

RDP Exploitation using Cain

I will demonstrate how to ARP poison a connection between a Windows 7 and Windows 2008 R2 Server using Cain.

The Microsoft Remote Desktop Protocol (RDP) provides remote display and input capabilities over network connections for Windows-based applications running on a server. RDP is designed to support different types of network topologies and multiple LAN protocols. Remote Desktop Services formerly known as Terminal Services on Windows 2000 Server allow a server to host multiple, simultaneous client sessions. Remote Desktop uses Remote Desktop Services technology to allow a single session to run remotely. Thus a user can connect to a Remote Desktop Session Host server by using Remote Desktop Connection (RDC) client software.

[Cain](#) has the ability to do a man in the middle attack against the RDP “Remote Desktop Protocol”. Cain is compiled using Microsoft Visual C++ and linked as a single executable file named “Cain.exe”. Developed with a simple Windows graphical user interface, its main purpose is to concentrate several hacking techniques and proof of concepts providing a simplified tool focused on the recovery of passwords and authentication credentials from various sources.

Windows 2008 R2 has added some remote desktop services to its server such as RemoteApp, RD gateway, and RD Visualization Host. The Windows server role now provides increased flexibility to deploy individual applications or full desktops via RDS or a VDI solution. The following is a list of Windows 2008 R2 individual applications that are available to users over the Remote Desktop Protocol (RDP)

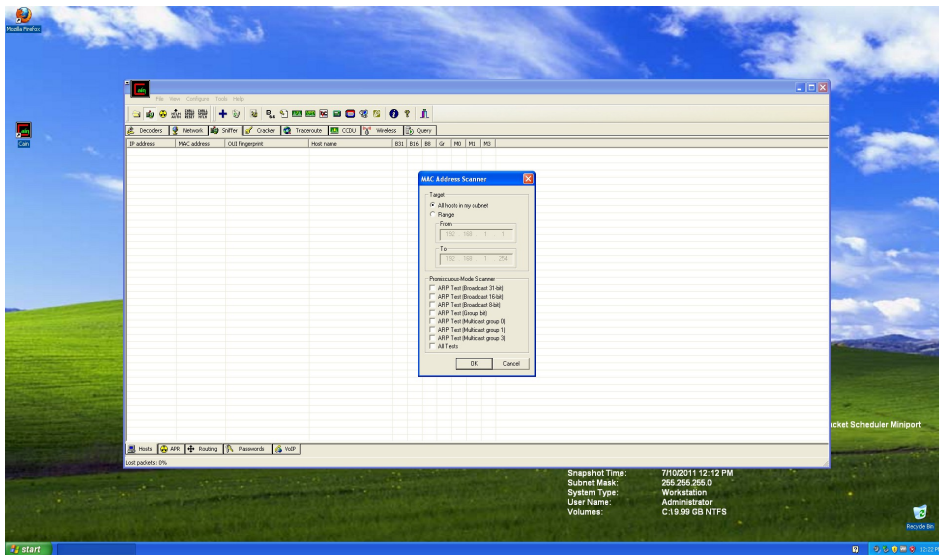
- *Remote Desktop Virtualization*
- *Remote Desktop IP Virtualization*
- *RDP & Remote Desktop Connection version 7.0*
- *Fair Share CPU scheduling*
- *Windows Installer compatibility*
- *True multiple monitor support for up to 16 monitors*
- *RDS Provider for PowerShell*

With these new improvements the door is open for potential security issues depending on how you deploy RDS. So RDS included a number of security mechanisms to help make RD connections more secure such as Network Level Authentication (NLA), Transport Layer Security (TLS), Group Policy, RD Web Access, and RD Gateway.

NLA requires that the user be authenticated to the RD Session Host server before a session is created. This helps protect the remote computer from malicious users and malware. The client computer must be at Windows XP service pack 3 or above to use NLA. Transport Layer Security (TLS) can use one of three security layers for protecting communications between the client and the RDS Session Host server:

- **RDP security layer** – uses native RDP encryption and is least secure
- **Negotiate** – TLS 1.0 (SSL) encryption used if the client supports it
- **SSL** – TLS 1.0 encryption will be used for server authentication and encryption of data sent between the client and Session Host server. This is the most secure. You will need a digital certificate, which can be issued by a CA or self-signed.

This security can be all for not with some lax controls and an effective formidable foe using Cain on your network segment. Cain runs on windows machines (Windows 9x & Windows NT/2000/XP) below is a screenshot of the interface. (see figure #1)

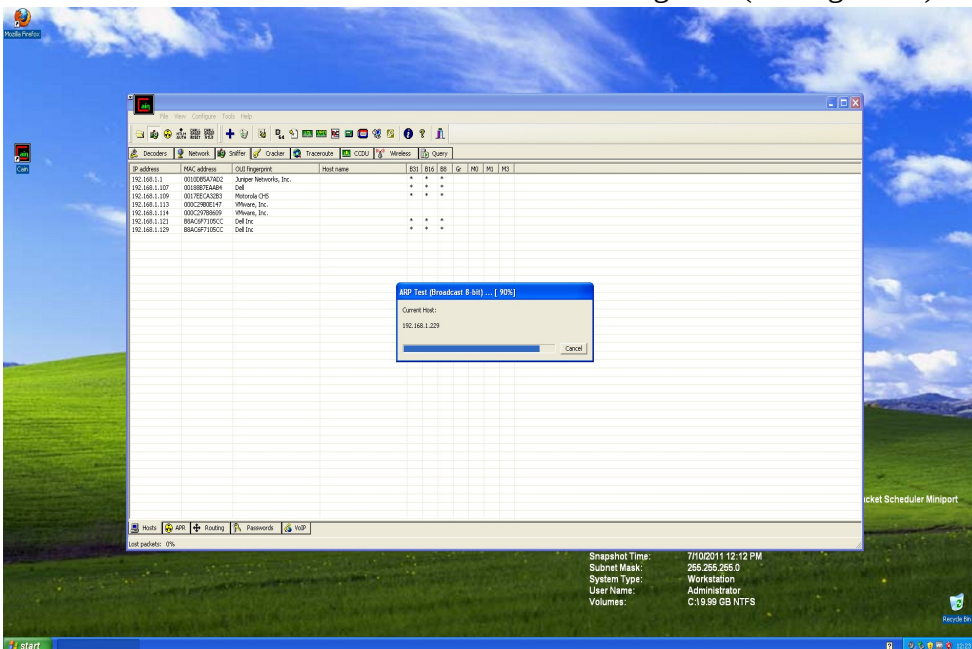


In this tutorial I have Cain running on a Windows XP machine and placed in the same network segment as a Windows 7 machine that is communicating with a Windows 2008 R2 server. Below is a list of the IP address assignments.

Figure #1 Cain screenshot

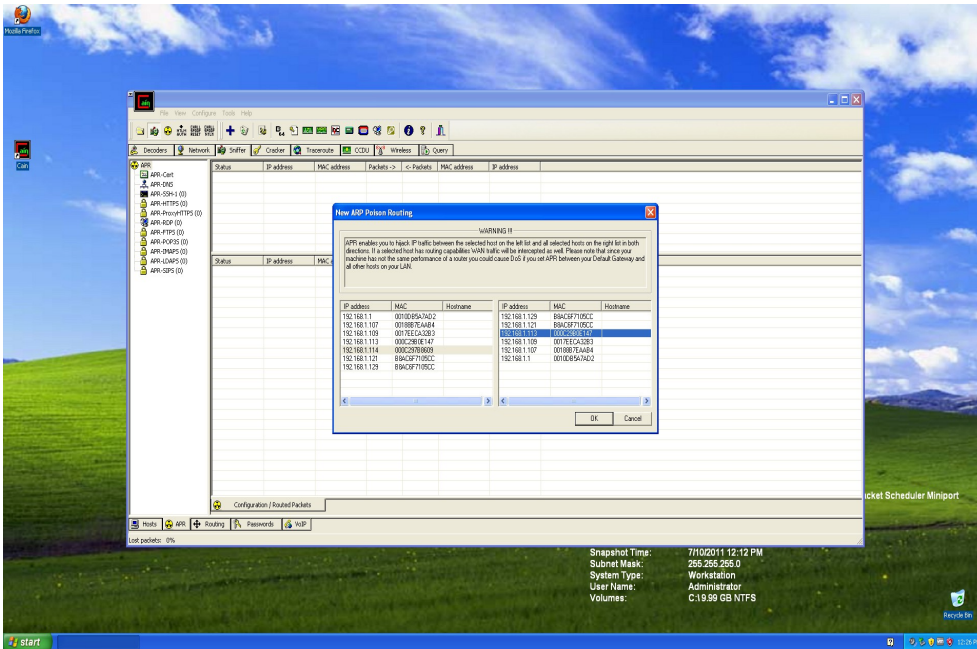
Windows 7	192.168.1.114
Windows XP (Attacker)	192.168.1.125
Windows 2008 R2	192.168.1.113

When you fire up Cain you will be given a gui interface with a list of decoders, network, sniffer, cracker, traceroute, CCDU, wireless, and query tabulated across the top. The first thing you want to do is select you interface by clicking on the sniffer icon at the top. If you have more than one interface then you need to select the appropriate interface for your work. This will start the sniffer and allow you to see what devices are available on the network segment (see Figure #2).



Next you can select what devices you would like to sniff traffic to and from (see Figure #3). Now you are ready to arp poison the network segment between your Windows 7 and Windows 2008 R2 Server by clicking on the radioactive button you will notice the status will change from idle to poisoning (see Figure #4).

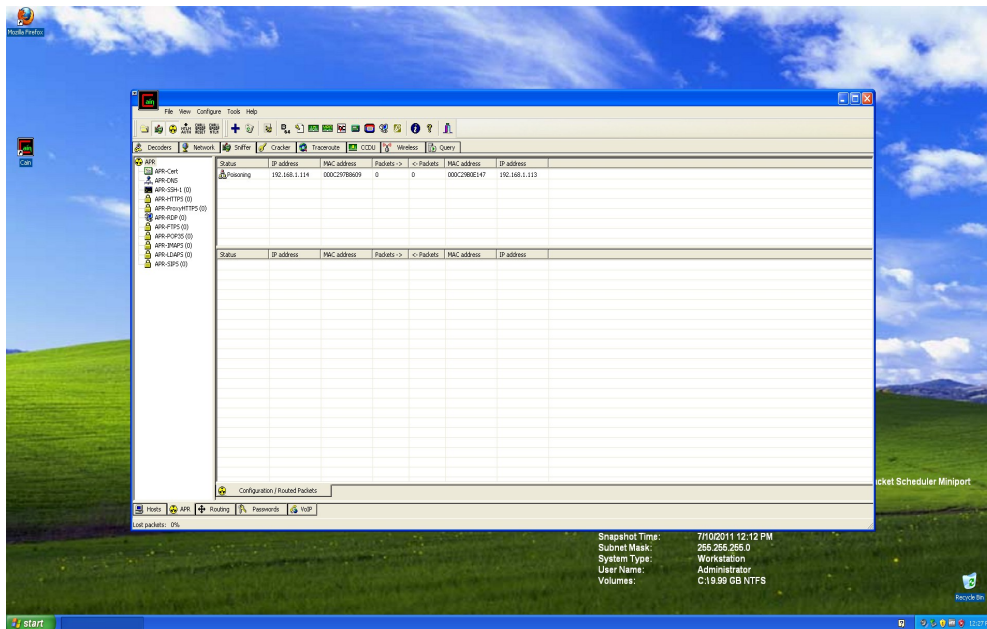
Figure #2 Devices available on network segment



Now that the attack is set we need to go to our Windows 7 client and RDP into our Windows 2008 R2 server. Now if you are doing this on a recent installation of Windows 2008 R2 server than you will have no problems within the first 120 days. After that the grace period is over and a license server needs to be installed and activated. But if your company is like most that I have worked for then money is

Figure #3 Select devices you want to sniff traffic to and from

tight and if there is a way found to get around purchasing a license all the better. A Client Access License (CAL) is required for each client requiring access to Windows Server 2008 Terminal Services. Once installed you can utilize the features I listed above to secure your RDP sessions. But if your like most organizations you can always find a way around this license requirement.



When you try connecting to the Windows 2008 R2 Server after the 120 day grace period via RDP you will get this error message if you don't have your Client Access License setup and configured (see Figure #5).

Figure #4 Status change from idle to poisoning


 The remote session was disconnected because there are no Remote Desktop License Servers available to provide a license. Please contact the server administrator.

Figure #5 error message



The remote session was disconnected because there are no Remote Desktop License Servers available to provide a license. Please contract the server administrator. Now when the client in this case the Windows 7 machine – connects to an RD Session Host server it determines if an RDS CAL is needed. The server then request an RDS CAL from a Remote Desktop license server on behalf of the client (Windows 7) and issued to the client and able to connect to the RD Session Host server (Windows 2008 R2). But windows allows for 2 RDP sessions to the host machine even without the license server. Network Administrators who do maintenance on servers (ie check status, run updates, etc) can RDP in with the admin command like so: mstsc /v: IP_or_hostname /admin
 Make sure there is a space after the IP address and before the /admin. Many Network Administrators using this function disregard the identity of the remote computer warning message and connect anyway. (see Figure #6)

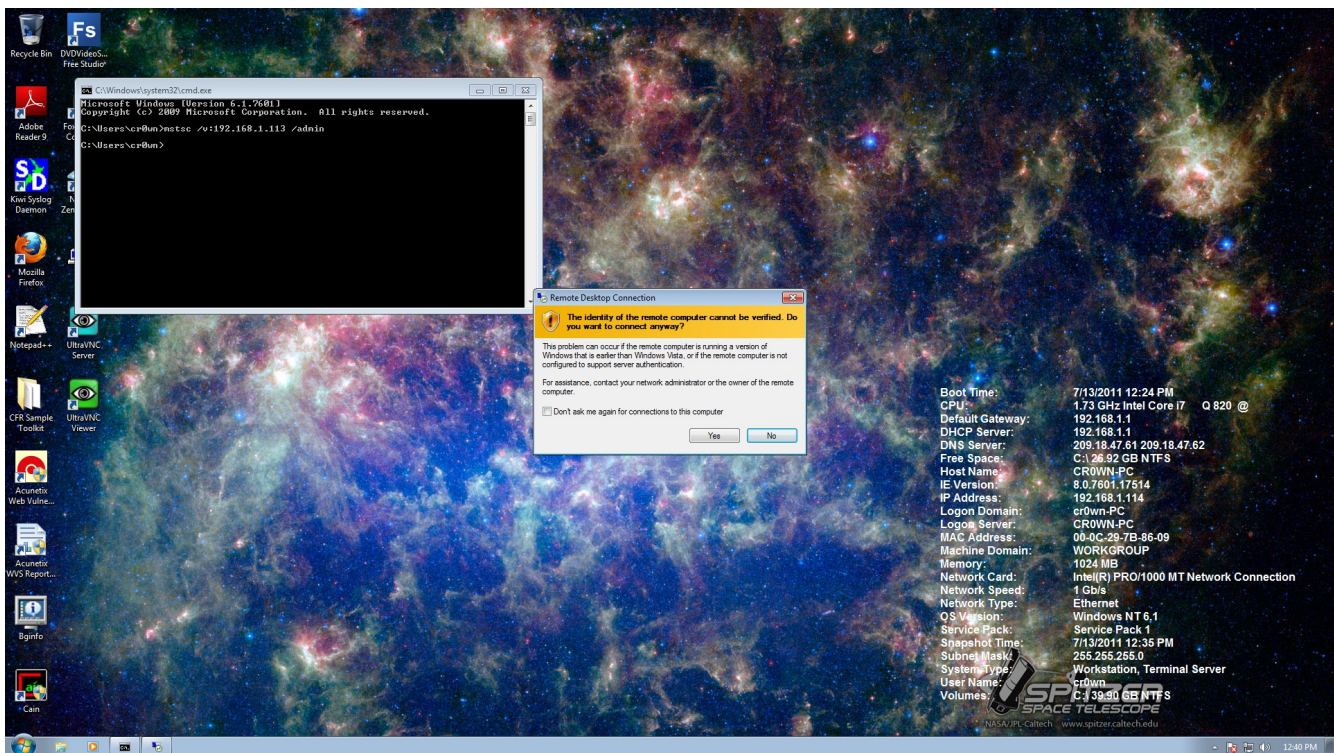


Figure #6 mstsc command and RDP warning message

Now we can connect from our Windows 7 client to our Windows 2008 R2 Server using RDP (see Figure # 7).

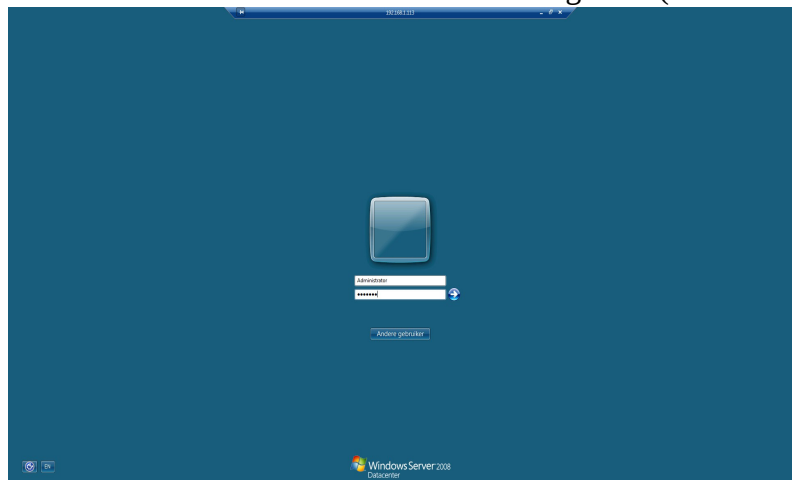


Figure #7 RDP login screen

Once we are provide the login information and password we are given access to our Windows 2008 R2 Server (see Figure #8). Now lets go back and see what information Cain has saved for us running on our attack machine Windows XP (see Figure #9)

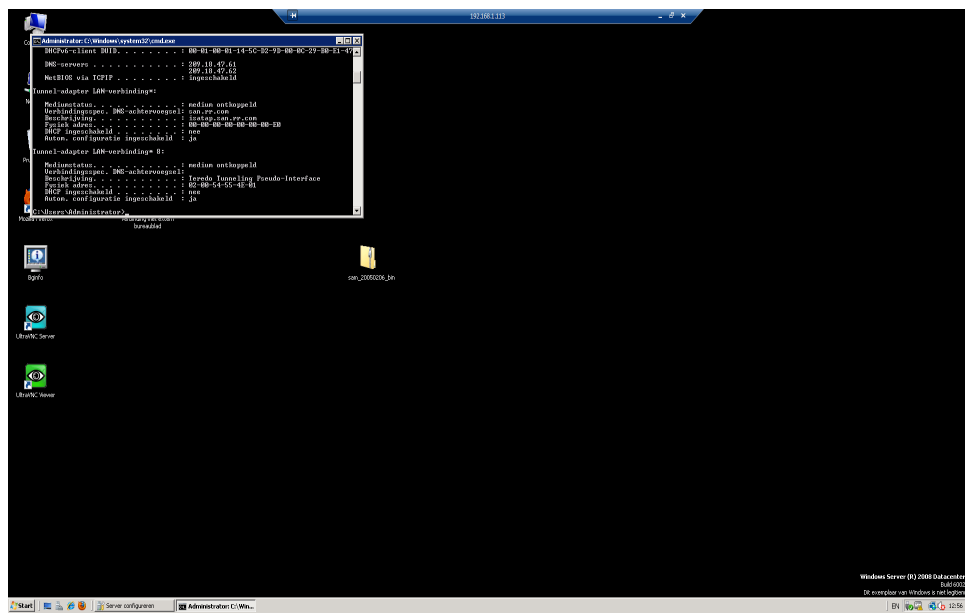


Figure #8 Administrator Desktop of Windows 2008 Server R2

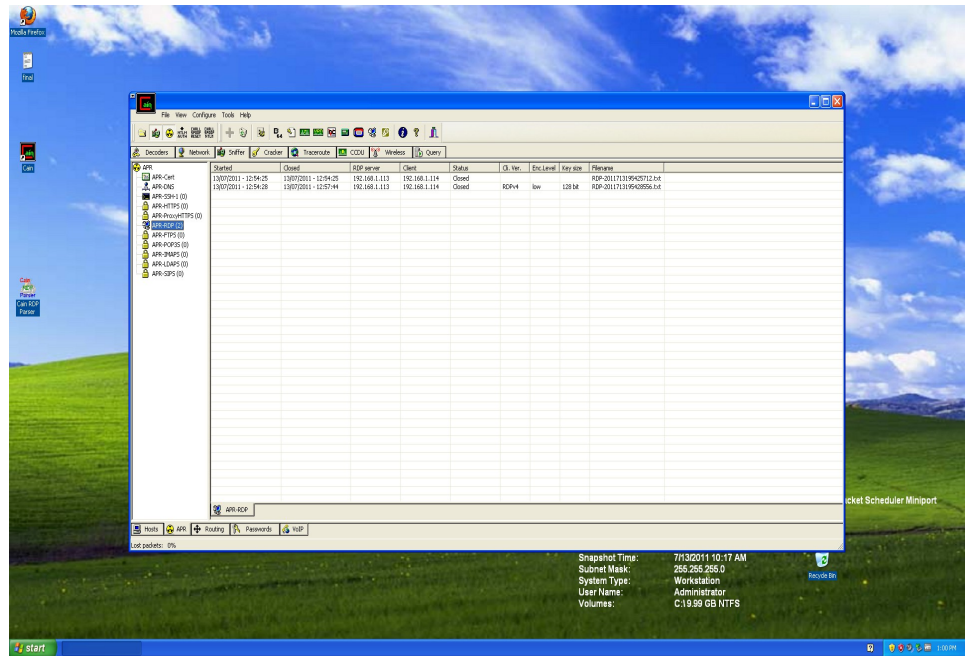


Figure #9 Files saved in ARP RDP (2)

This file that is saved pulls out keystrokes from the decrypted log file made by Cain and you can do this by hand or you can get the [script](#) by Adrian Crenshaw owner of irongeek.com. The script is written in AutoIt v3 and works quite well. Here is a screenshot of the parsed file run through Adrian's script (see Figure #10).

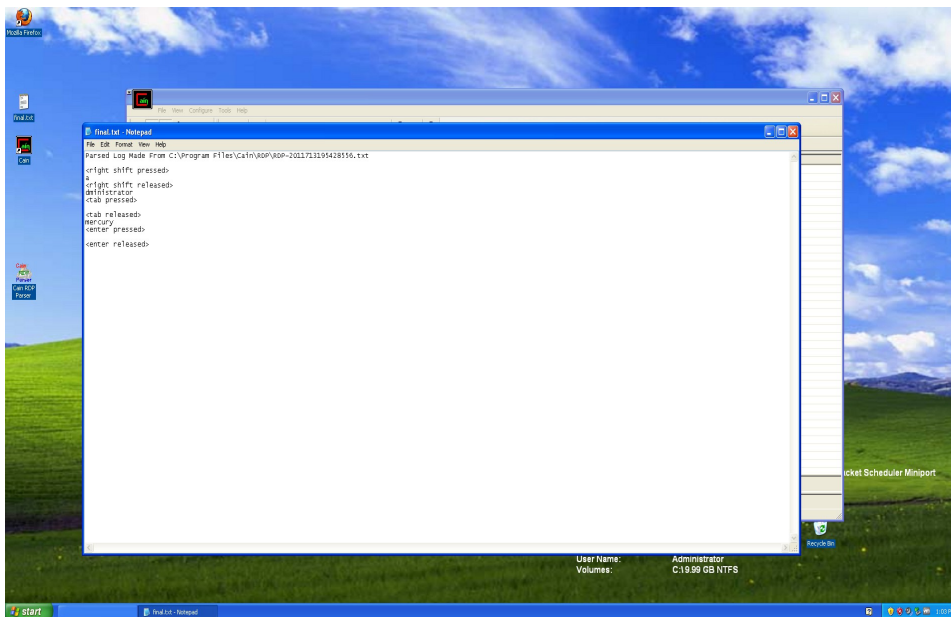


Figure #10 log parsed with Adrian's script

References on the Web:

<http://www.oxid.it/cain.html>

<http://www.irongeek.com/downloads/cain-RDP-parser.zip>

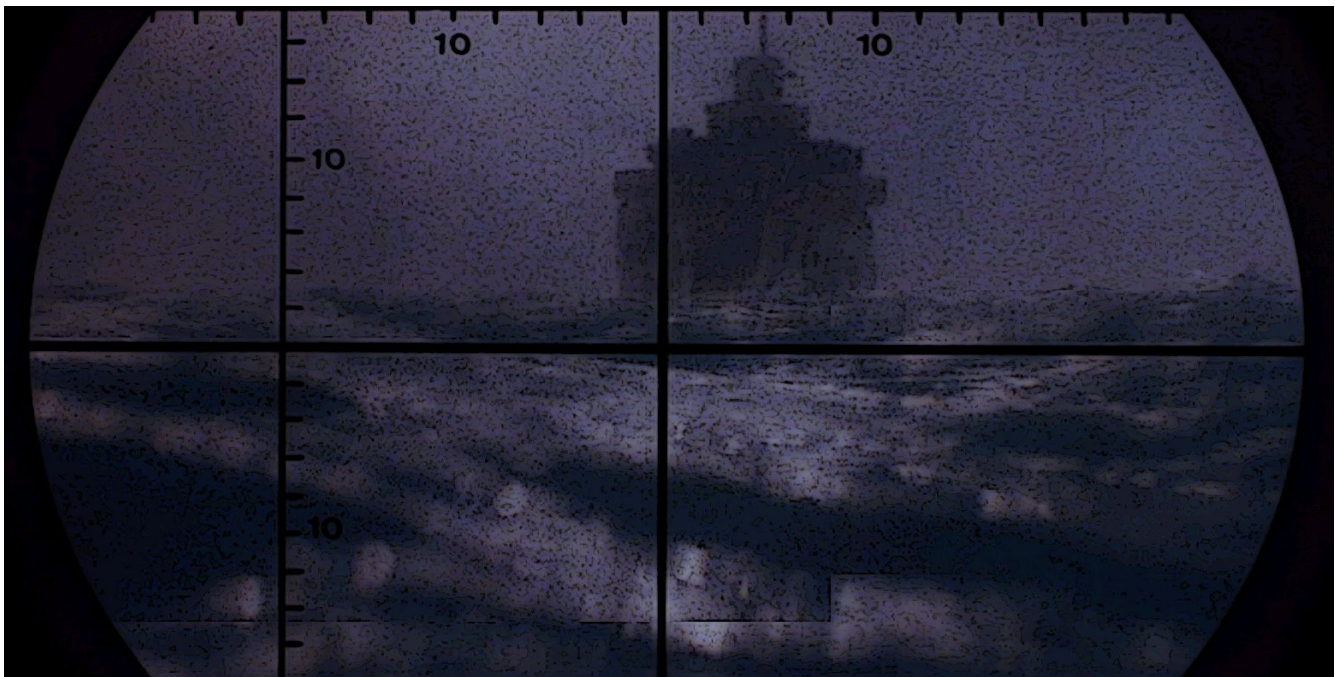
<http://pbnetworks.net/?cmd=bbs&id=37>



David J. Dodd is currently in the United States and holds a current 'Secret' DoD Clearance and is available for consulting on various Information Assurance projects. A former U.S. Marine with Avionics background in Electronic Countermeasures Systems. David has given talks at the San Diego Regional Security Conference and SDISSA, is a member of InfraGard, and contributes to Secure our eCity <http://securingoureality.org>. He works for pbnetworks Inc. <http://pbnetworks.net> a small service disabled veteran owned business located in San Diego, CA and can be contacted by emailing: dave@pbnetworks.net.



Let pbnetworks get your pen test on target



Visit us and learn how <http://pbnetworks.net>
How secure is your network?