Scylla & 1.0 Alpha

(101% Colombiano)

http://www.2secure.org

Sergio Valderrama (flacman at cuteam dot org)
Carlos Rodriguez (iker at cuteam dot org)

Special thanks to: RPM (Our designer, and webshell creator), Zealot (for his help with charybdis), Tronador (he build pieces of mail modules)

Download: You would be able to download the source only (with compiling instructions) from here: http://code.google.com/p/scylla-v1/ (will be uploaded the 22 of July)

Abstract

When there's no technical vulnerability to exploit, you should try to hack what humans left for you, and believe me, this always works.

Scylla provides all the power of what a real audit, intrusion, exclusion and analysis tool needs, giving the possibility of scanning misconfiguration bugs dynamically.

Scylla aims to be a better tool for security auditors, extremely fast, designed based on real scenarios, developed by experienced coders and constructed with actual IT work methods.

The words "Configuration Tracer" are the best definition for Scylla, a tool to help on IT audits.

Introduction

This document is a reference manual about what Scylla is, and what its capabilities are. This document will show the user a hypothetical scenario that shows what he/she is able to do when Using Scylla and basic explanation of each one of its modules and its features.

Scylla is not solely meant to be an exploitation tool or a tool to discover vulnerabilities within applications, but rather as a method to hack and patch "human stupidity", such as common errors or flaws unintentionally put in service configuration. Scylla is built over an extremely fast and reliable core, with anti-anti Brute force techniques, error

recovery protocols, and a lot of speedup tricks with most manual (and other types of attacks unknown to the user) being coded to avoid repetitive tasks.

BTW, if you haven't read well, this is 1.0a version, and the "a" comes from "A lot of work to do", "A lot of bugs (I think)" and "A lot of testing left", and we will appreciate a lot your help ☺.

Objective

Scylla is a tool to audit different online application protocols and configurations, built over a brute-force core.

This tool acts at a tool for unifying auditing techniques, in other words, it does what oscanner, winfingerprint, Hydra, DirBuster, and other tools do, and also what those tools don't do.
Scylla is arguably the first free-open source auditing/hacking tool for protocols such as LDAP, DB2, Postgres, terminal and Mssql; Scylla adds tons of new features to what those other tools do but with a key difference: it does them faster and smarter!

Supported Protocols

- ✓ **Terminal (Telnet, SSH, telnets)**
- ✓ **FTP (FTPS, FTP, SFTP)**
- ✓ **SMB (Also Windows RPC)**
- ✓ **LDAP**
- ✓ **POP3 (POP3S)**
- ✓ **SMTP (SMTPS)**
- ✓ **IMAP**
- ✓ **MySql**
- ✓ **MSSQL**
- ✓ **Oracle (Database and TNS Listener)**
- ✓ **DB2 (Database and DAS)**
- ✓ **HTTP(HTTPS; Basic AUTH Brute Force, Digest AUTH Brute Force, Form Brute Force, Directory and files Brute Force)**
- ✓ **DNS (DNS snooping)**
- ✓ **Postgres SQL**
- ✓ **And more coming…**

How does Scylla work?

Scylla functions on three basic stages:

Pre-Hack Stage:

This stage is defined as what information Scylla can readily obtain without resorting to brute-force attacks (something like enumeration). Here is where anti-anti-Brute Force techniques are implemented, such as getting information on password policies, latency times, etc. Scylla is also obtaining extra information to make the attack: searching for protocol and service versions, verify null sessions, and system enumeration among other things. It also builds specially crafted lists (based on other lists.) When applicable, the AutoPWN modules (such as a "one click" web shell upload on a MySQL attack or opening a blind shell using MSSQL services without any previous information).

Brute Force Stage

Here is where Scylla shines. It is an extremely fast brute force core. For example when hydra makes 7.000 tries/min, Scylla makes over 22.000 tries/min over MSFTPd.

Post Hack Stage:

What can you do with a user-password combination? Simple stuff like fetching the /etc/shadow file or the FEAT response of an FTP server, or more complex stuff such as spawning a shell with just one MSSQL command (a OneClickOwnage paper implementation). It is more or less like Maintaining Access or Expanding Influences.

Charybdis

Charybdis is Scylla's counterpart. He's at the other side of the river.

What if you "pwnd" a Linux server (or even a windows server) and you can't get heavy tools or don't have GUI access to it (or simply, you are a *Nix user)? This is why Charybdis was built: To be at the other side waiting for Scylla.

It's simply a multi-platform high speed pipe between Scylla and whatever is on the other side. Supporting Scylla from basic "bounce" functionality to socks proxy connection, Charybdis is specially crafted to provide the best performance to the attacker.

Deep Documentation (what you should see)

Basic features:

- ✓ User, password list based Brute force
- ✓ Multiple hosts support
- ✓ Multiple session support
- ✓ Nmap integration
- ✓ Non-synchronized threads (proof to be a bit faster)
- ✓ Ability to restore sessions
- ✓ Session auto-saving (based on SQL Server CE)
- ✓ Easy to use
- ✓ Auto configured options
- ✓ Hacker oriented
- ✓ Free, and always free
- ✓ Database browser (who have hacked a DB and don't have a DB client to connect to it? And worse if you don't have internet)
- ✓ Open source tool

List creation

List creation is a component to create new lists based on existing dictionaries. The idea is to take each word in a specific list and compose different words based on it.

As-Is: Nothing special, just leave the dictionary just as it is.

Double: Duplicates the word. Cut – CutCut.

CasePerms: Creates every letter-Case permutation of the word. Cut – CuT, CUT, cuT, cut, CUt, etc.

Reverse: Reverse the word. CUTeam – meaTUC.

LowerCase: Adds the lower case version of the word. Cut – cut.

UpperCase: Adds the upper case version of the word. Cut – CUT.

H4x0r: Adds the word in "hackers-jargon" (replace each vocal for numbers except u, b for 8, t for 7, l for 1 and s for 5). CUTeam – CU734m.

H4x0rPermutation: Creates every H4x0r-Case permutation of the word. Cuteam – Cu7eam, Cu73am, Cu7e4m, Cute4m, etc.

Date ap/prepend: Adds the word with different years appended or pre pended (from 1985 to the actual year). CUT – 1985CUT, 2000CUT, CUT1990, CUT2010, etc.

2Number Append: Adds the word with 2 numbers (from 00 to 99) appended. Cut – Cut00, Cut 01… Cut99.

Scylla Modules

Most of the hacks mentioned here are configurable options, and the default options are options that will let you "auto-pwn" or the ones considered less intrusive or the most important for the author.

FTP

Pre-Hacks:

- ✓ If a user is blocked, gets the maximum number of tries until a user gets blocked (numTries) and for the next user just tries numTries passwords.
- ✓ SFTP – Get supported ciphers

Hack:

Built from scratch SFTP brute force module, FTP and FTPS brute force.

Post-Hacks:

- ✓ Fetch FEAT response.
- ✓ Fetch PWD response (actual directory).
- ✓ Fetch SYST answer (Operating system information)
- ✓ Check LIST, STOR, MKD, DELE and RMD permissions (list, upload, make directory, delete file and remove directory).
- ✓ Basic Directory Transversal hacks
    - /../../../../../../etc/shadow
    - \..\..\..\..\..\..\config.sys

Terminal

Pre-Hacks:

- ✓ If a user is blocked, gets the maximum number of tries until a user gets blocked (numTries) and for the next user just tries numTries passwords.
- ✓ Process "connection limit" answers and wait 20 milliseconds until next try.
- ✓ Process MSTelnetd when user+password are correct but the user isn't in the TelnetClients group.
- ✓ SSH2 – Get and Set supported ciphers

Hack:

Built from scratch SSH brute force module, implemented as fast as possible in the login process (C++), Telnet, Telnets.

More servers supported (this makes it a bit slower…).

Post-Hacks:

- ✓ Fetch CD response.
- ✓ Fetch SUDO capabilities response.
- ✓ Ncat (or putty) integration
- ✓ Fetch /etc/shadow and /etc/passwd

POP3

Pre-Hacks:

- ✓ Verify authentication types supported by server
- ✓ Verify if APOP authentication is available (and use it if so)

Hack:

POP3, POP3S, Auth-login, Auth-plan Auth-md5

Post-Hacks:

- ✓ Retrieve first 10 e-mail headers
- ✓ Get number of messages in the account
- ✓ Get e-mail addresses used in mails received

SMTP

Pre-Hacks:

- ✓ VRFY brute force pre-attack (tries to get only valid users)
- ✓ Anonymous login
- ✓ Verify authentication types supported by server

Hack:

SMTP, SMTPS, Auth-login, Auth-plan Auth-md5

Post-Hacks:

- ✓ Try sending a mail to root

- ✓ Mail relay (tries to send from [attacker@cuteam.org and attacker@ specified_IP_or_URL] to [Your_mail@any_domain.com and pick_a_mail@specified_IP_or_URL])

MSSQL:

MSSQL has 2 modalities: FastAttack (really fast, raw brute force) and Normal (Using SQLClient). The difference is that SQLClient is safer, it has a better error management and has more pre-hacks making it a bit more intelligent; use it to avoid blocking accounts or stuff like that. Also, most post-hacks use SQLClient. If a hack is available only for SQLClient it would be marked as SC.

Pre-Hacks:

- ✓ SC: If a password must be changed it prompts a dialog for you to change it if you want.
- ✓ SC: If max users connection limit reached, wait 100 ms until next try (with the same thread).
- ✓ SC: If User+Password found but there is an error. Marks the user+password as found and displays the error.
- ✓ SC: If user is blocked, tries for next user.
- ✓ SC: Test for SSPI (actual Windows user authentication)
- ✓ SC: Specify System version type (SQLServer 2k, 2k5, 2k8 or lastest)
- ✓ SC: Specify Local Machine Name
- ✓ SC: Specify database to connect
- ✓ Try SA user with null password

Hacks:

SQLClient and raw brute force. SSL Support.

Post-Hacks:

- ✓ Open UI for command execution. Opens a basic GUI to execute commands. Saves the command log in the Report Database (see report section). If don't have enough permissions to execute commands, it tries to hack it using: sp_configure 'show advanced options', 1; RECONFIGURE; sp_configure 'xp_cmdshell', 1; RECONFIGURE
- ✓ One click ownage hack. Execute any payload you define (default is TX shell), just as specified in http://ferruh.mavituna.com/papers/oneclickownage.pdf.
- ✓ Show Databases the user can access
- ✓ Fetch Users Info, including: Usernames, SID, Password Hash, Creation date, is disabled and default database name.
- ✓ Open Scylla DB Browser

MySQL (MySQUAL in honor to SENA, Colombia xD)

This module uses MySQL.Data.dll or ODBC (no support available) to connect to the remote host. A "raw" and faster version will be also implemented with limited pre-hacks.

Pre-Hacks:

- ✓ If max users connection limit reached, wait 100 ms until next try (with the same thread).
- ✓ If received message "password to long must be hex", just try the passwords that meet: passLen LESSOREQUALTHAN #password(length received in the error) AND !password.haveDigits
- ✓ Just try passwords of less than 16 characters (mysql don't support more)
- ✓ If want to use SSL certificates or a special SSL cipher connection, it would use ODBC with the specified options. Also, an auto-signed certificate is provided.

Hacks:

SSL support, specially crafted SSL configuration, certificate based SSL

Post-Hacks:

- ✓ Fetch databases that can be accessed by the user.
- ✓ Fetch users profile, including: Host, User name, Password hash, Select_priv, Insert_priv, Update_priv, Delete_priv, Create_priv, Drop_priv, Reload_priv, Shutdown_priv, Process_priv, File_priv, Grant_priv, References_priv, Index_priv, Alter_priv.
- ✓ If there is a http server, try to upload a web based PHP shell to it (A specially basic auto-destroyable shell, or the famous C99).
- ✓ Execute server commands (via UDF)
- ✓ Open Scylla DB Browser

DB2

Pre-Hacks:

- ✓ Obtain DAS information (server database access profile)
- ✓ - User-ID auth only - brute force
- ✓ Fetch EXCSAT and other packet responses (used to Auto Configure the Hack phase and give additional info to the user).
- ✓ Host less than 18 characters accepted

Hack:

SSL support (if applicable), encrypted auth.

Post-Hacks:

- ✓ List all tables (list tables for all)
- ✓ List tables for specific users (select name, creator from systables order by name)
- ✓ Security policies check (select * from syssecuritypolicies)
- ✓ Audit policies check (select * from sysauditpolicies)
- ✓ Fetch Roles and Role authorizations
- ✓ Fetch for users authorizations (select grantee, tableschema, tablename from sysuserauth)
- ✓ Fetch users and users privileges

ORACLE

This module uses Ora.Net provider for database connection. TNSListener module is built as a partner of Oracle module.

Pre-Hacks

- ✓ Fetch SID
- ✓ SID Brute force
- ✓ TNS version detection
- ✓ Allow the user to specify a SID (obligatory if no SID could be fetched or guessed, if no, Scylla would use ORCL)
- ✓ Try to use over 500 default users-passwords before the real brute force
- ✓ Fetch Blocked accounts

Hack:

If user must connect as SYSOPER or SYSDBA, tell the user and append SYSDBA to the connection string for post-hacks.

Post-Hacks:

- ✓ Fetch usernames and user information
- ✓ Fetch users access dates
- ✓ Fetch new and old password hashes
- ✓ Fetch database names the user can see
- ✓ Fetch Policies
- ✓ Fetch Roles and Role information
- ✓ Fetch Links (useful to find clear-text passwords and other interesting info)
- ✓ Open Scylla DB Browser

SMB

The trick here is using the windows API that is actually faster than SAMBA. This module is not just about SMB, but windows RPC.

Pre-Hacks:

- ✓ Try for null or anonymous sessions.
- ✓ Try to fetch password policy and adjust the hack phase settings to avoid blocking users and stuff like that.
- ✓ If operating system just accepts LM authentication, remove all password of length greater than 14 from the password list.

Hack:

NT, LM, NTLMv2, all what WNetAddConnection3 supports

Post-Hacks:

- ✓ fgDump wrapper (get password hashes)
- ✓ Fetch Users
- ✓ Fetch groups (relation user-groups relation)
- ✓ Fetch OS Version
- ✓ Fetch RPC Binds
- ✓ Fetch network Adapters
- ✓ Fetch Disks and shares
- ✓ Fetch active sessions
- ✓ Fetch Event log
- ✓ System Date and Time
- ✓ Fetch patch level
- ✓ Via Active directory:
  - GetShares (directories)
  - GetGroups
  - Get Operating system version
  - Get Users

HTTP

This module is a bit different; it is divided into three sub-modules:

HTTP-Basic Auth: Where the only real pre-hack is to fetch the authentication type supported in basic-auth module (and auto-configure brute force hack depending in it). It supports Digest (using MD5) and basic auth.

HTTP-Form: This is like other brute-forcers but it is a bit more intuitive. For the next release (hope for this one) a new Charybdis module will be built to auto-configure brute force parameters depending on user navigation.

HTTP-Dir/File Brute Force: Tries to find hidden directories/files based on brute force. Also, this module maps the entire webpage to find its entire structure, based on HEAD commands for brute force and GET for web mapping. The 3xx response, searches in the location parameter. A bit of an intelligent modification, it doesn't show the user an apparently found file (from web mapping) if it doesn't receive a 200 or 403 response. It cuts down on the number of false positives and, like every Scylla module, "error proof".

Postgres

This module uses NPgsql.dll.

Pre-Hacks:

Try admin-admin user-password combination

Hacks:

SSL support, crypt, password, md5 and others supported by NPgsql.dll

Post-Hacks:

- ✓ Fetch databases that can be accessed by the user.
- ✓ Fetch user's profile (pg_shadow, pg_user, pg_group, etc.)
- ✓ Open Scylla DB Browser

LDAP

Ldap Query tool

Pre-Hacks:

Try null password
Try Anonymous Auth

Hacks:

SSL support

Post-Hacks:

- ✓ Fetch Users info
- ✓ Fetch Groups
- ✓ Fetch Computer info

DNS Snooping

Pre-Hacks:

- ✓ Try to see if the server is vulnerable by querying the server for common names

Hacks:

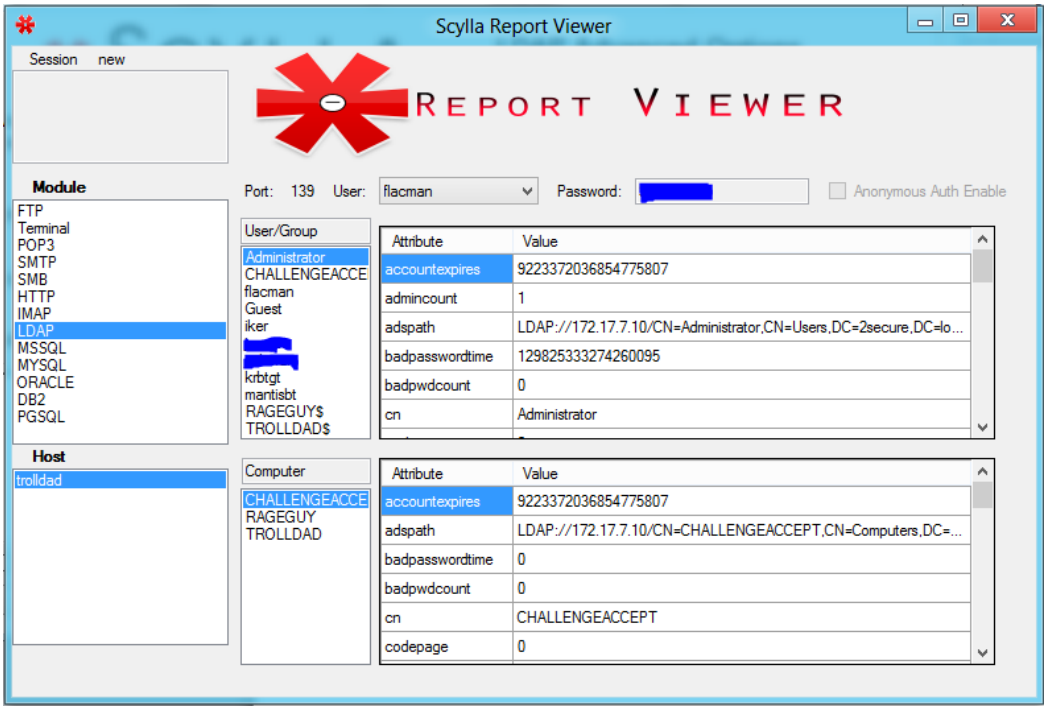
SSL support

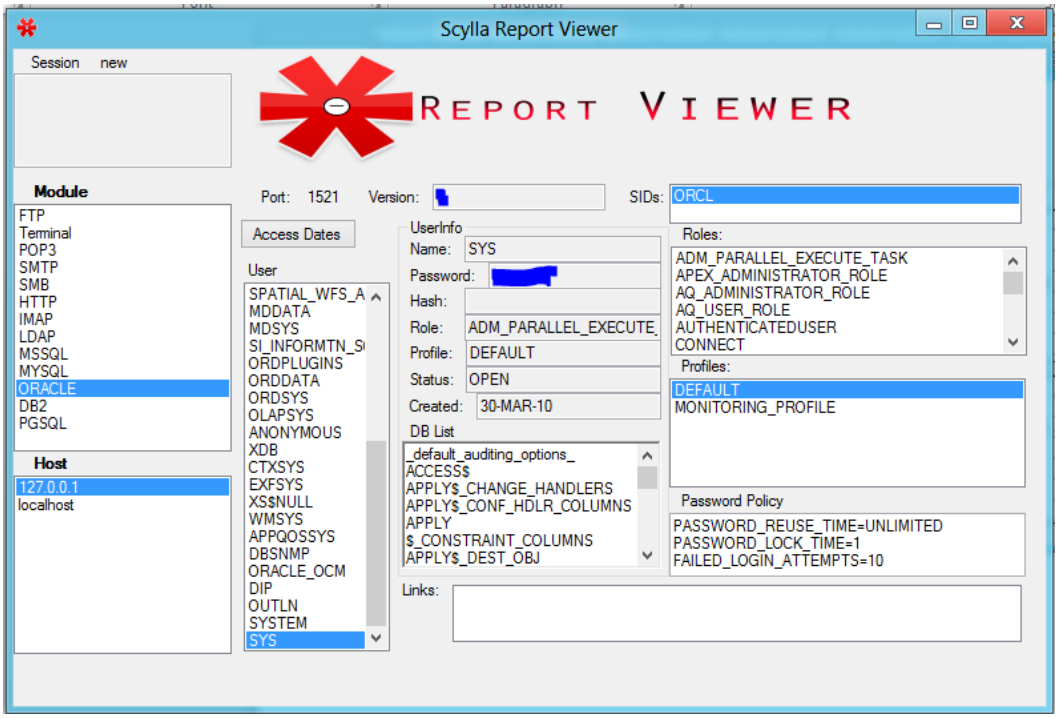
Post-Hacks:

- ✓ Fetch Answers
- ✓ Fetch Name Servers
- ✓ Fetch Additional info
- ✓ Determine if it's an authoritative server

Report Module

Every result the modules throw are stored in a SQLCE database, so your session information won't get lost. A report viewer was built so you can see the information easily.





THE FUTURE

This took eight months of work, just for DEF CON 20. Now try to imagine the future of this tool. We will work, primarily to try to make it faster and more accurate. There are other modules planed like SVN, CVS, RSH, RDP, and more. And at last we will be adding hacks, tons of hacks, we'll try to make it a more complete tool of this kind.

There are also plans to synchronize Scylla with other tools like MSF and Nessus.

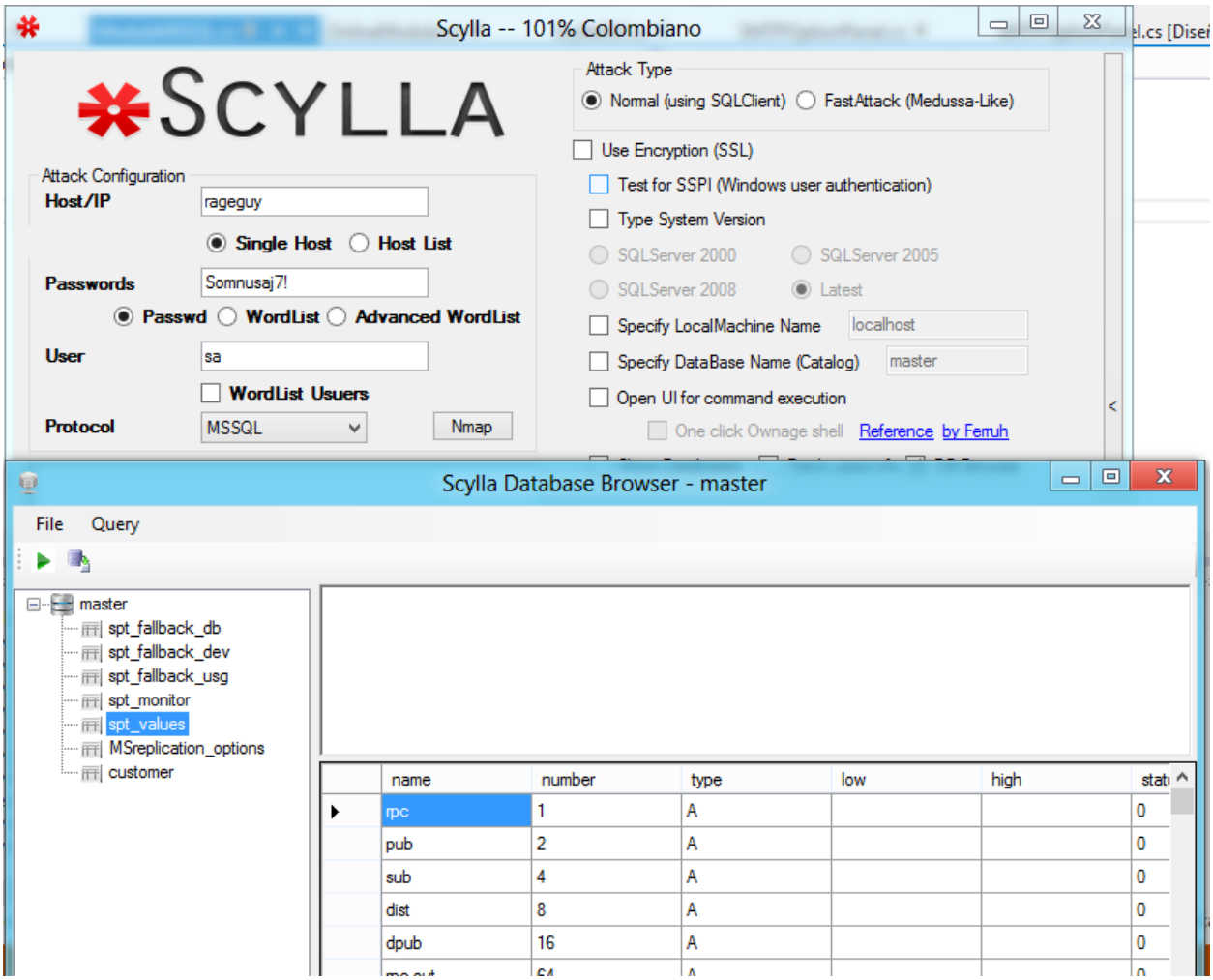
Our principal objective is to give as most capabilities to the users, and still be a "hacker-oriented" tool. We are conscious that there is no "wonder tool for everything" and that real hacking is more of a manual process, and that all we need is information, and possibly direction.

CONCLUSION

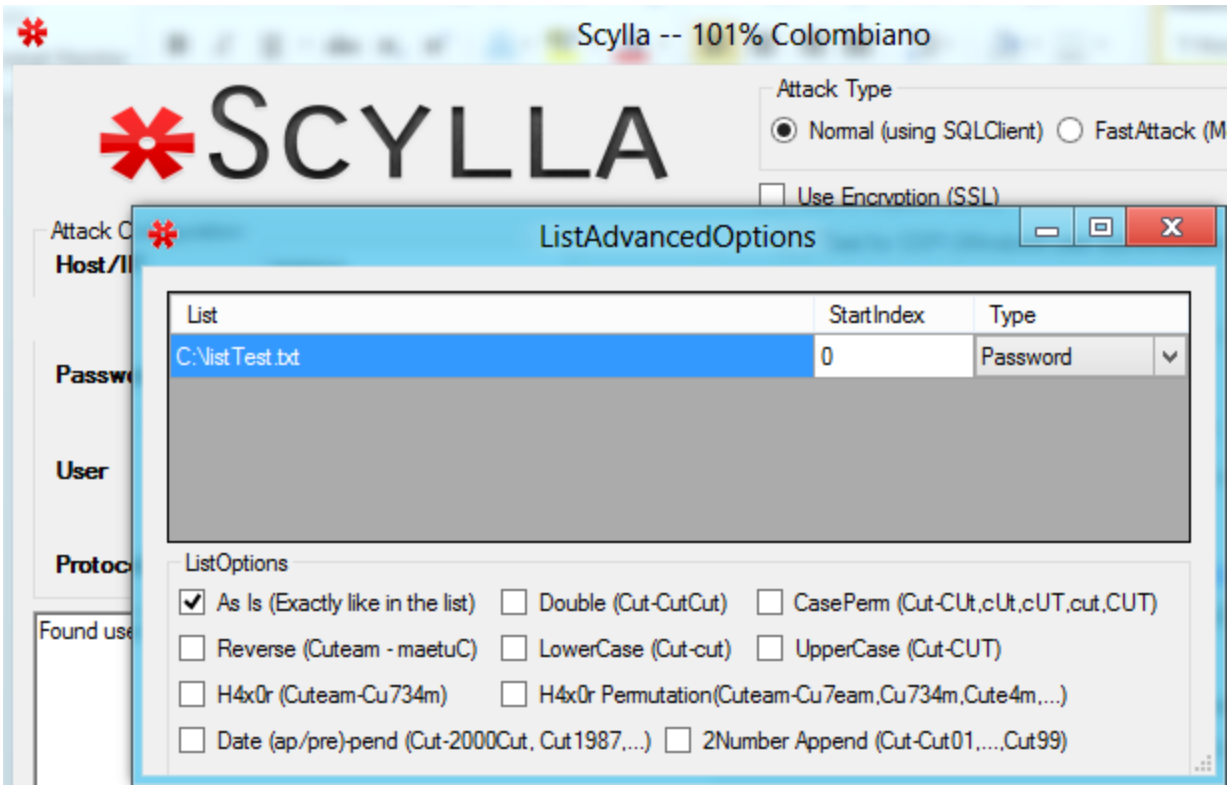
Perhaps I'm not the best qualified person to be writing something here, you have the documentation and you can try this tool, so you can make your own conclusions. There is one last thing to be said: I'm not intending to say that other tools are "worse" (they might be better than Scylla) just that maybe I got more free time. Every referenced tool here is a master piece (If you have got some time you should please check them out.), I thank the authors for building them and give people like me tools to work, and even better, Inspiration!

You could be sure of something… There is more coming soon!

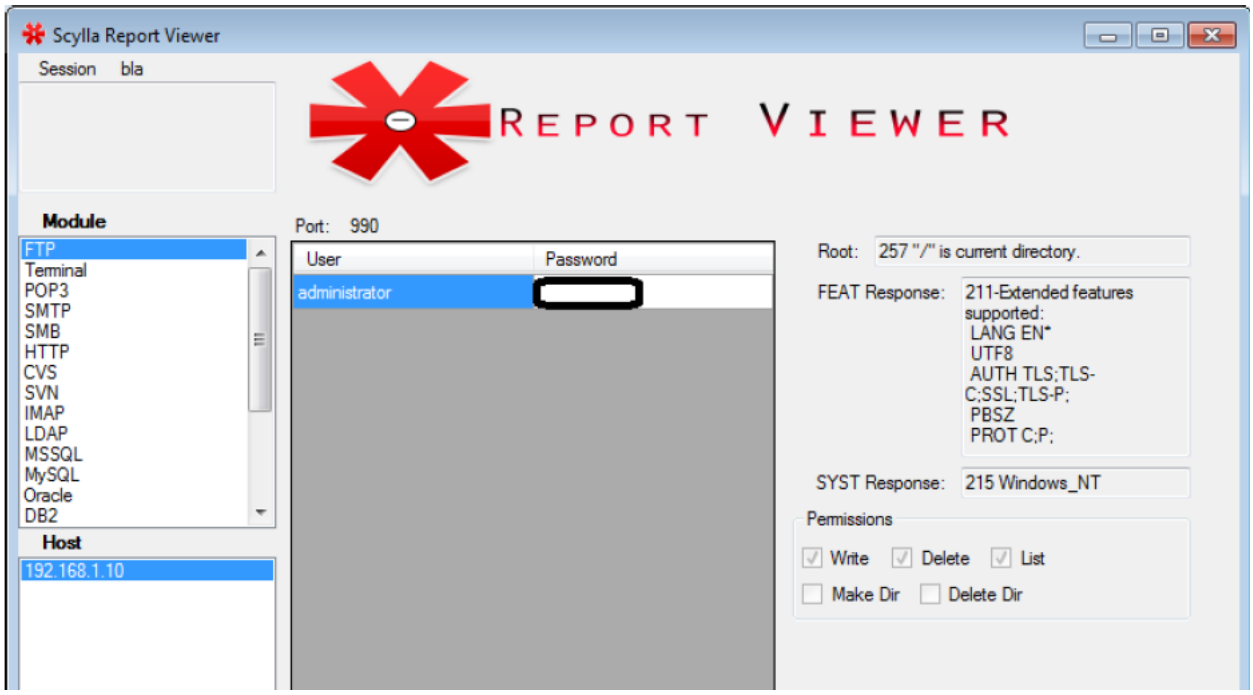
More pics :D



Scylla DBBrowser over MSSQL

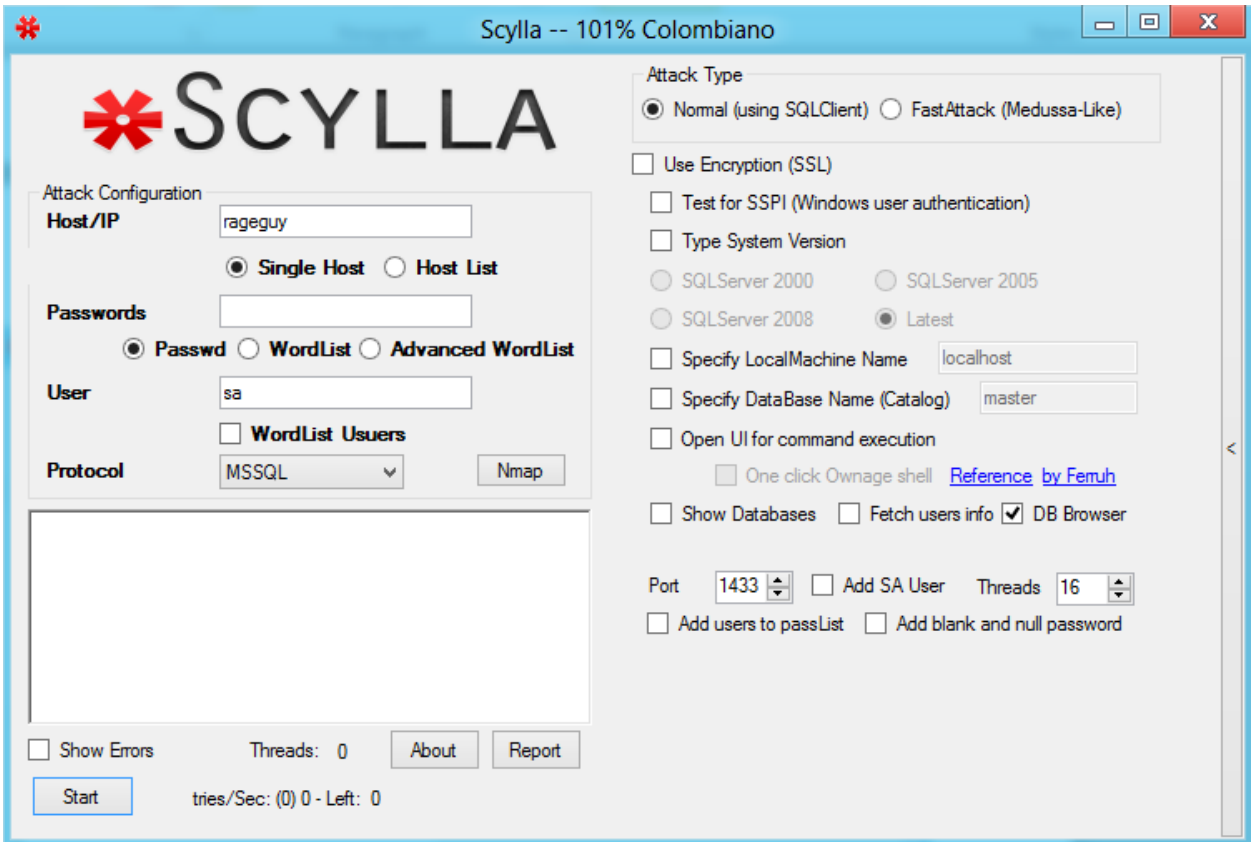


Advanced List Options

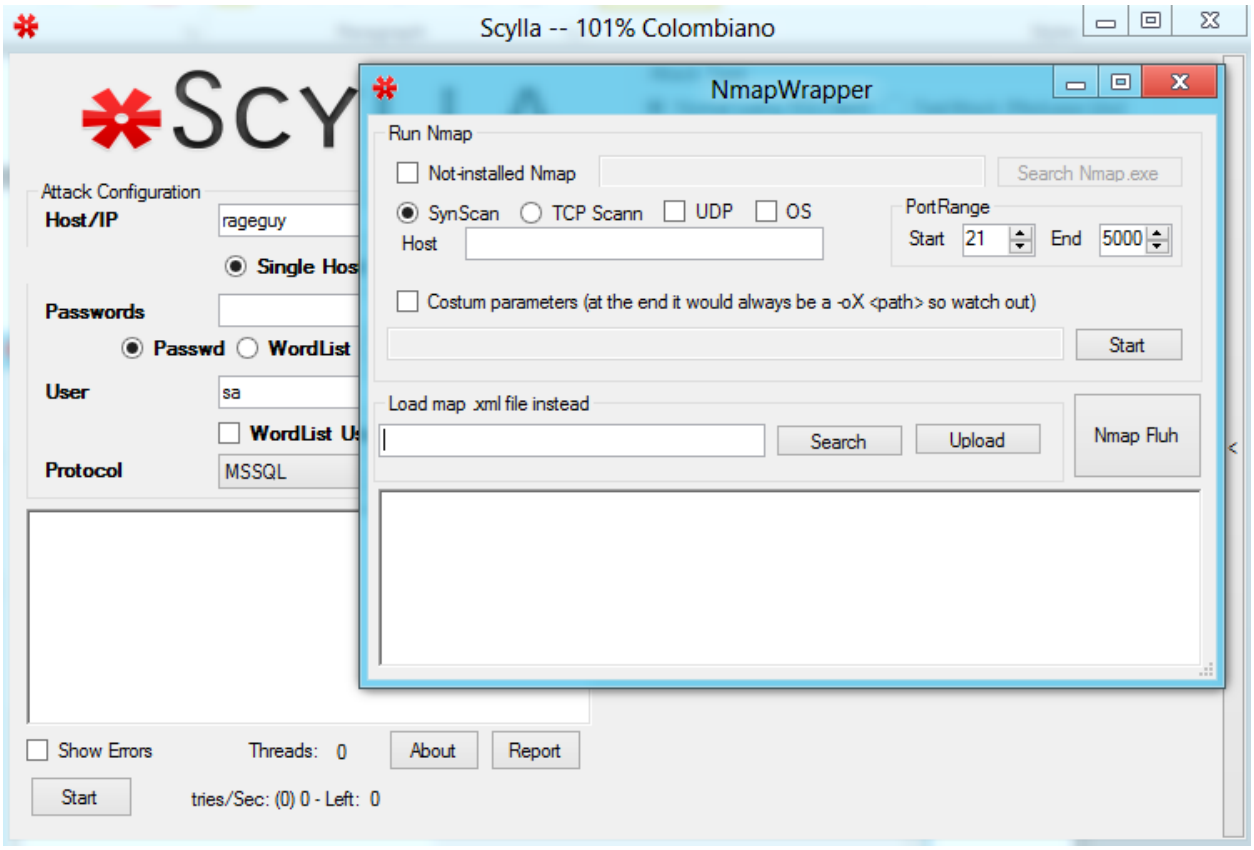


FTP Report Module

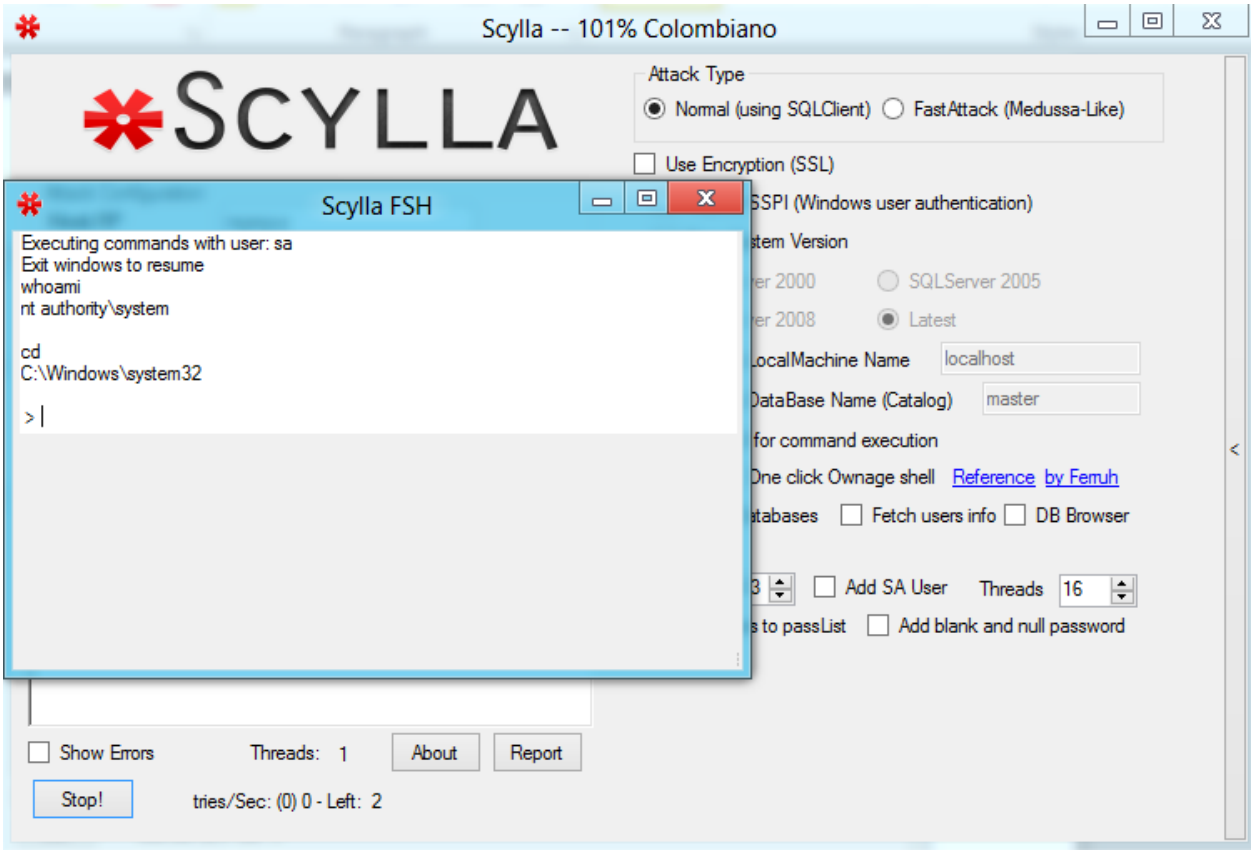
Scylla main GUI



MSSQL Advanced options



Nmap Wrapper



FSH over MSSQL