
oracle/roles	Vector of strings for role names	Oracle Role Audit
oracle/Parameters	Properties object of runtime vars	Oracle Parameters
mysql/connection	java.sql.Connection to the target My...	MySQL Authentication
mysql/pass	String of the successful password	MySQL Brute Forcer
mysql/user	String of the successful username	MySQL Brute Forcer
mysql/connection	java.sql.Connection object for Mysql	MySQL Brute Forcer
mysql/Variables	Properties object of runtime vars	MySQL Variables
mysql/users	Vector of strings of usernames	Mysql Users
mysql/hashtes	Vector of strings of password hashes	Mysql Users
mysql/Configuration	String of config file contents	MySQL Configuration
mssql/connection	java.sql.Connection to the target MS...	MSSQL Authentication Tester
mssql/connection	java.sql.Connection to the target MS...	MSSQL Authentication Tester
mssql/version	java.lang.String MSSQL version string	JDBC Version Tester
oracle/Version	lib.Version object for Oracle	JDBC Version Tester
mssql/metadata	java.sql.DatabaseMetadata for MSSQL	JDBC Version Tester
mysql/version	java.lang.String MySQL version string	JDBC Version Tester
oracle/version	java.lang.String Oracle version string	JDBC Version Tester
mssql/Version	lib.Version object for MSSQL	JDBC Version Tester
oracle/metadata	java.sql.DatabaseMetadata for Oracle	JDBC Version Tester

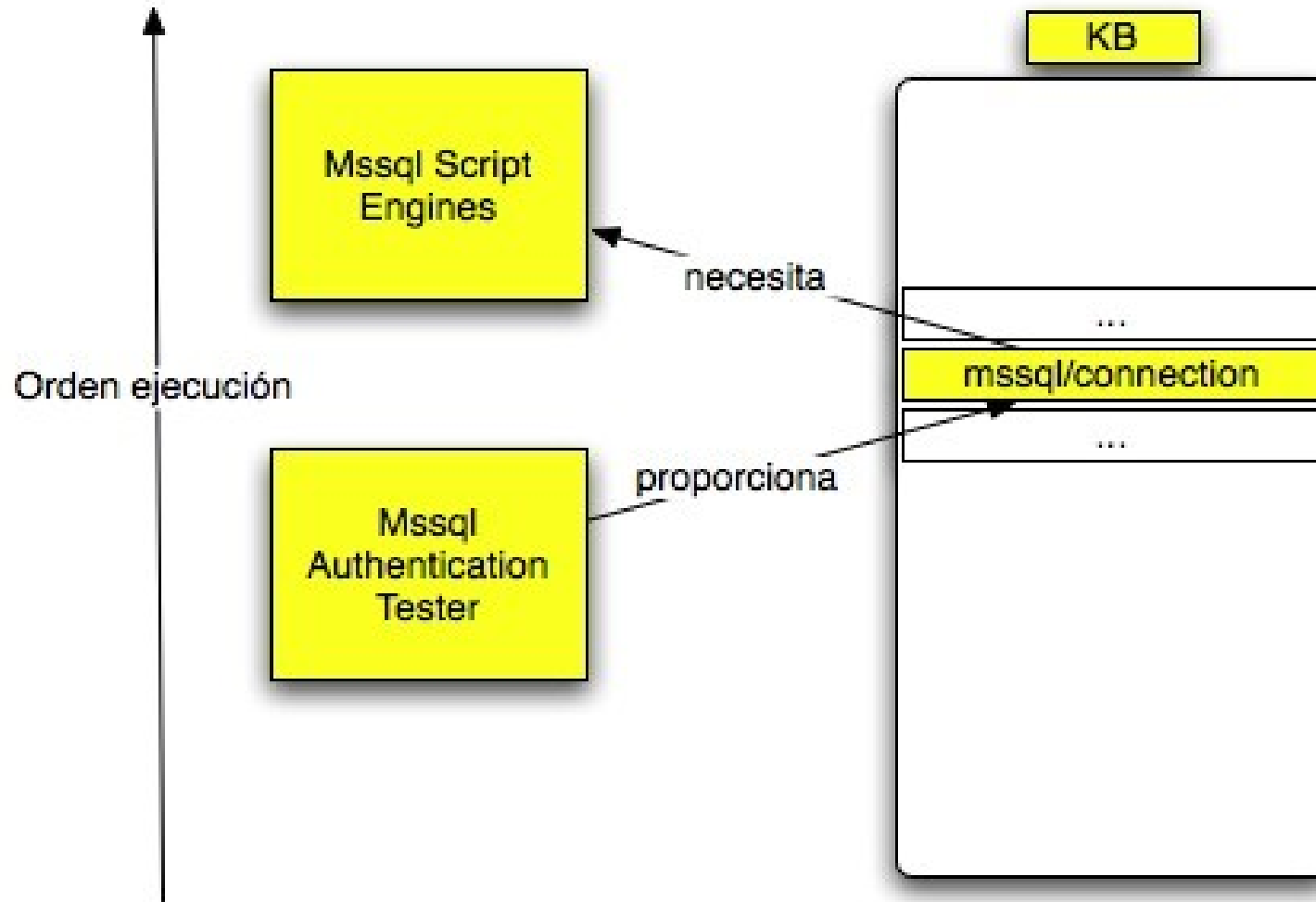


## 2.4.2 Dependencias

- El framework está preparado para que los probes tengan un orden de ejecución. Para ello, los probes tienen dependencias.
- Las dependencias son objetos de la KB. Una mejora será la ordenación automática de la lista de probes en función de las dependencias.



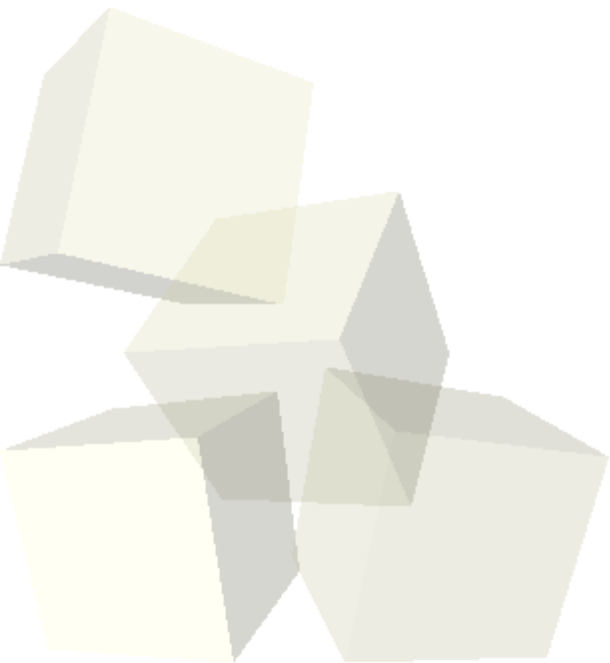
## ■ Ejemplo Mssql:





## 2.4.3 Características

- Permite crear probes personalizados como extensión de varias clases (AbstractProbe, AbstractVersionProbe, ...).
- No hay límites más allá de los impuestos por el lenguaje de programación (en este caso Java). Por ejemplo, raw sockets :(





## 2.4.3 Características

- Estructura de un probe
  - ♦ Métodos get/set de atributos básicos
    - Id, Family, Name, Target, Enabled ,...
  - ♦ Métodos sobre la configuración
    - Provides, Depends, Options
  - ♦ Comunicación con la KB
    - KbPut, kbGet, kbHas
  - ♦ Generación de informes
    - AddReport, setReport, isReportable
  - ♦ Y función con la lógica de ejecución
    - probe



# Ejemplo Probe

```
import java.sql.*;
import java.util.Vector;
import com.securitycentric.metacoretex.lib.AbstractDatabaseProbe;
import com.securitycentric.metacoretex.lib.ProbeException;

public class MssqlVersion extends AbstractDatabaseProbe {

    public MssqlVersion() {

        setName("Mssql Versions");
        setFamily("MS SQL");
        setVersion("$Id: MssqlTest.java,v 1.2 2005/09/21 22:44:11 $");
        setProbeld(102);
        setCopyright("Edge-Security");

        addDepends("mssql/connection");
        addProvides("mssql/users","Vector of strings for usernames");
        addProvides("mssql/hashtes","Vector of strings for password hashes");

        setReport("Mssql Versions:\n"+
            "=====\n");

        setHelp("This probe attempts to enumerate DB users and passwords. "+
            "It will only be capable of doing so if the JDBC Connection "+
            "stored in mssql/connection has sufficient privileges\n");
    }
}
```



# Ejemplo Probe

```
public void probe() throws ProbeException {  
    try {  
        if (! kbHas("mssql/connection")) return;  
  
        con = (Connection) kbGet("mssql/connection");  
        Vector users = new Vector();  
        Vector hashes = new Vector();  
  
        Statement stmt = con.createStatement();  
        res = stmt.executeQuery("xp_msver");  
  
        setReportable(true);  
  
        while (res.next()) {  
            addReport(res.getString("name") + "\t" + res.getString("Internal_value") + "\t" +  
res.getString("Character_value"));  
        }  
  
    } catch (SQLException se) {  
        throw(new ProbeException("Lack of Permission? " + se.getMessage()));  
    }  
}
```



- Metacoretex dispone de un Wizard para la creación de probes sin conocimientos de programación.

Probe Wizard

Probe Name

Probe JAVA Name

Probe Family

Probe Version

Probe Id :

Copyright Info

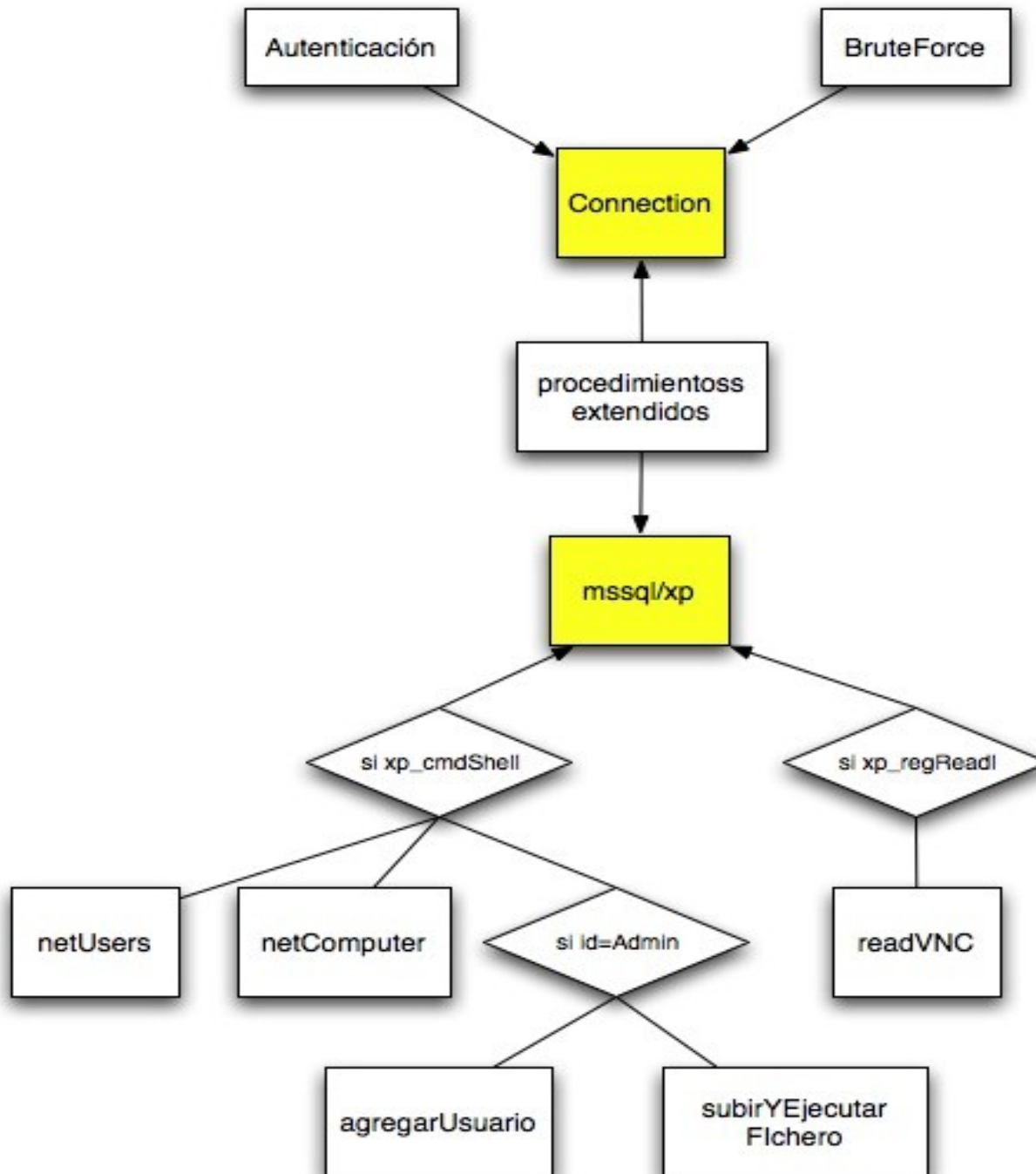


### ■ Modo pentest:

- ♦ 1- Identificar puertos
- ♦ 2- Identificar instancias de bases de datos
- ♦ 3- Intentar acceder a las mismas
  - a-Default accounts
  - b-Dictionary attack
- ♦ 4- Consultas específicas
  - Hashes de usuarios/contraseñas
  - Roles
  - Tablas de usuarios, de aplicación, etc.
- ♦ 5 -Acceso al sistema operativo (más allá de la BD)
  - Usuarios/contraseñas sistema operativo
  - Llaves de registros (win)
  - Agregar usuarios
  - Leer ficheros
  - Subir ficheros



# Diagrama PenTest



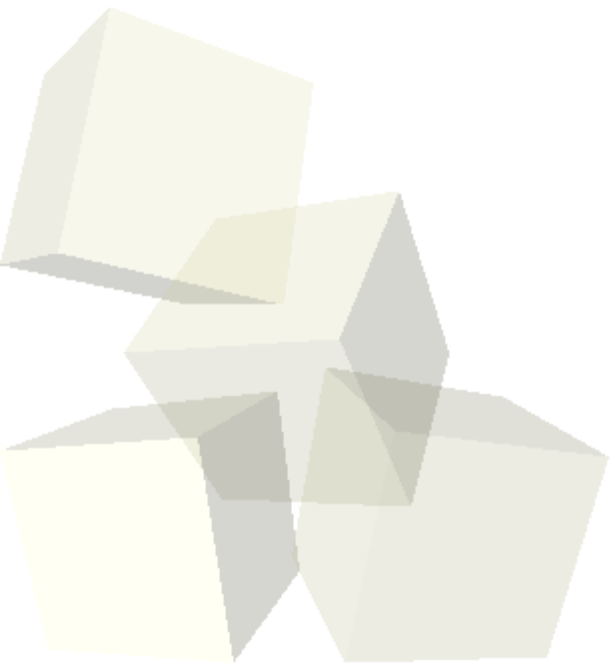


- En este apartado veremos el Metacoretex-NG en acción:
- Recorrido por la interface
- Ejemplo prácticos
- Reporte en HTML





- ¿Ruegos, preguntas, dudas?





- Web del proyecto nuevo

- ◆ <http://metacoretex-ng.sourceforge.net>

- Web original:

- ◆ <http://www.metacoretex.com>

- Edge-security:

- ◆ <http://www.edge-security.com>



Muchas gracias por su atención

