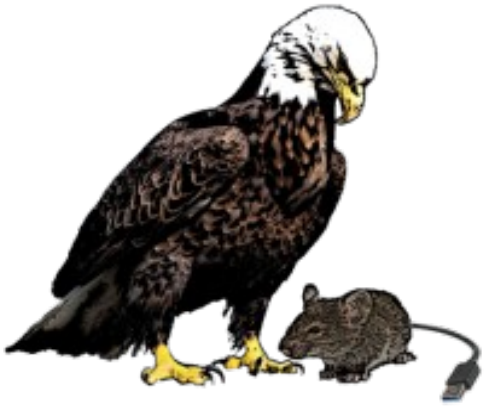
RAPTOR

Realtime Anti-Phishing Training Online Resource

Tame Your Mice



Raptor is a training tool to help educate the “weakest link” on how they can avoid being a doormat for hackers. By launching harmless phishing campaigns, tracking results, and providing immediate education, Raptor can help you tame your company's mice.

Installation

Raptor runs on the MVProc web platform. MVProc is an Apache module – source code and documentation can be found at: http://sourceforge.net/projects/mvproc/ To run Raptor on another webserver (like lighttpd), MVProc is also available as an FCGI, available here: http://sourceforge.net/projects/mvproc-fgci/

After installing MVProc and Raptor, configuration examples can be found in Raptor's **webserver/** directory. Please be aware that changing any of the database names, passwords, etc. must be reflected in the MVProc configuration.

Next, Raptor will need the emailer udf, located in Raptor's **db/udf/** directory. The udf requires libmysqlclient-dev, libesmtp and libesmtp-dev to be installed for compilation. Standard **make** and **make install** will need to be followed with **mysql -u root -p mysql < udf_emailer.sql**

Next is the setup of the databases and users for Raptor. The script to automate its setup can be found at **db/setup.sql** – if you want to change the names or passwords for the webserver's access users, just find the relevant part of setup.sql and edit prior to insertion, but remember that the MVProc configuration will have to be adjusted to match. Note that the raptorAdmin and raptorResponse users only have access to execute stored procedures, each in one database. This protects the data and functions in the raptorData database in case the webserver is ever compromised. It's the best way to run MVProc. So then:

mysql -u root -p < db/setup.sql
mysql -u root -p raptorData < db/data.sql
mysql -u root -p raptorData < db/views.sql
mysql -u root -p raptorData < db/functions.sql
mysql -u root -p raptorResponse < db/responseProc.sql
mysql -u root -p raptorAdmin < db/adminProcs.sql

Edit **db/adminLogin.sql** to change the username and password for the admin login credentials. (If more than one admin is desired just copy the SQL and edit.)
Now **mysql -u root -p raptorData < db/adminLogin.sql** and copy the **db/raptor.cnf** file into **/etc/mysql/conf.d/** or copy and paste its contents into **/etc/my.cnf**

(If you changed the names or passwords of the users earlier, you'll need to edit this file.)

How to use Raptor

There are three main entities to know for using Raptor: Clients, Phishes, and Campaigns.

Clients

In the database, clients are just an id and a name. Only one client is required to launch campaigns.

Phishes

A Phish has two main elements: an email and a response page.

Email

Raptor does search and replacement for the following tags in the email file:

__NAME__ - replaced by the name supplied in the uploaded list of emails for a campaign. More on that later.

EMAIL - replaced by the email address to which the phish will be sent.

DATE - replaced by a UTC timestamp.

HASH - this tag is the uri of the phishing link, but can also be used anywhere else in the phishing email. When the user clicks the "bad" link, they should be directed to the Response Site, where the visit will be logged for reporting, then the user will be redirected to an educational page.

MESSAGEID - replaced by a unique id to make mail servers happy.

There is a sample in the **phishes/** directory called **fedex_simple.txt** which started as an actual phishing email I found in my spam folder. This simple phish is a good template for what Raptor expects you to upload for the phishing email.

Response Page

Raptor will redirect a user who clicks a phishing link to any url you choose for education. The url can be relative or absolute.

It's a good idea to use the "Test Send Phish" page to verify that your phish is what it should be and also that the mail server receiving the phishes responds in a desirable way. Raptor will send the emails to any server on any port specified, but the most reliable function will come from sending them to a well-configured mail server locally (localhost:25). The *__NAME__* and ***HASH*** tags will be left unchanged when Test Sending.

Campaigns

A Campaign is a batch send of phishing emails. When launching a campaign, you'll upload a text file with the format: [email address],[Name of Recipient] on each line. There can be as many addressees as you wish. For each email successfully sent, an entry is registered in the database containing an MD5 sum of the email address (in case of compromise, the list of emails will not be retrievable), the response hash, and of course the id of the Campaign.

Email Response History

This is an easy way to look up what activity has occurred with an email address that's been included in any campaigns.

CSV

Throughout the administrative interface, there are "csv" links. These are for downloading raw csv data for the item identified in the row.

Raptor is an actively maintained project. Really!
Any questions, requests, issues... please post them to sourceforge.net at the Raptor page.