AntiOnline.com

Hackers Know The Weaknesses In your System. Shouldn't You?

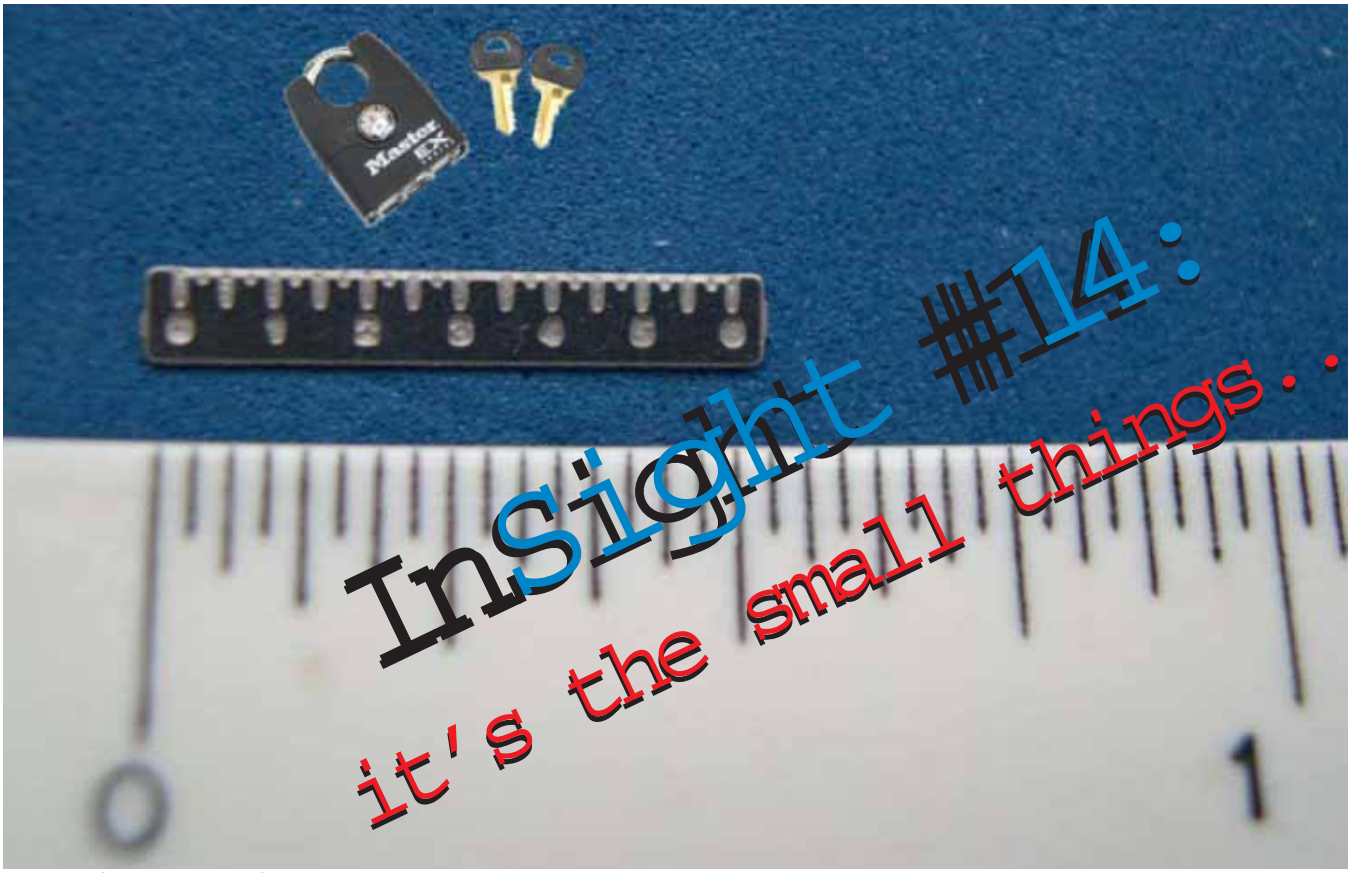


Image of Google Search sourced from http://www.transittools.co.uk/Toolpics/Misc/master%20lock%2050mm.jpg and http://www.thedollshousemall.com/paypal/044a%20%20%206%20inch%20ruler.jpg

Table of Contents

MsMittens' Editorial

What a challenging week this has been in preparation of this issue. Ranging from personal issues at work (boss goes out on canoe and only canoe returns) to the London bombings, it has reminded me of the importance of physical safety and security. In this day and age where we are rather complacent and "used to" violence, it may be important to re-iterate some of the things that we do need to be aware of.

It's not just a matter of who we think may not belong but rather what they bring with them that doesn't belong or that they left behind. Physical security isn't just about the hardware but should also include the "wetware" (i.e., human beings). Whatever security measures I put in place I have to be aware of the impact it will have on individuals and whether enough is being done to protect individuals. Physical security always puts the lives of humans first (certain industries, such as military, doesn't necessarily fit into this). We should be doing what we can to ensure the safety of co-workers and other employees (no matter how whiny or annoying they may be).

This means paying attention to things like fire hazards (how many Dilbert cartoons are piled up beside the fan for the server?); devices used for safety (when was the last time the fire extinguishers were checked? what about the First Aid kit? do you know where the First Aid kit is? do you even have one beyond 10-year old bandaids?); safety training (CPR training can help save a life); safety hazards (it may be easier to have 50-foot printer cables all over but how often is someone tripping and nearly hitting your desk corner?); etc. We are so focused on the big pictures these days (those that terrorists want us to focus on) that we often forget those little pictures around us. It's all about those little daily details that are important; not just the big picture with Osama on it.

And, of course, physical security has to be part of those daily details. It's not just a matter of building the ultimate server room or having the biggest Kryptolock; it's about watching who is going into what room and *WHY* are they going in there. You need to ask if the boss really needs to have the code into the server room. He/she may sign your cheque but you are the one responsible for everything that happens in there, physical or computational sense. Other items to pay attention to: what's on people's desk (people leave all sorts of confidentail material on their desk that "cleaning staff" can see — quite often cleaning staff are "transient" in nature); white boards that covered with confidentail material (passwords and such); "recycle bins" that hold confidential material; rarely, if ever, checking for rogue wireless and/or hardware keyloggers.

These aren't big things but they can result in big things. Much like this issue. It isn't huge but pay attention to those details and they can help you in the long run.

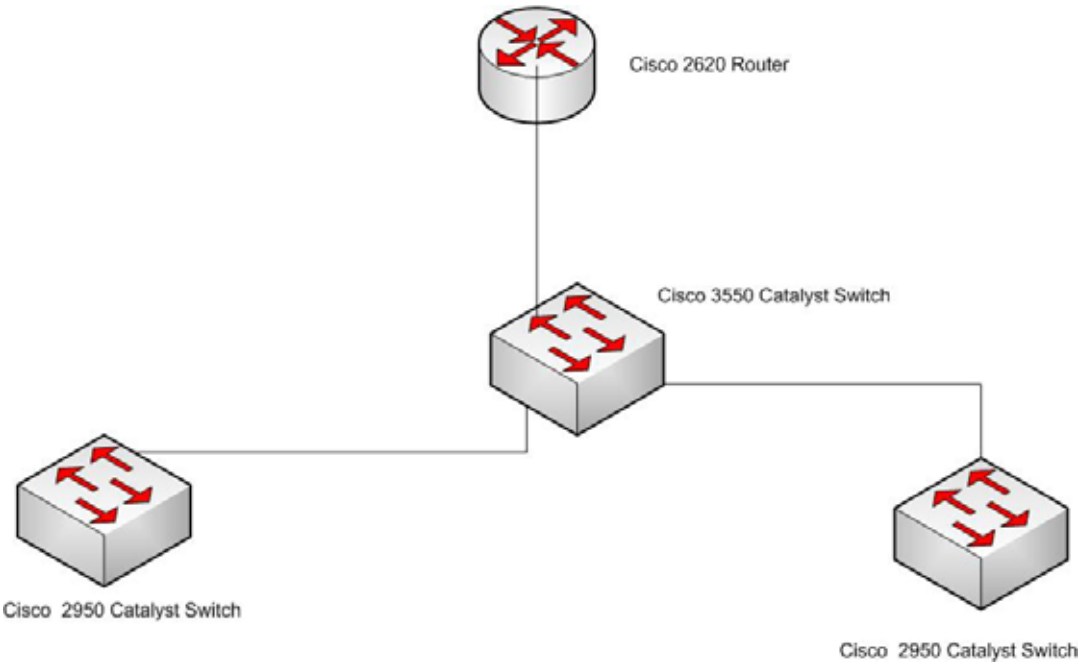
Many thanks to HTRegz (who submitted two articles) and Wolfrune (who submitted one). Next issue deadline is September 5th, 2005. Hopefully YOU can add your important details to this small world of security.

Have a good rest of the summer.

Defending Against Rogue DHCP Servers on a Collapsed Core Network

by HTRegz

For some time now, large corporations that provide public internet/intranet access have been plagued with the problem of Rogue DHCP servers. This issue can be easily solved with some basic switch security techniques. Due to a lack of hardware this will be demonstrated with the following network setup, layered in a collapsed core design.



The network will consist of 4 VLANs and require 5 address ranges. The details of this setup can be found in the table below.

Address Range	VLAN ID	Services	Devices
192.168.1.0/24	1	Management VLAN	3550, 2950-1, 2950-2
192.168.2.0/24	2	Client Access	2950-1 – Ports: 3 – 132950-2 – Ports: 3 – 73550 – Ports: 3 – 10
192.168.3.0/24	3	Client Access	2950-1 – Ports: 14 – 242950-2 – Ports: 8 – 123550 – Ports: 13 – 20
192.168.4.0/24	N/A	DHCPInternet Access	2620 Router3550
192.168.5.0/24	4	Web Server	Web Server3550

The web server used in this example is a Catalyst 2950 switch, so the config for that switch has also been included.

There are two forms of security used to restrict access and limit the attacks caused by Rogue DHCP servers. The first thought is to go immediately to 802.1x, since the devices support it. However, being a public institution this would limit availability and also require users to have usernames and passwords, while not fully protecting from an accident rogue DHCP server (for example from a VMWare client). The first item that is required is the Cisco command *switchport protected*; this should be placed on all ports that will be used for client access. To better explain the *switchport protected* command we can reference a CiscoPress website[1]

> A protected port feature is used in those environments where no traffic can be forwarded between two ports on the same switch. This way, one neighbor connected to one port does not see the traffic that is generated by another neighbor connected to the second port. The blocking of traffic (unicast, broadcast, or multicast) only works when both ports are protected. When a protected port is communicating with an unprotected port, the traffic is forwarded in the usual manner. Once the ports are protected, traffic between them can only be forwarded by a Layer 3 device.

This means that our clients will not know that each other exist. This is requires because VLANs are still using layer 2 functions and will Forward, Flood or Filter. If traffic from 192.168.1.1 came into our access-layer switch on physical port 2, logical VLAN 2 and was bound for 192.168.1.2 on physical port 3, logical VLAN 2 the traffic would be forwarded. 192.168.1.2 would automatically receive the information that 192.168.1.1 had sent. What *switchport protected* will do is block that communication deny communication between devices

on the same Virtual LAN. The port protection will be dropped when the frame hits the layer 3 device to become a packet.

That means that a packet originating at 192.168.1.1 (remember this was, for example, Port 2 – VLAN 2) and bound for 192.168.2.1 (Port 4 – VLAN 3) will not be protected using port protection. This is where our layer 3 switch at the collapsed core and distribution layers comes into play. Placed on the layer 3 switch are a series of access-lists which are combined to form a VLAN Map. This VLAN Map allows us to filter Layer 2 bound data while using Layer 3 protocols. In our example this will be DHCP. All we want to do is eliminate rogue DHCP servers while continuing to allow all other traffic. This will require 5 access-lists.

The first list will allow traffic from clients to port 67 on the DHCP server and the second list will be its counter part allowing traffic from the DHCP server back to its clients on port 68. The next two rules will limit our DHCP traffic. They will deny any traffic bound for port 68 or 67. This will not apply to traffic originating from or destined for the real DHCP server because the first rule that matches will be applied and processing of the VLAN Map will stop. The last rule will be an 'allow any any' this is because we don't want all our other traffic to be blocked when the end of the access map is reached. Now we just want to set the match statements. We want to forward anything that matches our first to rules, while dropping anything matching the next two (Rogue Protection) and again forwarding the traffic matching the last rule.

These two simple steps will allow for almost complete protection from Rogue DHCP servers on your network, while allowing for normal, uninterrupted network communication for the remainder of your network.

Footnotes:

[1] http://www.ciscopress.com/articles/article.asp?p=99029&seqNum=3

Appendices: Configuration Files

```
!
version 12.1
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname 2950-1
!
!
ip subnet-zero
!
!
spanning-tree mode pvst
no spanning-tree optimize bpdu transmission
spanning-tree extend system-id
!
!
!
!
interface Port-channel1
 switchport mode trunk
 flowcontrol send off
!
interface FastEthernet0/1
 switchport mode trunk
 channel-group 1 mode desirable
!
interface FastEthernet0/2
 switchport mode trunk
 channel-group 1 mode desirable
!
interface FastEthernet0/3
 switchport access vlan 2
 switchport mode access
 switchport protected
!
interface FastEthernet0/4
 switchport access vlan 2
 switchport mode access
 switchport protected
!
```

```
interface FastEthernet0/5
 switchport access vlan 2
 switchport mode access
 switchport protected
!
interface FastEthernet0/6
 switchport access vlan 2
 switchport mode access
 switchport protected
!
interface FastEthernet0/7
 switchport access vlan 2
 switchport mode access
 switchport protected
!
interface FastEthernet0/8
 switchport access vlan 2
 switchport mode access
 switchport protected
!
interface FastEthernet0/9
 switchport access vlan 2
 switchport mode access
 switchport protected
!
interface FastEthernet0/10
 switchport access vlan 2
 switchport mode access
 switchport protected
!
interface FastEthernet0/11
 switchport access vlan 2
 switchport mode access
 switchport protected
!
interface FastEthernet0/12
 switchport access vlan 2
 switchport mode access
 switchport protected
!
interface FastEthernet0/13
 switchport access vlan 2
 switchport mode access
 switchport protected
```

```
!
interface FastEthernet0/14
 switchport access vlan 3
 switchport mode access
 switchport protected
!
interface FastEthernet0/15
 switchport access vlan 3
 switchport mode access
 switchport protected
!
interface FastEthernet0/16
 switchport access vlan 3
 switchport mode access
 switchport protected
!
interface FastEthernet0/17
 switchport access vlan 3
 switchport mode access
 switchport protected
!
interface FastEthernet0/18
 switchport access vlan 3
 switchport mode access
 switchport protected
!
interface FastEthernet0/19
 switchport access vlan 3
 switchport mode access
 switchport protected
!
interface FastEthernet0/20
 switchport access vlan 3
 switchport mode access
 switchport protected
!
interface FastEthernet0/21
 switchport access vlan 3
 switchport mode access
 switchport protected
!
interface FastEthernet0/22
 switchport access vlan 3
 switchport mode access
 switchport protected
```

```
!
interface FastEthernet0/23
 switchport access vlan 3
 switchport mode access
 switchport protected
!
interface FastEthernet0/24
 switchport access vlan 3
 switchport mode access
 switchport protected
!
interface GigabitEthernet0/1
!
interface GigabitEthernet0/2
!
interface Vlan1
 ip address 192.168.1.2 255.255.255.0
 no ip route-cache
!
ip http server
!
line con 0
line vty 5 15
!
!
end
```

```
!
version 12.1
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname 2950-2
!
!
ip subnet-zero
!
spanning-tree extend system-id
!
!
interface Port-channel1
 switchport mode trunk
 no ip address
 flowcontrol send off
!
interface FastEthernet0/1
 switchport mode trunk
 no ip address
 channel-group 1 mode desirable
!
interface FastEthernet0/2
 switchport mode trunk
 no ip address
 channel-group 1 mode desirable
!
interface FastEthernet0/3
 switchport access vlan 2
 no ip address
 switchport mode access
 switchport protected
!
interface FastEthernet0/4
 switchport access vlan 2
 no ip address
 switchport mode access
 switchport protected
```

```
!
interface FastEthernet0/5
 switchport access vlan 2
 no ip address
 switchport mode access
 switchport protected
!
interface FastEthernet0/6
 switchport access vlan 2
 no ip address
 switchport mode access
 switchport protected
!
interface FastEthernet0/7
 switchport access vlan 2
 no ip address
 switchport mode access
 switchport protected
!
interface FastEthernet0/8
 switchport access vlan 3
 no ip address
 switchport mode access
 switchport protected
!
interface FastEthernet0/9
 switchport access vlan 3
 no ip address
 switchport mode access
 switchport protected
!
interface FastEthernet0/10
 switchport access vlan 3
 no ip address
 switchport mode access
 switchport protected
!
interface FastEthernet0/11
 switchport access vlan 3
 no ip address
 switchport mode access
 switchport protected
```

```
!
interface FastEthernet0/12
 switchport access vlan 3
 no ip address
 switchport mode access
 switchport protected
!
interface FastEthernet0/13
 no ip address
 switchport mode access
 switchport protected
!
interface FastEthernet0/14
 no ip address
 switchport mode access
 switchport protected
!
interface FastEthernet0/15
 no ip address
 switchport mode access
 switchport protected
!
interface FastEthernet0/16
 no ip address
 switchport mode access
 switchport protected
!
interface FastEthernet0/17
 no ip address
 switchport mode access
 switchport protected
!
interface FastEthernet0/18
 no ip address
 switchport mode access
 switchport protected
!
interface FastEthernet0/19
 no ip address
 switchport mode access
 switchport protected
!
interface FastEthernet0/20
 no ip address
 switchport mode access
 switchport protected
!
```

```
interface FastEthernet0/21
 no ip address
 switchport mode access
 switchport protected
!
interface FastEthernet0/22
 no ip address
 switchport mode access
 switchport protected
!
interface FastEthernet0/23
 no ip address
 switchport mode access
 switchport protected
!
interface FastEthernet0/24
 no ip address
 switchport mode access
 switchport protected
!
interface Vlan1
 ip address 192.168.1.3 255.255.255.0
 no ip route-cache
!
ip http server
!
!
line con 0
line vty 5 15
!
end
```

```
!
version 12.1
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname 3550
!
!
ip subnet-zero
ip routing
!
!
spanning-tree mode pvst
spanning-tree extend system-id
!
!
vlan access-map DHCPRogueBlock 10
 action forward
 match ip address DHCPtoServer
 vlan access-map DHCPRogueBlock 20
 action forward
 match ip address DHCPtoClient
vlan access-map DHCPRogueBlock 30
 action drop
 match ip address DHCPtoRogue
vlan access-map DHCPRogueBlock 40
 action drop
 match ip address DHCPfromRogue
vlan access-map DHCPRogueBlock 50
 action forward
 match ip address AllowAll

!
!
interface Port-channel1
 switchport trunk encapsulation dot1q
 switchport mode trunk
 no ip address
!
interface Port-channel2
 switchport trunk encapsulation dot1q
 switchport mode trunk
 no ip address
!
```

```
interface FastEthernet0/1
 switchport trunk encapsulation dot1q
 switchport mode trunk
 no ip address
 channel-group 1 mode auto
!
interface FastEthernet0/2
 switchport trunk encapsulation dot1q
 switchport mode trunk
 no ip address
 channel-group 1 mode auto
!
interface FastEthernet0/3
 switchport access vlan 2
 no ip address
 switchport mode access
 switchport protected
!
interface FastEthernet0/4
 switchport access vlan 2
 no ip address
 switchport mode access
 switchport protected
!
interface FastEthernet0/5
 switchport access vlan 2
 no ip address
 switchport mode access
 switchport protected
!
interface FastEthernet0/6
 switchport access vlan 2
 no ip address
 switchport mode access
 switchport protected
!
interface FastEthernet0/7
 switchport access vlan 2
 no ip address
 switchport mode access
 switchport protected
!
interface FastEthernet0/8
 switchport access vlan 2
 no ip address
 switchport mode access
 switchport protected
```

```
!
interface FastEthernet0/9
 switchport access vlan 2
 no ip address
 switchport mode access
 switchport protected
!
interface FastEthernet0/10
 switchport access vlan 2
 no ip address
 switchport mode access
 switchport protected
!
interface FastEthernet0/11
 switchport trunk encapsulation dot1q
 switchport mode trunk
 no ip address
 channel-group 2 mode auto
!
interface FastEthernet0/12
 switchport trunk encapsulation dot1q
 switchport mode trunk
 no ip address
 channel-group 2 mode auto
!
interface FastEthernet0/13
 switchport access vlan 3
 no ip address
 switchport mode access
 switchport protected
!
interface FastEthernet0/14
 switchport access vlan 3
 no ip address
 switchport mode access
 switchport protected
!
interface FastEthernet0/15
 switchport access vlan 3
 no ip address
 switchport mode access
 switchport protected
```

```
!
interface FastEthernet0/16
 switchport access vlan 3
 no ip address
 switchport mode access
 switchport protected
!
interface FastEthernet0/17
 switchport access vlan 3
 no ip address
 switchport mode access
 switchport protected
!
interface FastEthernet0/18
 switchport access vlan 3
 no ip address
 switchport mode access
 switchport protected
!
interface FastEthernet0/19
 switchport access vlan 3
 no ip address
 switchport mode access
 switchport protected
!
interface FastEthernet0/20
 switchport access vlan 3
 no ip address
 switchport mode access
 switchport protected
!
interface FastEthernet0/21
 no switchport
 ip address 192.168.4.2 255.255.255.0
 ip helper-address 192.168.4.1
!
interface FastEthernet0/22
 no ip address
 switchport mode access
 switchport protected
!
interface FastEthernet0/23
 no ip address
 switchport mode access
 switchport protected
```

```
!
interface FastEthernet0/24
 switchport trunk encapsulation dot1q
 switchport mode trunk
 no ip address
!
interface GigabitEthernet0/1
 no ip address
!
interface GigabitEthernet0/2
 no ip address
!
interface Vlan1
 ip address 192.168.1.1 255.255.255.0
!
interface Vlan2
 ip address 192.168.2.1 255.255.255.0
 ip helper-address 192.168.4.1
!
interface Vlan3
 ip address 192.168.3.1 255.255.255.0
 ip helper-address 192.168.4.1
!
interface Vlan4
 ip address 192.168.5.1 255.255.255.0
!
router rip
 network 192.168.1.0
 network 192.168.2.0
 network 192.168.3.0
 network 192.168.4.0
 network 192.168.5.0
!
ip classless
ip http server
!
ip access-list extended AllowAll
 permit ip any any
ip access-list extended DHCPtoClient
 permit udp host 192.168.4.1 any eq bootpc
ip access-list extended DHCPtoServer
 permit udp any host 192.168.4.1 eq bootps
ip access-list extended DHCPtoRogue
 deny udp any any eq bootps
ip access-list extended DHCPfromRogue
 deny udp any any eq bootpc
!
!
!
line con 0
```

```
!
version 12.1
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname WebServer
!
!
ip subnet-zero
!
spanning-tree extend system-id
!
!
interface FastEthernet0/1
 no ip address
!
interface FastEthernet0/2
 switchport access vlan 3
 switchport mode trunk
 no ip address
!
interface FastEthernet0/3
 switchport access vlan 4
 no ip address
!
interface FastEthernet0/4
 switchport access vlan 4
 no ip address
!
interface FastEthernet0/5
 switchport access vlan 4
 no ip address
!
interface FastEthernet0/6
 switchport access vlan 4
 no ip address
!
interface FastEthernet0/7
 switchport access vlan 4
 no ip address
!
interface FastEthernet0/8
 switchport access vlan 4
 no ip address
```

```
!
interface FastEthernet0/9
 switchport access vlan 4
 no ip address
!
interface FastEthernet0/10
 switchport access vlan 4
 no ip address
!
interface FastEthernet0/11
 switchport access vlan 4
 no ip address
!
interface FastEthernet0/12
 switchport access vlan 4
 no ip address
!
interface FastEthernet0/13
 switchport access vlan 4
 no ip address
!
interface FastEthernet0/14
 switchport access vlan 4
 no ip address
!
interface FastEthernet0/15
 switchport access vlan 4
 no ip address
!
interface FastEthernet0/16
 switchport access vlan 4
 no ip address
!
interface FastEthernet0/17
 switchport access vlan 4
 no ip address
!
interface FastEthernet0/18
 switchport access vlan 4
 no ip address
!
interface FastEthernet0/19
 switchport access vlan 4
 no ip address
interface FastEthernet0/20
 switchport access vlan 4
 no ip address
```

```
!
interface FastEthernet0/21
 switchport access vlan 4
 no ip address
!
interface FastEthernet0/22
 switchport access vlan 4
 no ip address
!
interface FastEthernet0/23
 switchport access vlan 4
 no ip address
!
interface FastEthernet0/24
 switchport access vlan 4
 no ip address
!
interface Vlan1
 ip address 192.168.1.4 255.255.255.0
 no ip route-cache
 shutdown
!
interface Vlan3
 no ip address
 no ip route-cache
 shutdown
!
interface Vlan4
 ip address 192.168.5.2 255.255.255.0
 no ip route-cache
!
ip http server
!
!
line con 0
line vty 5 15
!
end
```

End-Users: Hopeless Cases?

By WolfRune (Matt M)

During a recent visit to my parent's home, I did what I always do: maintenance on their PC. On this occasion though, a conversation with my dear mum gave me pause for thought.

Upon requesting the use of her car for the evening (something I rarely do), I was ribbed a bit about not helping her with the load of top soil she was moving. I countered that I had been engrossed in running spyware scans, virus scans, software updates, and defragging the hard drive. She turned to me and asked "Does it need all that?" I replied that she didn't HAVE to get the oil changed in her car, but would it be a good idea not to?

I've been considering this conversation, and it has helped me to realize that we, as security folk (since this community has a mix of tinkerers like me to full-blown Ph.D. expert-types), can't forget the end-users. The people who, like my mum, don't understand why all of this is necessary.

The first and most important thing is to learn to speak their language. This may seem silly, since we're probably speaking to family or friends, who probably speak the same language as us. However, many of us also speak Technospeak (also known as Technobabble). All too often, when asked an innocuous computer question, many of us may end up babbling on in Techospeak, resulting in a glazed-over look from the person who wishes they'd never asked the question. To educate a user who has little-to-no knowledge of computers or how they work, you must frame your explanations in such a way as to be understood.

An example of such framing: My mum, while not exactly Mrs. Auto Mechanic, knows the importance and the reasoning behind maintaining her car (she commutes a long distance for work). By framing explanations in that context, she may better understand why it's important to run scans and check for updates proactively.

Almost as important is patience. Sometimes, with a hard-headed user who doesn't seem to be listening or just doesn't get it, this can be quite difficult. However, flying off the handle at them is going to make your job a lot more difficult for you down the road, so try to keep your cool.

Finally, the technical stuff. You don't have to be a security guru, but you should be familiar with the use of basic tools and concepts for performing security upgrades and maintenance. Its also can be quite helpful to have a CD with things like legit free software, such as AVG Antivirus, Spybot: Search and Destroy, and Process Explorer. Service packs are helpful, but a full copy which doesn't require an internet connection to install can take up a big chunk of your CD. USB Drives, DVDs, and other portable media can be used, but always remember that home users don't always have the appropriate hardware/software/operating system to support your media. CDs are as close as you can get to ubiquitous high-capacity media these days.

Also, you should carry a copy of Knoppix, or another LiveCD Operating System for those systems too clogged with spyware and viruses to boot properly. In terms of physical tools,

there are lots of really groovy pieces of equipment that will make people go “Ooh,” but when you get right down to it, there’s only a couple of basic tools you should carry for general purpose repairs:

-A Screwdriver (One that comes with multiple heads is a good idea)

-A Pair of tweezers (for grabbing dropped screws and picking out jumpers)

I’ve also recently obtained a set of what I like to call my “Extend-o-cables.” You can pick up a set with just about everything you could need for less than $50 CDN, and I’ve found it to be a valuable investment.

Good luck, and remember to Expect the Unexpected!

See your name in print!

Insight is always looking for more articles on various security topics.

Next submission date:

September 5, 2005

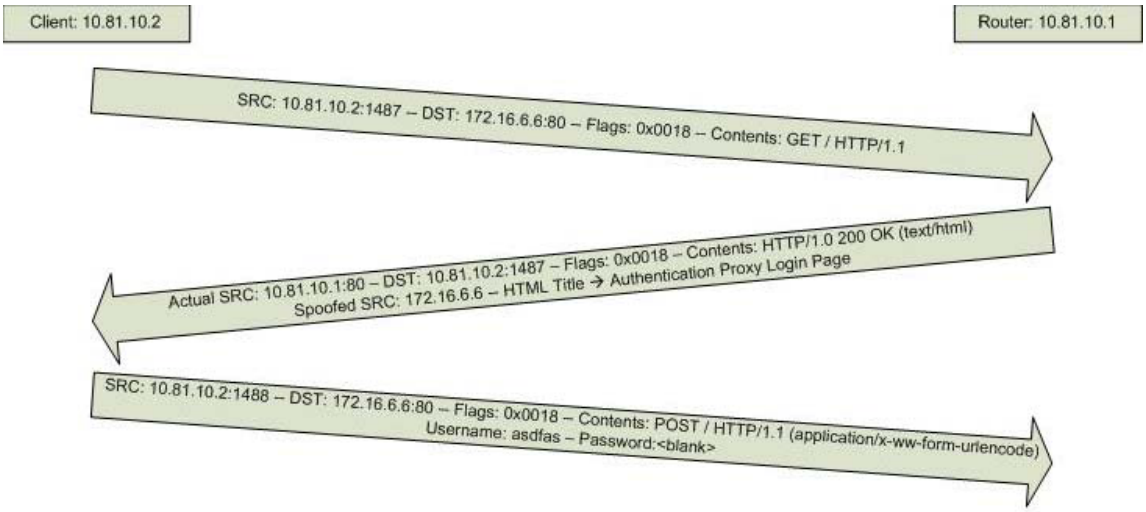
Authentication Proxy

by HTRegz

Authentication proxy is a rather intelligent way of regulating access to web servers on your network. Assuming that you have RADIUS or TACACS+ setup, you can quite easily allow for user authentication before accessing web pages, without worrying about insecure web scripting or future web server exploits. Your router will sit between the client and the web server and authenticate the user before allowing them access to the actual web server, and the best part is that it's entirely transparent to the end user. You have accomplished two goals by enabling authentication proxy, your network is more secure and there is less demand on the resources of your network (no need for a separate server to house the username/ authentication database for the web server).

In order to fully investigate how authentication proxy truly works, we'll require the details of a series of packet captures, which will be represented with a series of block diagrams. To conserve space, and keep the diagrams as on track as possible, block diagrams of the TCP 3-Way Handshake (SYN, SYN/ACK, ACK) and 4-Way Close (FIN, ACK, FIN, ACK) will not included. All demonstrated traffic is TCP and the handshakes can be assumed.

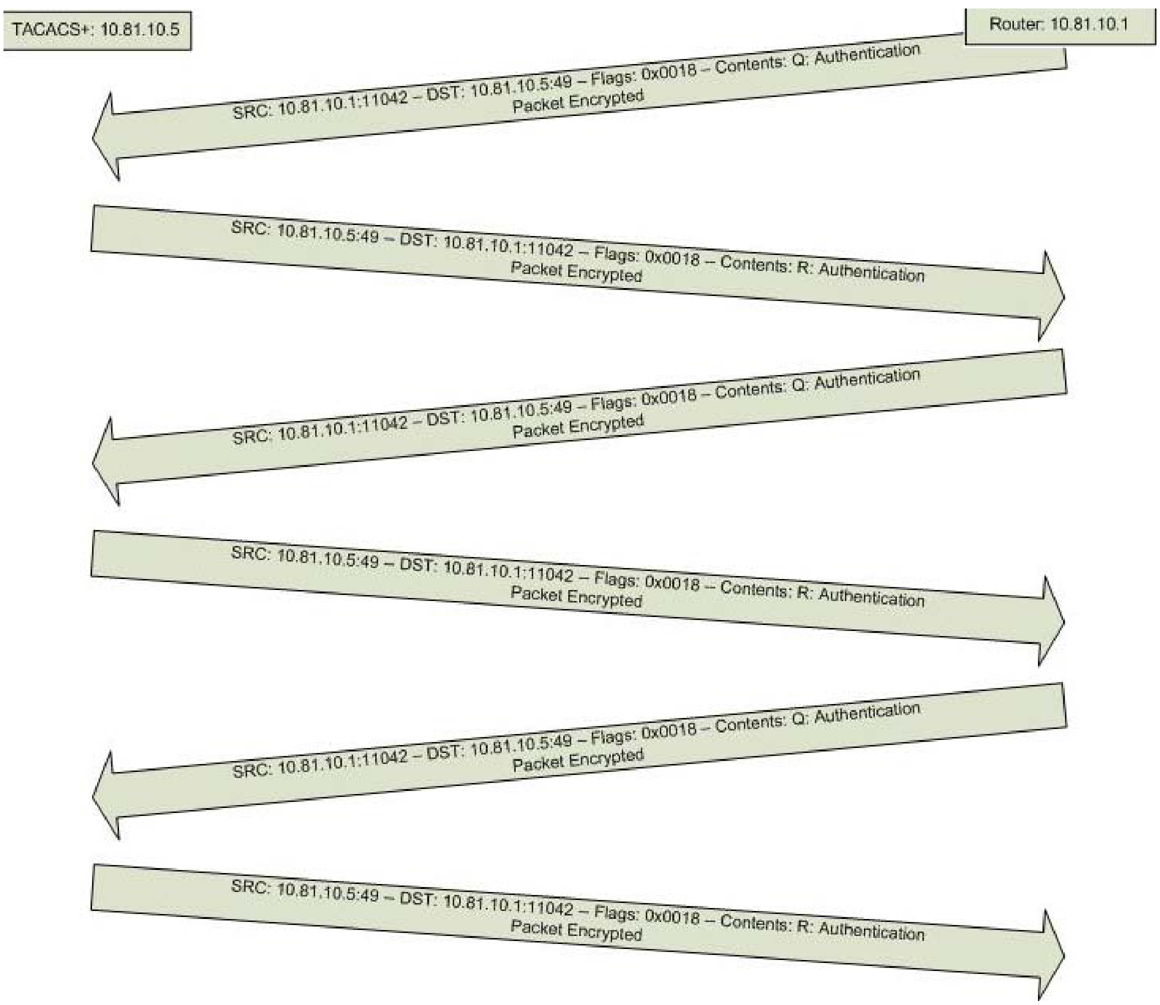
STEP 1 – The Client Requests a Website (Incorrect Password)



As you can see the client with the address of 10.81.10.2 (hereafter referred to as 'the client') requests the default webpage located on the web server at 172.16.6.6 (hereafter referred to as 'the web server'). The client then receives a page in response with the title 'Authentication Proxy Login Page' (*Figure 1*). This page appears to come from the web server but in reality has originated from the router running the authentication proxy service at 10.81.10.1 (hereafter referred to as 'the router'). The router has spoofed the IP Address of the

web server and is now waiting for the client to authenticate. Here's where the security flaw in this entire process comes into play. As you will see in upcoming steps, communication between the router and the TACACS+ server is encrypted, which is good… no sniffing can occur, however at this point the username and password are passed, by means of an HTTP POST, in plain text. To demonstrate exactly how this process works and fill in all the details, incorrect credentials were passed to the router at this point.

STEP 2 – Communication with the TACACS+ Server (Authentication Failed)



The client is completely oblivious to this step in the operation; it solely waits for a response to its post. While it's waiting for a response, the router is verifying the credentials it has received against the TACACS+ server at 10.81.10.5 (hereafter referred to as 'the TACACS+ server'). I was a little confused at first, as to why there were 6 packets exchanged between the two devices. I had assumed that a single exchange would take place, where the username and password were passed and then a success or failure response would come back to the router. After thinking for a while, I decided to search the internet for software that would dissect the TACACS+ packets. With a little help from internet forums, it was discovered

that Ethereal (the packet sniffer used in this case) has a built in TACACS+ dissector, all that is required is the key that was used. Since we had set this up in lab, the key was known (info5025) and I was able to enter it and obtain the encrypted portion of the TACACS+ packet in clear text.

The process was actually quite interesting. The first packet from the router to the TACACS+ server tells the router that it has an 'Inbound Login' with a current Privilege Level of 0 coming from port 'FastEthernet0/0' with a client IP Address of 10.81.10.2. The TACACS+ server responds with the message 'User Access Verification' and then a new line and then 'username:' (status 0x04 (send username)). The router then responds with 'user: asdfas' (the username from the HTTP POST packet). The TACACS+ server receives this information and responds with the message 'Password:', this is also the first time that the flags in the decrypted TACACS+ packet are not set to 0x00, this time they are set to 0x01 which means No Echo. The router replies without a message (nothing along the lines of password:) the only response is that the user length is 0 (meaning the blank password that we sent in our HTTP POST). The TACACS+ server then responds with the status 0x2 which means Authentication Failed.

We can break this down to make it a little easier to understand:

Packet 1
10.81.10.1:11042 à 10.81.10.5:49
Action: Inbound Login
Privilege Level: 0
Port: FastEthernet0/0
Client IP: 10.81.10.2

Packet 2
10.81.10.5:49 à 10.81.10.1:11042
Status: 0x04 (send username)
Message: *User Access Verification*

Username:

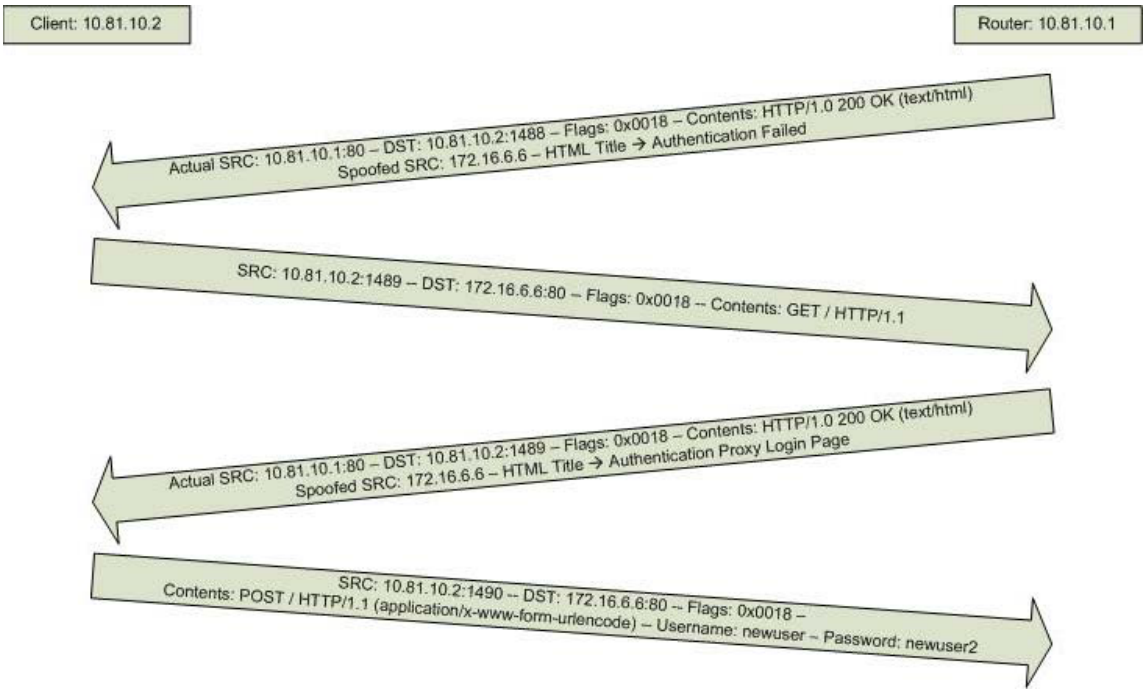
Packet 3
10.81.10.1:11042 à 10.81.10.5:49
User Length: 6
User: asdfas

Packet 4
10.81.10.5:49 à 10.81.10.1:11042
Status: 0x05 (send password)
Flags: 0x01(No Echo)
Message: *Password:*

Packet 5
10.81.10.1:11042 à 10.81.10.5:49
User Length: 0

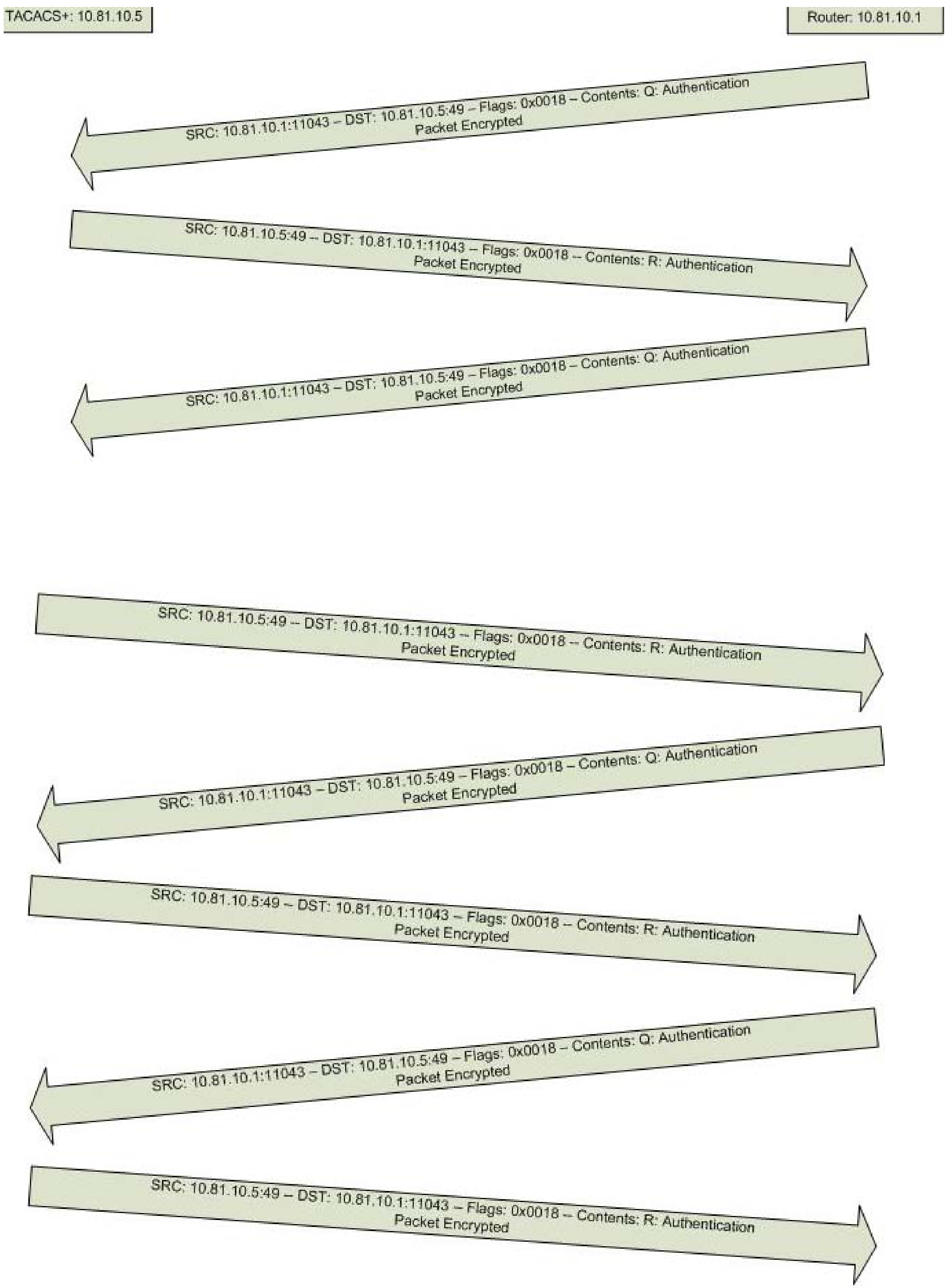
Packet 6
10.81.10.5:49 à 10.81.10.1:11042
Status: 0x2 (Authentication Failed)

STEP 3 – Client Attempts Website Again (Correct Password)



At this stage, we see a repeat of most of step 1. The first step is to respond to the client that the TACACS+ server didn't successfully authenticate the supplied credentials. This is seen by the user in the form of a web page with the title 'Authentication Failed'. Then step one is repeated. The web site is requested from the web server, the router steps in and spoofs a response back to the client. The client again attempts to log in, this time passing the correct credentials (again through an insecure plain-text HTTP Post) to the router.

STEP 4 – Communication with the TACACS+ Server (Authentication Successful)



This step is where the successful authentication actually occurs. As you can see the first 5 packets are exactly the same as the packets sent during the failed attempt. It is packets 6, 7 and 8 that differ. Since I'm assuming that the reader already has a basic understanding of the process from Step 2, I'm going to bypass the explanation, and detail the packets in their decrypted state and then

Packet 1
10.81.10.1:11042 à 10.81.10.5:49
Action: Inbound Login
Privilege Level: 0
Port: FastEthernet0/0
Client IP: 10.81.10.2

Packet 2
10.81.10.5:49 à 10.81.10.1:11042
Status: 0x04 (send username)
Message: *User Access Verification*

Username:

Packet 3
10.81.10.1:11042 à 10.81.10.5:49
User Length: 7
User: newuser

Packet 4
10.81.10.5:49 à 10.81.10.1:11042
Status: 0x05 (send password)
Flags: 0x01(No Echo)
Message: *Password:*

Packet 5
10.81.10.1:11042 à 10.81.10.5:49
User Length: 8
User: newuser2

Packet 6
10.81.10.5:49 à 10.81.10.1:11042
Status: 0x1 (Authentication Passed)

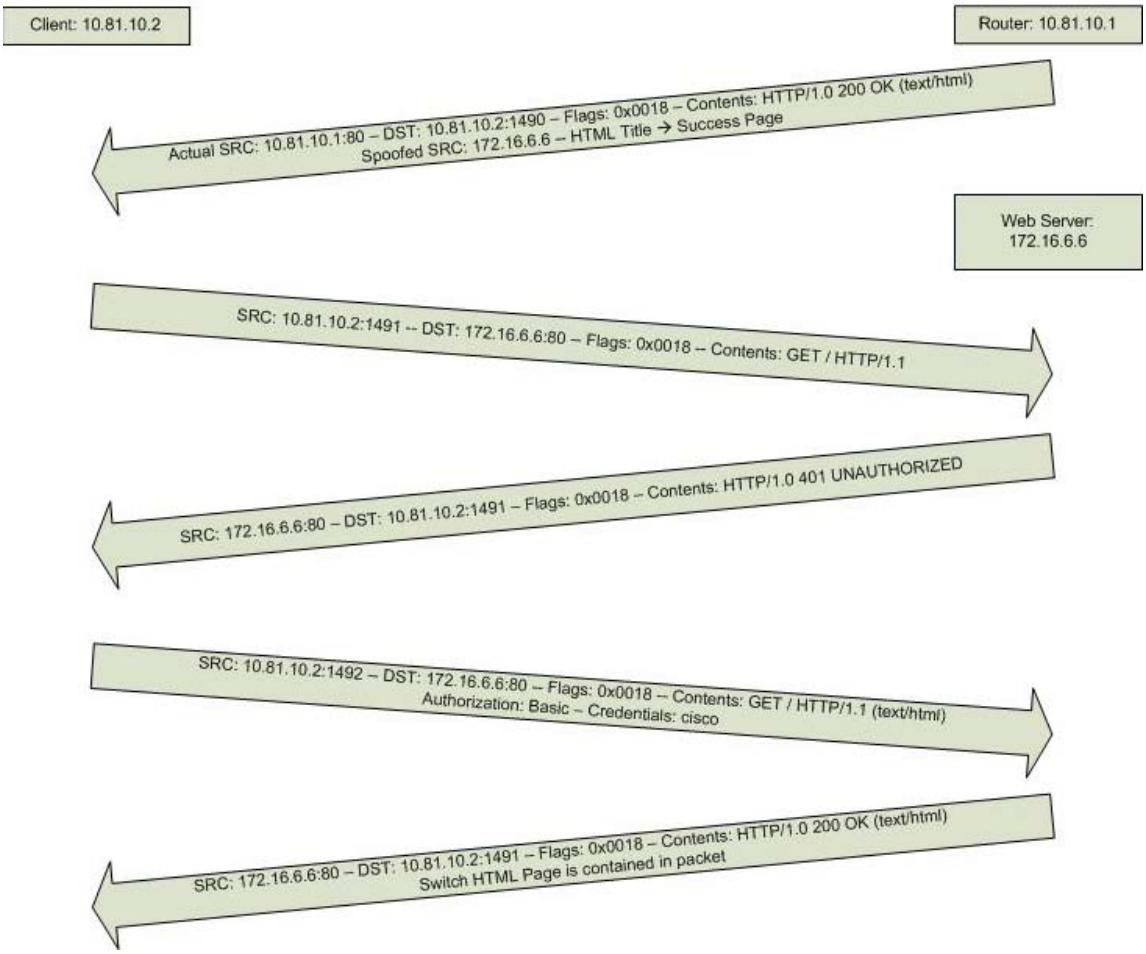
Packet 7
10.81.10.1:11042 à 10.81.10.5:49
Auth Method: TACACSPLUS
Privilege Level: 0
Service: Login
User len: 7
User: newuser

Port: FastEthernet0/0
Remote Address: 10.81.10.2
Arg count: 2
Arg[0] length: 18
Arg[0] value: service=auth-proxy
Arg[1] length: 4
Arg[1] value: cmd*

Packet 8
10.81.10.5:49 à 10.81.10.1:11042
Auth: Status: 0x1 (PASS_ADD)
Arg count: 2
Arg[0] length: 29
Arg[0] value: proxyacl#1=permit tcp any any
Arg[1] length: 11
Arg[1] value: priv-lvl=15

As you can see, we have a repeat of the previous attempt (with different credentials this time) and then we have ‘Authentication Passed’ instead of ‘Authentication Failed’. The router then passes along all the information again, username, port, and IP Address. The router is asking for any commands available for the service with the name auth-proxy, a service which we created on the CSACS. The response from the TACACS+ server is the series of commands which we entered in the user’s authorization profile, the Proxy ACL and a Privilege Level. The router uses this profile to create dynamic Access Control Entries (ACEs) and applies them to the inbound ACL of the interface; it will also apply them to the outbound ACL of the interface if the outbound ACL exists1.

Step 5 – Communication with Web Server



During this last stage, the router falls out of the picture (beyond its normal operation). After receiving the successfully authentication from the TACACS+ server and applying the proper access lists, the router returns to the client (which is still waiting from its HTTP POST in step 3) a success page (*Figure 2*). The success page contains a 'DONE' button. When the client clicks the 'DONE' button an HTTP GET for the default page (this was the original command that initiated this entire process) is sent to the web server. This time the router does not intercept the traffic, but allows it to pass freely. The web server receives the HTTP GET Request, however, since in this case we were using a Cisco Switch, which is configured to require authentication, we receive the response of HTTP 401 UNAUTHORIZED. This generates a pop-up authentication window on our client. The client supplies the proper credentials (the password of 'cisco' with no username) in plain text through an HTTP GET. The web server authenticates the password and sends the user back the html page.

This completes the process, which has happened completely transparent to the end user. This process has its merits and its downfalls… One of the advantages is that you can

prevent unauthorized access to production services. The user will not be able to access the machine and use SQL injection (if you authenticate against a SQL database), or session hijacking (since it's on a per machine basis) in order to access to protected data. The downfall is that you could run your web server authentication across an SSL connection where as with Authentication Proxy, your username and password are sent in plain-text through the initial HTTP post and the TACACS+ server communication can be dissected and displayed with knowledge of the secret key.

Cisco has also provided a list of reasons that person may want to run authentication proxy2. Those reasons are:

- You want to manage access privileges on an individual (per-user) basis using the services provided by the authentication servers instead of configuring access control based on host IP address or global access policies. Authenticating and authorizing users from any host IP address also allows network administrators to configure host IP addresses using DHCP.
- You want to authenticate and authorize local users before permitting access to intranet or Internet services or hosts through the firewall.
- You want to authenticate and authorize remote users before permitting access to local services or hosts through the firewall.
- You want to control access for specific extranet users. For example, you might want to authenticate and authorize the financial officer of a corporate partner with one set of access privileges while authorizing the technology officer for that same partner to use another set of access privileges.
- You want to use the authentication proxy in conjunction with VPN client software to validate users and to assign specific access privileges.
- You want to use the authentication proxy in conjunction with AAA accounting to generate "start" and "stop" accounting records that can be used for billing, security, or resource allocation purposes, thereby allowing users to track traffic from the authenticated hosts.

A very handy Cisco guide is available @ http://www.cisco.com/en/US/products/sw/iosswrel/ps1835/products_configuration_guide_chapter09186a00800ca7c7.html#wp1000871

Figures

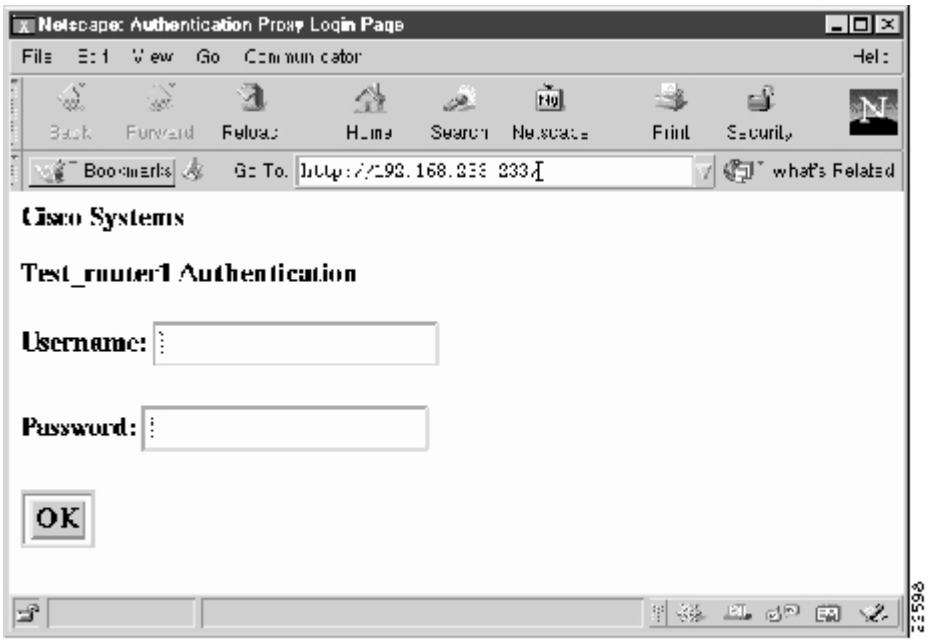


Figure 13

Figure 24

References:

[1]http://www.cisco.com/en/US/products/sw/iosswrel/ps1835/products_configuration_guide_chapter09186a00800ca7c7.html#wp1000952

[2]http://www.cisco.com/en/US/products/sw/iosswrel/ps1835/products_configuration_guide_chapter09186a00800ca7c7.html#wp1001067

[3] http://www.cisco.com/univercd/illus/2/98/22598.jpg

[4] http://www.cisco.com/univercd/illus/2/99/22599.jpg

Want to make some CA$H?

Well, don't look at us. We don't pay!

But, you could get your name in print!
(Hey, it you could be famous!)

Deadline for next issue:

September 5, 2005