

GSoC - Fun & Profit Combined - The Nmap/Ncrack paradigm

ithilgore

sock-raw.org

26 Nov 2009

Google What?

- ▶ Annual program held & sponsored by Google
- ▶ For students only
- ▶ Open Source projects only
- ▶ Work from ~
- ▶ Mentor - Student approach

Some Numbers

- ▶ 180 Open Source organizations
- ▶ 1000 Students
- ▶ \$4500 stipends
- ▶ 3-4 months, summer period
- ▶ >14k lines of code! (personal number)

Selection Criteria

- ▶ A *good* application form
- ▶ Previous experience with the project
- ▶ A feature-rich CV always helps
- ▶ General open source involvement
- ▶ Passion

How does it work?

1. Write the application form. If you get selected then...
2. get acquainted with mentor, developers and codebase during the community bonding period.
3. Brainstorm. Start coding. Report on your status. Discuss problems with others.
4. ??????
5. PROFIT (& fun)

The Nmap paradigm

- ▶ Weekly reports on project status and goals achieved
- ▶ Weekly meetings with mentor (Fyodor) on IM
- ▶ `nmap-dev@seclists.org` as the main communication channel
- ▶ Feedback from both users and developers.

The goal: Ncrack

Ncrack is designed to be a fast and flexible network authentication cracker. You can point it at a service (ssh, msrpc, http, imap, pop3, SNMP, telnet, ftp, etc.) and it will make repeated authentication attempts. The goal is, of course, to find working credentials by brute force. It is a very handy tool to have during pen-tests, as many/most users still choose weak passwords.

<http://seclists.org/nmap-dev/2009/q2/238>
RFC on Ncrack, A new network authentication cracker

Why?

- ▶ Weak passwords more common than actual exploits.
- ▶ Competitors (THC-Hydra, Medusa etc):
 - ▶ are not very actively maintained
 - ▶ some of them are way too old
 - ▶ many limitations (e.g not good support for multiple hosts/multiple ports)
 - ▶ interfaces too restricted (nothing like Nmap which provides a dozen ways of specifying IP addresses for example)
 - ▶ no support for contemporary protocols like IPv6
 - ▶ not portable to many platforms (i.e Windows)
- ▶ Because we can.

Starting from scratch

1. Initial Draft
(http://sock-raw.org/nmap-ncrack/ncrack_draft.html)
2. Notes on underlying libraries (Nsock, Nbase)
3. Read through Nmap code to get ideas (ServiceScan)
4. Compiled first rough codebase with libs and all. Makefile Hell.
5. Experimented with Nsock and its API.

Coding the first .text

- ▶ Command-line interface. A lot of discussion [here](#).
- ▶ First parts of core engine - experimental stage.
- ▶ FTP module - something easy for starters.
- ▶ Experimenting against testbed servers.

Ncrack Architecture - Overview

Ncrack is based on a modularized architecture, where each protocol/service corresponds to the equivalent module that handles all the authentication steps.

Ncrack's architecture is thus built in a way so that a module is separated as much as possible from the more low level details of timing and connection management which are handled by the core engine.

http://sock-raw.org/nmap-ncrack/ncrack_engine.html

A brief overview of Ncrack's architecture

Ncrack Architecture - Main constituents

- ▶ *Core Engine*: Nsock event-driven API
- ▶ *ServiceGroup*: main object, services lists, general info
- ▶ *Service*: main class holding timing/statistical information, user options associated with specific service and a pointer to Target
- ▶ *Target*: stripped Nmap class holding IP address and/or hostname
- ▶ *Connection*: class describing current attack session
- ▶ *module state machine*: mechanism which describes the authentication phase of each module - modules have their own unique struct of info (void * technique)

Ncrack Architecture - Core Engine

The Ncrack brain (`ncrack.cc`):

- ▶ registers Nsock callback handlers and calls modules
- ▶ handles connection and authentication endings
- ▶ takes care of timing and adapts to dynamic conditions
- ▶ essentially is responsible for everything network related

What makes it fast?

Speed is gained through the ability to adapt to network conditions and leverage of clever techniques in the application protocol layer. Some of the speed factors:

- ▶ Nsock - provides a fast and optimized event-driven API
- ▶ opensshlib - custom library based on OpenSSH code and providing low-level access to SSH handlers
- ▶ HTTP pipelining - (through HTTP persistent connections) massive speedup
- ▶ Dynamic Timing engine - flexible adaptation to changing network conditions (e.g lower amount of parallel probes when something goes wrong)

What makes it better?

- ▶ Total control - from timing restrictions and network aggressiveness to credential iteration scheme.
- ▶ Support for Nmap notation in host/service specification + advanced extension for more complex scenarios.
- ▶ Target input from Nmap and automatic service parsing (-iX)
- ▶ Ability to save and resume a cracking session (--resume)
- ▶ Platform portability: Linux, Windows, Mac OS X, *BSD
- ▶ Ships in with high-quality username/password lists.
- ▶ Extensive documentation
(see <http://nmap.org/ncrack/man.html>)

Ncrack options 1/2

Ncrack 0.01ALPHA (<http://ncrack.org>)

Usage: ncrack [Options] {target and service specification}

TARGET SPECIFICATION:

Can pass hostnames, IP addresses, networks, etc.

Ex: scanme.nmap.org, microsoft.com/24, 192.168.0.1; 10.0.0-255.1-254

-iX <inputfilename>: Input from Nmap's -oX XML output format

-iN <inputfilename>: Input from Nmap's -oN Normal output format

-iL <inputfilename>: Input from list of hosts/networks

--exclude <host1[,host2][,host3],...>: Exclude hosts/networks

--excludefile <exclude_file>: Exclude list from file

SERVICE SPECIFICATION:

Can pass target specific services in <service>://target (standard) notation or using -p which will be applied to all hosts in non-standard notation.

Service arguments can be specified to be host-specific, type of service-specific

(-m) or global (-g). Ex: ssh://10.0.0.10,at=10,cl=30 -m ssh:at=50 -g cd=3000

Ex2: ncrack -p ssh,ftp:3500,25 10.0.0.10 scanme.nmap.org google.com:80,ssl

-p <service-list>: services will be applied to all non-standard notation hosts

-m <service>:<options>: options will be applied to all services of this type

-g <options>: options will be applied to every service globally

Misc options:

ssl: enable SSL over this service

path <name>: used in modules like HTTP ('=' needs escaping if used)

TIMING AND PERFORMANCE:

Options which take <time> are in milliseconds, unless you append 's' (seconds), 'm' (minutes), or 'h' (hours) to the value (e.g. 30m).

Service-specific options:

cl (min connection limit): minimum number of concurrent parallel connections

CL (max connection limit): maximum number of concurrent parallel connections

at (authentication tries): authentication attempts per connection

Ncrack options 2/2

cd (connection delay): delay <time> between each connection initiation
cr (connection retries): caps number of service connection attempts
to (time-out): maximum cracking <time> for service, regardless of success so far
-T<0-5>: Set timing template (higher is faster)
--connection-limit <number>: threshold for total concurrent connections

AUTHENTICATION:

- U <filename>: username file
- P <filename>: password file
- user <username_list>: comma-separated username list
- pass <username_list>: comma-separated password list
- passwords-first: Iterate password list for each username. Default is opposite.

OUTPUT:

- oN/-oX <file>: Output scan in normal and XML format, respectively, to the given filename.
- oA <basename>: Output in the two major formats at once
- v: Increase verbosity level (use twice or more for greater effect)
- d[level]: Set or increase debugging level (Up to 10 is meaningful)
- nsock-trace <level>: Set nsock trace level (Valid range: 0 - 10)
- log-errors: Log errors/warnings to the normal-format output file
- append-output: Append to rather than clobber specified output files

MISC:

- 6: Enable IPv6 cracking
- sL or --list: only list hosts and services
- datadir <dirname>: Specify custom Ncrack data file location
- V: Print version number
- h: Print this help summary page.

Ncrack ASCII art!

```

~00000
00000000
,000$ 0$+~
$=0= .0+0
000 000
:000 0~0
0000. 0
00000 .
.000000
0?= +. ,.
,?00.$000
00000~.:~0
.$+00~?~000
:00000.=0000
?00?00+=: ,0,
00000..0000~ 000000. $0
00..0~0?0::00,?0::?$0. 00 ~
.0. ,0?00000.0$,+,000.00 $00
0. 00.?00=00000~0+0:0000?0,~0?.
.0 +00 0+0000 0000=?~0000?00 00
.: ~~ .000=00000~00=000000+0.0~0$$$.
00 , ?00.. 000~0000000000000000.:0.0:0~ 0$00.+
00.0 00 00?~000~0000000000+00 + ~0000000000=$0000
$ 00 00. .00,0000000000000$00000. .0000+$+~00
0 00 .0 0000000000?~0000000. 0. .0$000000+$0
0 0 0 000:$~0000=0.0000,$. 00 0000000000
0 00 ?0000 $0 0 . .0000
. $ ?000. 0 0
0 +~?000
0. :000000?0 |-----[ Ncrack ]-----|
0000$?+00
00+0:~0$0+
.0$000?00
0?000000
.000~0
```



Demonstration time

Resources

<http://ncrack.org> OR <http://nmap.org/ncrack>
<http://nmap.org/ncrack/man.html>

```
svn co --username guest --password "" svn://svn.insecure.org/ncrack
```

<http://sock-raw.org>
<http://sock-raw.org/nmap-ncrack.html>
<http://sock-raw.org/nmap-ncrack/reports09.html>

Questions?

