BackTrack John The Ripper MPI Instant Cluster

BackTrack Development Team

muts [at] remote-exploit [dot] org

Up and running with an Instant Cluster

Table of Contents

BackTrack JTR MPI Cluster Edition

Foreword

Before we begin, I'd like to stress that I am no expert in either MPICH not Linux.

This project was completed in several days, where in the end, I needed something functional, not beautiful. The cluster design is far from secure, and should only be used in sterile environments.

Description

BackTrack JTR MPI Edition (BTJTRMPI) is an extension of the BackTrack Live CD which is able to PXE boot a cluster of machines which participate in a JTR cracking session. The process is still not fully automated, and requires a bit of user intervention.

History

In more than one occasion I found myself in the need of a password cracking cluster. The traditional methods such as djohn and bob the butcher were not stable enough for reliable cracking, and Cecilia is limited in the hash types it accepts. The JTR Openmosix trick was also very limited, as it could only run on dictionary files.

Not long ago I noticed the JTR MPI patch by John Anderson, and I thought I'd give it a shot. After much trial and more error, I managed to get JTR MPI running, and successfully integrated it into a small footprint Slax image.

The final goal would be to use a bootable CD in a classroom environment, with 30-40 machines. It suddenly dawned on me, that to accomplish this, I'd need 30-40 Cd's with the image on them - Enter PXE.

BackTrack Setup

In order to get the cluster up and running, we need to add several external modules to the **optional** directory inside the CD ISO file. This can be easily done on windows, using the MySlax Creator, or using ISO editing software like UltraISO.

The four modules needed are:

- pxe.mo   - includes the DHCP, NFS, TFTP servers needed.
- client.mo   - includes the client utilities and configurations.
- server.mo   - includes the server utilities and configurations.
- john-mpi.mo – includes JTR MPI binaries.

These files can be downloaded from:

http://www.offensive-security.com/downloads.html

Offering Professional Offensive Security

Please NOTE!

BackTrack has issues booting on Dual Core machines. By default, BackTrack will use a single CPU. To attempt to use both CPU's on a dual core system, add the letter "d" to any boot parameter. For example:

bt load=server|pxe|john	**(for single CPU)**

would change into:

dbt load=server|pxe|john	**(for Dual Core)**

If you get nasty SQUASHFS errors, your machine will not run BackTrack with both CPUs.

Booting Backtrack

BIG FAT HAIRY WARNING

PLEASE MAKE SURE YOU TURN OFF ANY DHCP SERVERS PRESENT ON YOUR SUBNET.

To start with, you'll probably want to boot the Master Cluster Server. You can do this by booting BackTrack with the following argument:

bt load=server|pxe|john-mpi

This will load both the PXE module, the Cluster Server module and the john binary. After a while, you will be prompted for a hostname and a Cluster Key (password) for the server.

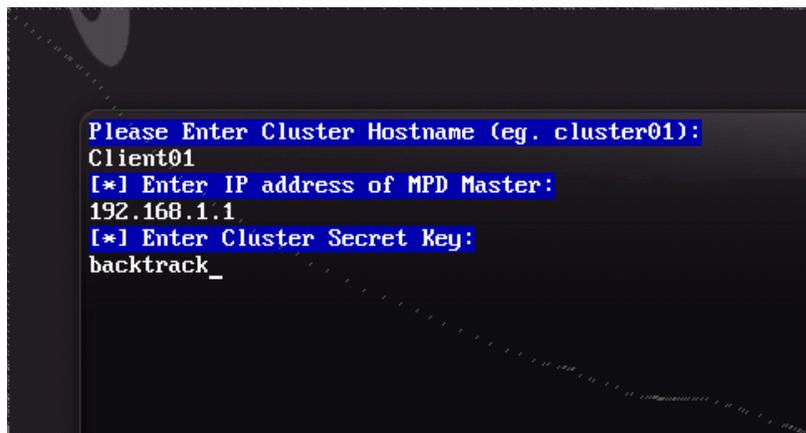I used the name hostname "Server01" and "backtrack" as the cluster key :
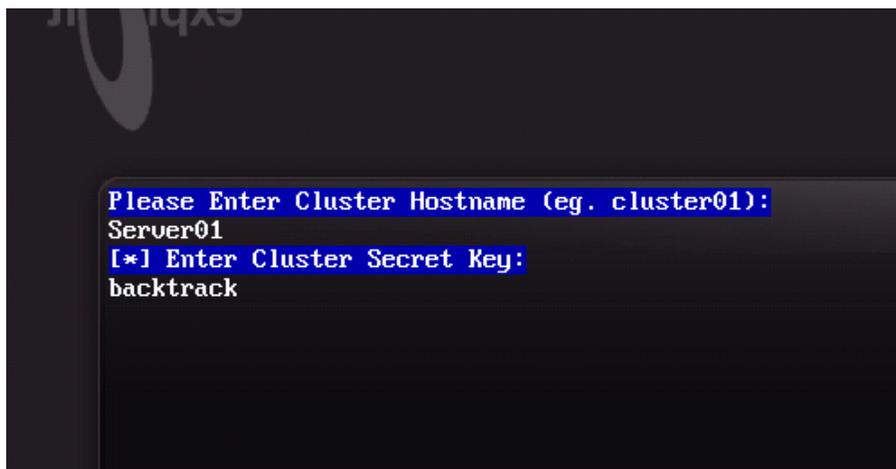
Once the server is up, you can log in, and try the command **mpdtrace**. This will show you the cluster participants. Notice that the PXE server is assigned the IP address 192.168.1.1 by default.

Offering Professional Offensive Security

Booting the Cluster Clients

Once the PXE server is up, the cluster clients can be booted. You will need to set the cluster clients to boot from PXE (network) in the BIOS settings.

Hopefully the booting machine will receive the initrd from the PXE server. When prompted with the **boot:** prompt, enter **bt load=client|john-mpi**

Backtrack should then boot, and prompt for the Client hostname (Cluster01), IP of the MPD Master (192.168.1.1) , and the Cluster key, as set on the Master (backtrack).

Log in to the client machine, and enter the command **mpdtrace.** You should see both computers participating in the cluster.

7

Running John MPI

The first thing we need to do before running john , is distribute the hash to all the cluster participants. The beta version of BackTrack still does not have automation of this – we're still working on it. For now, I suggest using scp, tftp, or some other method of transfer.

Load additional cluster as available, distribute the password hash file to all of them.

For testing purposes, I will change the root password on Client01 and crack the shadow file.

```
Client01 ~ # passwd

Client01 ~ # cp /etc/shadow /john/crackme

Client01 ~ # scp  /john/crackme 192.168.1.1:/john/

Client01 ~ # mpdrun -np 2 -path /john -wdir /john /john/john /john/crackme

Loaded 1 password hash (FreeBSD MD5 [32/32 X2])

Loaded 1 password hash (FreeBSD MD5 [32/32 X2])
```

Both computers should be running john, and cracking the password, a la cluster.