COVERTLY BYPASSING THE FIREWALL

BLUE SKY FOR EVERYONE APPROACH

(INSIDER THREAT SCENERIO)

BY L0RDLOKI

Covertly Thinking Plan

BLUE SKY EFFORT

HOW TO FOLLOW CORPORATE/GOVERNMENT RULES

The thinking behind this is to access resources that the anal retentive firewall administrators restrict access from because they think that the proxy is saving them from diversity. As a firewall admin myself I have to think on both sides of the fence J . The information on this topic has a vast range of capabilities and will surely make those admin's rethink, "Why do we need a firewall?"

The biggest problem with some of these covert programs is that they forget that the real corporate/government world is moving toward an authentication based proxy firewall for Waste, Fraud, and Abuse or just who to point the finger at on usage at this particular time. What I am saying is "Screw That!" But in reality the applications out there do not take in the account for the authentication. Well, here is the fix for that. Most proxy firewalls authentication username and password is base64, which you can easily reproduce in any programming language you chose. If someone wanted to port this to another OS have at it. Just give "shout" credit where "shout" credit is due J . One program out there called ProxyTunnel will handle the firewall authentication but will not handle true proxy inspection of the protocol headers. This is a big problem with proxytunnel and its implementation. But it is a step in the right direction. I had the working concept going with some basic research (An afternoon of Jelly-Bellies and my friend Debbie) but the stopper was the authentication and the proxy firewall. So it stayed dormant for about five to six months. The next piece in the puzzle is sslproxy. This piece of code basically creates a valid SSL connection to any website from the command line via SSL. But still there was no capability to authenticate to the firewall. Follow the yellow brick road to gconnect. Gconnect allows this connection to happen with some setup on the server-side where the SSH daemon is running. (Please note that we only used ssh for this example. Any protocol can be used in its place. Just redirect it to the proper port.)

First problem: a Proxy firewall is checking the header information of the connection using the proxytunnel program while trying to connect to a service running on port 443. For example, if you were trying to fool the firewall that your Ssh server is really running on port 443(i.e. SSL) it will probable not happen. The error message back to the client will be similar to (SSH_Exchange_identification: Connection closed by remote host.) This error message is not directly generated by the host you are trying to connect to but from the Firewall. A true proxy is looking for the SSL-Client/Server handshake and it will drop the connection. Ummm? How do we get around this? "SATAN!!!!" <Saturday Night Live Church lady> No, Gconnet. Have your application do a valid SSL connection to a valid SSL connection. Client program à (ssl proxy to handshake) à (Optional Authentication to Proxy firewall) à the proxy does its beautiful thing and forwards the request to the SSL server à SSL server/proxy listening on 443 is redirected to port 22 for sshd. Waaa LAaa J And now the flood gates are open and blue sky for everyone.

If you want to send XWindows back and forth through the established connection just configure X11 forwarding in the sshd_config. Another thing that I notice was that if you open up your web browser to the site that has this tunnel to SSH server on the back end is that the Banner of the SSH server you are running is displayed. This was an easy modification fix for the "compile junkies." To fix this just edit the ssh2includes.h file and locate the appropriate text to modify. Make it a web page front-end if you desire. The Blue Sky and Balloons project will add other requested functionality along with cleaner code. J

XWindows are now flowing through the connection. Also if you need a different password so it is not your user id going through the firewall all the time just set up a temporary arp spoof on your system and remember to do ip_forwarding and a sniffer and look for base64 encoded hashes to the firewall or just run Dsniff. Even on a locked Switch network, it only checks the source mac address coming out of the port and not the destination. And as long as your mac address is allowed on the network. Well figure it out from there. There are some Hacking books that keep preaching that a locked switch will prevent a legitimate system that is assigned to be on that port can't do arp spoofing. Well they are wrong this time.

This is just a small educational case study and is not really intended for malicious behavior in the hands of the "Free World." Please use this program wisely to better your accesses to the Internet.

For an updated Project status and images, please go to www.digitalsynapse-ks.com for all continuing efforts of Blue Sky.

Network Layout

Figure Illustration One