Con el crecimiento de Internet, y el desarrollo de sistemas de información bajo la arquitectura Cliente/Servidor, los sistemas de cómputo ,en general, están expuestos a múltiples amenazas, vulnerabilidades y ataques cada vez más complejos. Por lo tanto, es importante que las organizaciones implementen bitácoras (o archivos logs) para almacenar los sucesos que ocurren en el sistema. La información contenida en una bitácora es muy importante y útil cuando ocurre un incidente de seguridad o cuando se realiza una auditoría de sistemas.

Una bitácora puede registrar mucha información acerca de eventos relacionados con el sistema que la genera los cuales pueden ser:

* Fecha y hora.
* Host origen.
* Usuario.
* Actividad realizada.

La importancia de las bitácoras es la de recuperar información ante incidentes de seguridad, detección de comportamiento inusual, información para resolver problemas, evidencia legal, es de gran ayuda en las tareas de cómputo forense.

Enseguida plantearé un ejemplo de una bitácora desarrollada para la siguiente base de datos de MySQL, llamada **proyecto**, que tiene las tablas **carrera, departamento** y **maestros.**

**CREATE DATABASE proyecto;**

**USE proyecto**

**CREATE TABLE IF NOT EXISTS `carrera` (`clave\_carrera` int(11) NOT NULL, `nom\_carrera` varchar(20) NOT NULL, `num\_depto` int(11) NOT NULL, PRIMARY KEY (`clave\_carrera`), KEY `num\_depto` (`num\_depto`) ) ENGINE=InnoDB DEFAULT CHARSET=latin1;**

**CREATE TABLE IF NOT EXISTS `departamento` ( `num\_departamento` int(11) NOT NULL,`nombre\_dept` varchar(20) NOT NULL, `jefe\_num\_tarjet` int(11) NOT NULL, PRIMARY KEY (`num\_departamento`), KEY `jefe\_num\_tarjet` (`jefe\_num\_tarjet`) ) ENGINE=InnoDB DEFAULT CHARSET=latin1;**

**CREATE TABLE IF NOT EXISTS `maestros` (`num\_tarjeta` int(11) NOT NULL DEFAULT ‘0’,`nombre` varchar(50) DEFAULT NULL, PRIMARY KEY (`num\_tarjeta`)) ENGINE=InnoDB DEFAULT CHARSET=latin1;**

La estructura de la tabla **bitácora** sería la siguiente:

**CREATE TABLE IF NOT EXISTS `bitacora` (`id` int(11) NOT NULL AUTO\_INCREMENT, `operacion` varchar(10) DEFAULT NULL, `usuario` varchar(40) DEFAULT NULL, `host` varchar(30) NOT NULL, `modificado` datetime DEFAULT NULL, `tabla` varchar(40) NOT NULL, PRIMARY KEY (`id`) ) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO\_INCREMENT=1 ;**

La bitácora debe registrar todos los movimientos (insertar, eliminar y modificar) que se realicen en las tablas de la base de datos. Para lograr lo anterior es necesario crear un trigger para que se ejecute después de la operación de insertar, otro para después de eliminar y el último para después de modificar para cada una de las 3 tablas de la base de datos. Los nueve triggers necesarios para que funcione la bitácora son los siguientes:

**DROP TRIGGER IF EXISTS `bit\_carr\_ins`;  
DELIMITER //  
CREATE TRIGGER `bitacora` AFTER INSERT ON `carrera`  
FOR EACH ROW INSERT INTO bitacora(host, usuario, operacion, modificado, tabla) VALUES (SUBSTRING(USER(), (INSTR(USER(),’@’)+1)), SUBSTRING(USER(),1,(instr(user(),’@’)-1)), “INSERTAR”, NOW(), “CARRERA”)  
//**

DROP TRIGGER IF EXISTS `bit\_carr\_upd`;  
CREATE TRIGGER `bit\_carr\_upd` AFTER UPDATE ON `carrera`  
FOR EACH ROW INSERT INTO bitacora(host, usuario, operacion, modificado, tabla) VALUES (SUBSTRING(USER(), (INSTR(USER(),’@’)+1)), SUBSTRING(USER(),1,(instr(user(),’@’)-1)), “ACTUALIZAR”, NOW(), “CARRERA”)  
//

DROP TRIGGER IF EXISTS `bit\_carr\_del`;  
CREATE TRIGGER `bit\_carr\_del` AFTER DELETE ON `carrera`  
FOR EACH ROW INSERT INTO bitacora(host, usuario, operacion, modificado, tabla) VALUES (SUBSTRING(USER(), (INSTR(USER(),’@’)+1)), SUBSTRING(USER(),1,(instr(user(),’@’)-1)), “ELIMINAR”, NOW(), “CARRERA”)  
//

**DROP TRIGGER IF EXISTS `bit\_depto\_ins`;  
CREATE TRIGGER `bit\_depto\_ins` AFTER INSERT ON `departamento`  
FOR EACH ROW INSERT INTO bitacora(host, usuario, operacion, modificado, tabla) VALUES (SUBSTRING(USER(), (INSTR(USER(),’@’)+1)), SUBSTRING(USER(),1,(instr(user(),’@’)-1)), “INSERTAR”, NOW(), “DEPARTAMENTO”)  
//**

DROP TRIGGER IF EXISTS `bit\_depto\_upd`;  
CREATE TRIGGER `bit\_depto\_upd` AFTER UPDATE ON `departamento`  
FOR EACH ROW INSERT INTO bitacora(host, usuario, operacion, modificado, tabla) VALUES (SUBSTRING(USER(), (INSTR(USER(),’@’)+1)), SUBSTRING(USER(),1,(instr(user(),’@’)-1)), “ACTUALIZAR”, NOW(), “DEPARTAMENTO”)  
//

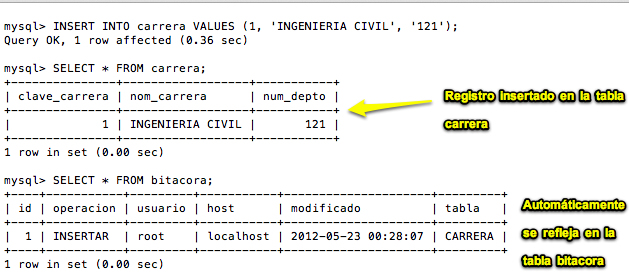
DROP TRIGGER IF EXISTS `bit\_depto\_del`;  
CREATE TRIGGER `bit\_depto\_del` AFTER DELETE ON `departamento`  
FOR EACH ROW INSERT INTO bitacora(host, usuario, operacion, modificado, tabla) VALUES (SUBSTRING(USER(), (INSTR(USER(),’@’)+1)), SUBSTRING(USER(),1,(instr(user(),’@’)-1)), “ELIMINAR”, NOW(), “DEPARTAMENTO”)  
//

**DROP TRIGGER IF EXISTS `bit\_mae\_ins`;  
CREATE TRIGGER `bit\_mae\_ins` AFTER INSERT ON `maestros`  
FOR EACH ROW INSERT INTO bitacora(host, usuario, operacion, modificado, tabla) VALUES (SUBSTRING(USER(), (INSTR(USER(),’@’)+1)), SUBSTRING(USER(),1,(instr(user(),’@’)-1)), “INSERTAR”, NOW(), “MAESTROS”)  
//**

DROP TRIGGER IF EXISTS `bit\_mae\_upd`;  
CREATE TRIGGER `bit\_mae\_upd` AFTER UPDATE ON `maestros`  
FOR EACH ROW INSERT INTO bitacora(host, usuario, operacion, modificado, tabla) VALUES (SUBSTRING(USER(), (INSTR(USER(),’@’)+1)), SUBSTRING(USER(),1,(instr(user(),’@’)-1)), “ACTUALIZAR”, NOW(), “MAESTROS”)  
//

DROP TRIGGER IF EXISTS `bit\_mae\_del`;  
CREATE TRIGGER `bit\_mae\_del` AFTER DELETE ON `maestros`  
FOR EACH ROW INSERT INTO bitacora(host, usuario, operacion, modificado, tabla) VALUES (SUBSTRING(USER(), (INSTR(USER(),’@’)+1)), SUBSTRING(USER(),1,(instr(user(),’@’)-1)), “ELIMINAR”, NOW(), “MAESTROS”)  
//

El resultado que se espera de la bitácora se muestra en la siguiente imagen.

**[](http://i2.wp.com/tavoberry.com/blog/wp-content/uploads/2010/06/bitacora1.jpg)**

Espero que puedan implementar sus propias bitacoras en todas  las bases de datos que implementen de ahora en adelante.