

Exploiting Cisco Routers: Part 2

Mark Wolfgang 2003-12-01

Access Granted -- Now What?

Welcome back! The [first article](#) in this two-part series covered a few different methods of getting into the target router. This article will focus on what we can do once we've gotten in. For the remainder of this article, we'll assume that the only progress we've made is that we've gotten the below router config via the vulnerable HTTP server. At this point, Access Control Lists (ACLs) prevent us from logging in directly to the router.

Analyzing the Router Config

As imagined, router config files can give the penetration tester a TON of useful information. One can identify new targets, identify sensitive systems or networks by analyzing the ACLs, learn passwords that may be used elsewhere, and a bunch of other information.

Now that we have the router config, we can analyze it for weaknesses, and hopefully glean other useful information from it. Our sample router config looks like this:

```
!  
version 12.0  
service timestamps debug uptime  
service timestamps log uptime  
service password-encryption  
!  
hostname router2  
!  
logging buffered errors  
enable secret 5 $1$sz0o$PYahL33gyTuHm9a8/UfmC1  
!  
username xyzadmin password 7 05331F35754843001754  
ip subnet-zero  
no ip routing  
!  
!  
!  
interface Ethernet0  
  description Internal Corporate Link  
  ip address 10.0.1.199 255.255.255.0
```

```
no ip directed-broadcast
no ip route-cache
no ip mroute-cache
!
interface Ethernet1
description Link to DMZ
ip address 172.16.1.1 255.255.255.0
no ip directed-broadcast
no ip route-cache
no ip mroute-cache
!
interface Serial0
description Link from PSInet
bandwidth 1536
no ip address
no ip directed-broadcast
no fair-queue
!
interface Serial1
no ip address
no ip directed-broadcast
no ip route-cache
no ip mroute-cache
shutdown
!
ip default-gateway 10.0.1.1
ip http server
ip classless
!
logging history critical
logging trap warnings
logging 10.0.1.103
access-list 100 permit tcp host 192.168.2.99 host 10.0.1.199 eq telnet
access-list 100 permit tcp host 192.168.2.99 host 10.0.1.199 eq finger
access-list 100 permit ip 0.0.0.0 255.255.255.248 host 10.0.1.199
access-list 100 permit ip host 10.0.1.103 any
access-list 100 deny ip any any
snmp-server community public RO
snmp-server community private RW
snmp-server location XYZ Widgets Inc. Server Room (417)
snmp-server contact Network Admins
```

```
snmp-server host 10.0.1.112 h3rn3c4
banner motd ^C
THIS IS A PRIVATE COMPUTER SYSTEM.
This computer system including all related equipment, network devices
(specifically including Internet access), are provided only for
authorized use. All computer systems may be monitored for all lawful
purposes, including to ensure that their use is authorized, for
management of the system, to facilitate protection against
unauthorized
access, and to verify security procedures, survivability and
operational security. Monitoring includes active attacks by authorized
personnel and their entities to test or verify the security of the
system. During monitoring, information may be examined, recorded,
copied and used for authorized purposes. All information including
personal information, placed on or sent over this system may be
monitored. Uses of this system, authorized or unauthorized,
constitutes
consent to monitoring of this system. Unauthorized use may subject
you
to criminal prosecution. Evidence of any such unauthorized use
collected during monitoring may be used for administrative, criminal
or
other adverse action. Use of this system constitutes consent to
monitoring for these purposes. ^C
!
line con 0
  password 7 01030717481C091D25
  transport input none
line aux 0
line vty 0 4
  password 7 095C4F1A0A1218000F
  login
!
end
```

Cracking the Enable Password

The first thing we'll do is attempt to "crack" the enable password. It is represented in the form of an MD5 hash, which is said to be uncrackable. We're not going to attempt to decrypt the password, as this

is not possible. Instead, we'll run a dictionary attack against it. In much the same way as John the Ripper plows through an /etc/shadow file, the very popular tool [Cain and Abel](#) is capable of conducting both brute-force and dictionary attacks on Cisco MD5 hashes.

This tool could be described as the Swiss Army Knife of cracking tools. The following screenshot shows this tool conducting a dictionary attack on the above enable hash.

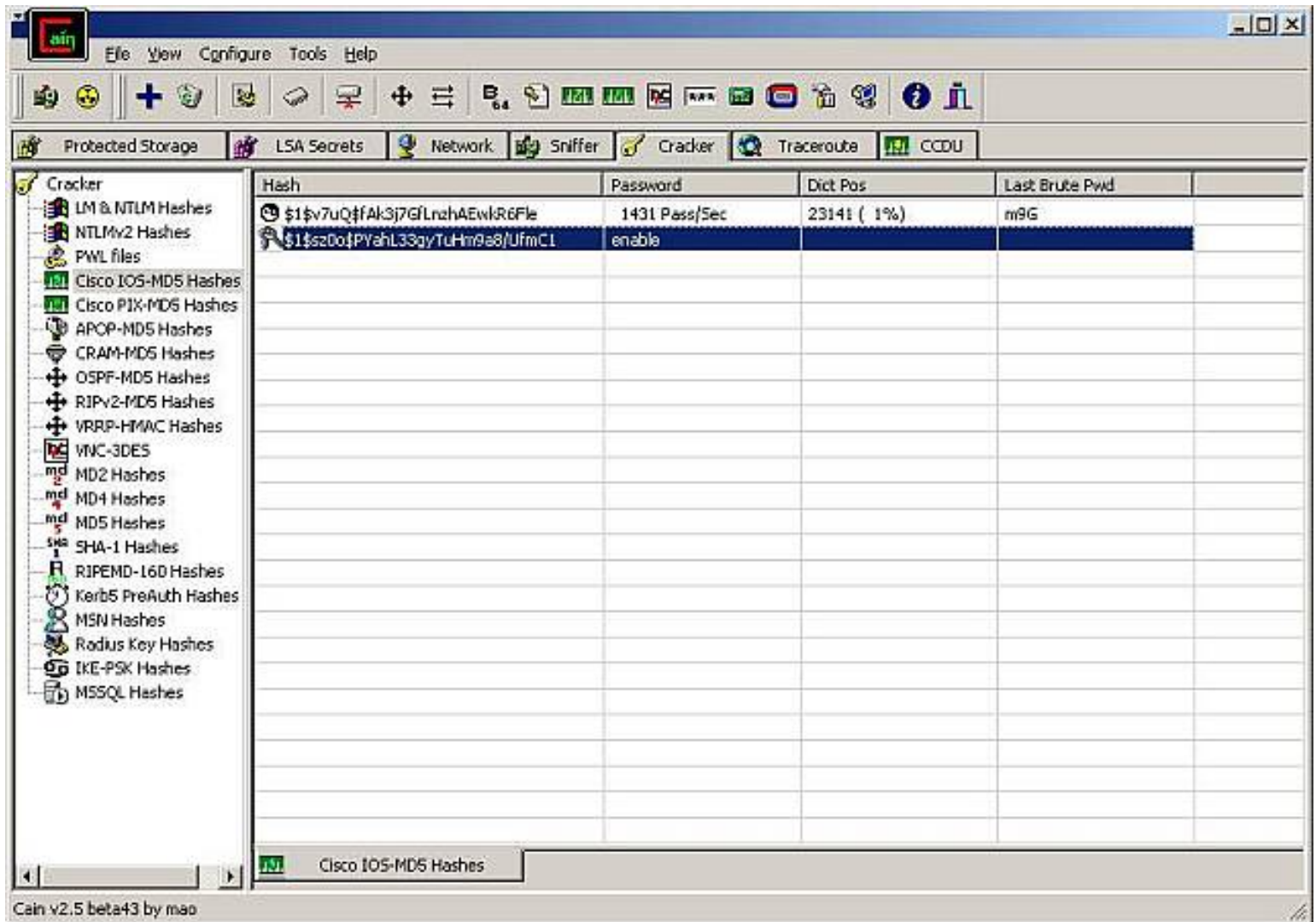


Figure 1: Cain and Abel

Take note of all the other types of passwords Cain and Abel can crack. This is a tool every Pen Tester has in his toolbox.

As displayed in the above image, Cain and Abel was successful in figuring out the enable password. Now that we have this critical piece of information we can attempt to log into the device. If pesky ACLs prevent us from logging in directly, we can either add a rule that allows us to log in, or completely disable an ACL. I'd guess that your organization is like mine in the fact that we want to increase the security of the target network and not increase the risk to the network. If this is the case, you'll likely just add an ACL that allows you to telnet into the router.

Before we modify anything with the router though, we'll take a quick look at the entire router config. Too often I find myself moving way too quickly, and inevitably I end up screwing something up. The one thing to point out at this time is the following lines:

```
logging buffered errors
logging history critical
logging trap warnings
logging 10.0.1.103
```

This router is logging at log level 4 to the syslog server 10.0.1.103. One could turn logging off completely, but that may raise suspicions too much. You may just want to increase the log level to a level where only emergencies are logged. This way, when we do modify the router nothing gets sent to the syslog server.

At this point, any changes we to make to the router will have to be done via the HTTP server.

We open up our favorite web browser (mine is not IE by the way) and take advantage of the previously mentioned HTTP vulnerability.



Figure 2: Raise the router's log level

Though we did modify the log level of the router so that most events will not be reported, no guarantees can be made that the network admins aren't using some other method of monitoring the device.

We then use the same method to add an ACL permitting us to access the router.

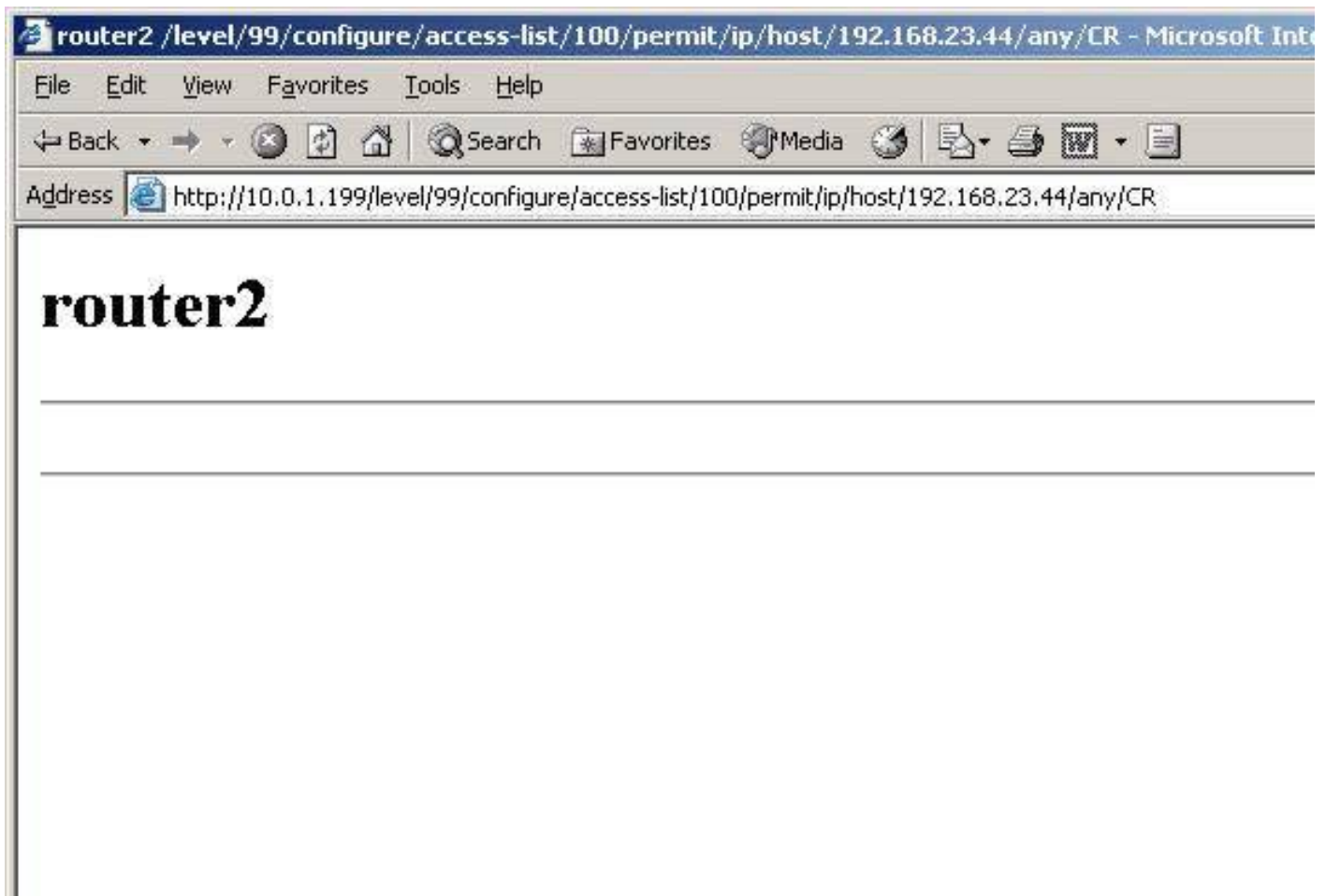


Figure 3: Add an ACL, giving us access to the router

Now that we've successfully added an ACL that allows us full access to the router and beyond, we can crack the weaker Vigenere password protecting the VTY ports and telnet in. Once we telnet in, we can use the newly acquired enable password to fully administer the router.

There are several other items worth mentioning in the router config file. The ACLs show a trusted host. This host (192.168.2.99) is permitted to login to the router and to also finger the router to see who is logged on. We also have the snmp read and write community strings, which can more than likely be used on other systems. I will typically put all of the passwords along with some other company-related words into a dictionary file for latter use. Additionally, there is the username "xyzadmin" defined.

All of the above information can be used in furthering the penetration test. A common practice in most networks I've seen is a class-based password scheme. That is, each different class of systems, whether it be Unix servers, NT servers, routers and switches etc. has a shared password. Believe it or not, some networks just have a common administrator password for everything! If this is true in this case, we may be able to leverage the access gained to this one device to attempt to login to other systems.

Just for the heck of it, we'll try to login to the syslog server. It's more than likely a Unix-based server,

so logging in as root won't be possible unless they've really loosened up security. We'll use the newly learned username of "xyzadmin" and quickly crack the associated password using GetPass. If this login attempt is successful, we'll have at the very minimum, a user level shell on the XYZ Widget Company's internal network. With any luck, the syslog server will have some local root vulnerabilities that once exploited give the attacker/pen tester a root shell. And all of this was possible because of a vulnerability in the Cisco HTTP server, that more than likely shouldn't have been running in the first place. Other targets may be TFTP servers that might be listed in the config file.

Another way to discover routers and switches on the network is to use CDP. The Cisco Discovery Protocol is extremely useful for "browsing" the network for other Cisco devices. It's useful for all types of people that gain access to your Cisco devices, including the bad ones. It's definitely not a necessary service, and shouldn't provide any information a well administered network doesn't already have. Once the devices are learned, the attacker can try the learned passwords and community strings on the new devices.

Once the pen tester is logged into the router, he will likely want to know what other systems he can access. Until he compromises more powerful systems, he can use both traceroute and telnet from the router to explore the internal network. Interface descriptions help greatly in learning the network. In the above router config, the interfaces have nice descriptive labels as reiterated below:

```
interface Ethernet0
  description Internal Corporate Link
  ip address 10.0.1.199 255.255.255.0
  no ip directed-broadcast
  no ip route-cache
  no ip mroute-cache
!
interface Ethernet1
  description Link to DMZ
  ip address 172.16.1.1 255.255.255.0
  no ip directed-broadcast
  no ip route-cache
  no ip mroute-cache
!
interface Serial0
  description Link from PSInet
  bandwidth 1536
  no ip address
  no ip directed-broadcast
```

```
no fair-queue
!
```

Want to shutdown the company's Internet access? Okay. Just shutdown the Serial link from the ISP. It's not likely that a pen tester would want to do this, but I list this example to hopefully gain some attention of the importance of the router. Also learned from the description is a couple more target networks to attack. Since the router provides the traffic for the networks, it will more than likely be trusted. Though using Telnet from the router may not appear to be very powerful, don't forget the above example concerning shared passwords.

The ACLs present in the above config aren't great, but analyzing a more complex set of ACLs may be quite revealing. You might learn a good source port to use for port scanning (if you don't want to modify the ACLs). Perhaps you'll learn of trusted networks or hosts.

You may be wondering about sniffing traffic from the router, since this one device does pass every packet into and out of the network. Though I won't cover it here, sniffing is possible. It's possible to establish a GRE tunnel with another router or system that can speak GRE. Policy routing can then be setup so that certain or all traffic can be sent to this other system via the GRE tunnel. On the other end of the GRE tunnel, systems capable of running sniffers can be setup to run ethereal or dsniff. All this, and the traffic still gets to its intended destination. Setting up sniffing like this is quite elaborate, and conditions and the load on the router have to be just right. It would be easy to overload the router's Internet connection, potentially shutting down the network. For this reason, I doubt most professional pen testers would take the risk of conducting this sort of an attack, unless they were expressly given permission. Some of my past customers haven't been happy if I knocked off one router, let alone the entire Internet connection! For more information on this type of attack, check out the excellent paper written by [David Taylor](#) .

Summary

Hopefully this series on Exploiting Routers has been somewhat enlightening to you, and perhaps you now have a greater sense of awareness on what you as a pen tester can do on your next task.

Links and Reference Material

[Hardening Cisco Routers](#), by Thomas Akin

[Stealing the Network: How to Own the Box -- Chapter 4](#)

Author Info

View [more articles](#) by Mark Wolfgang on SecurityFocus.

[Privacy Statement](#)

Copyright 2006, SecurityFocus