TUNNA

A tool designed to bypass firewall restrictions on remote webservers

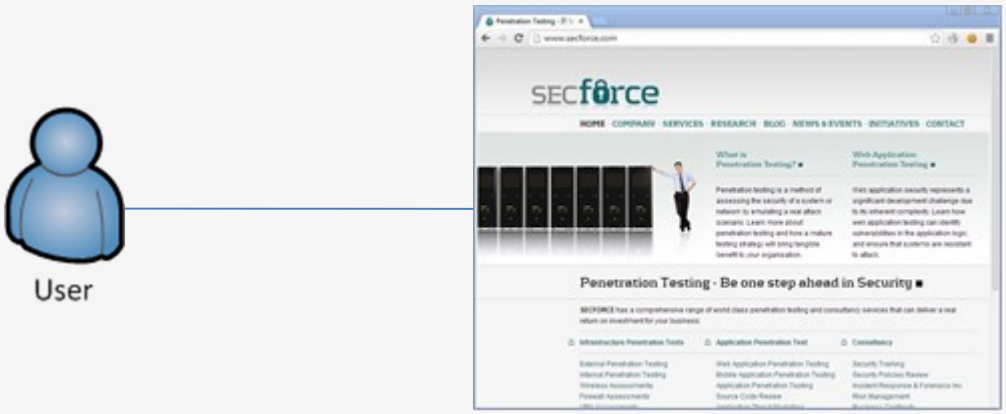
By:

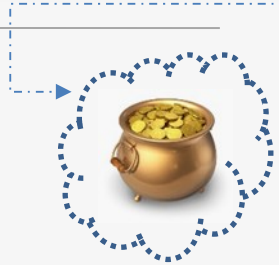
Rodrigo Marcos

Nikos Vassakis

Web Applications

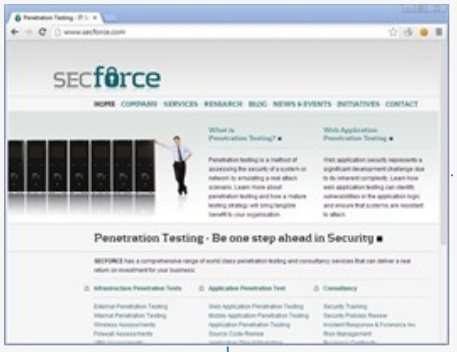
What a User sees

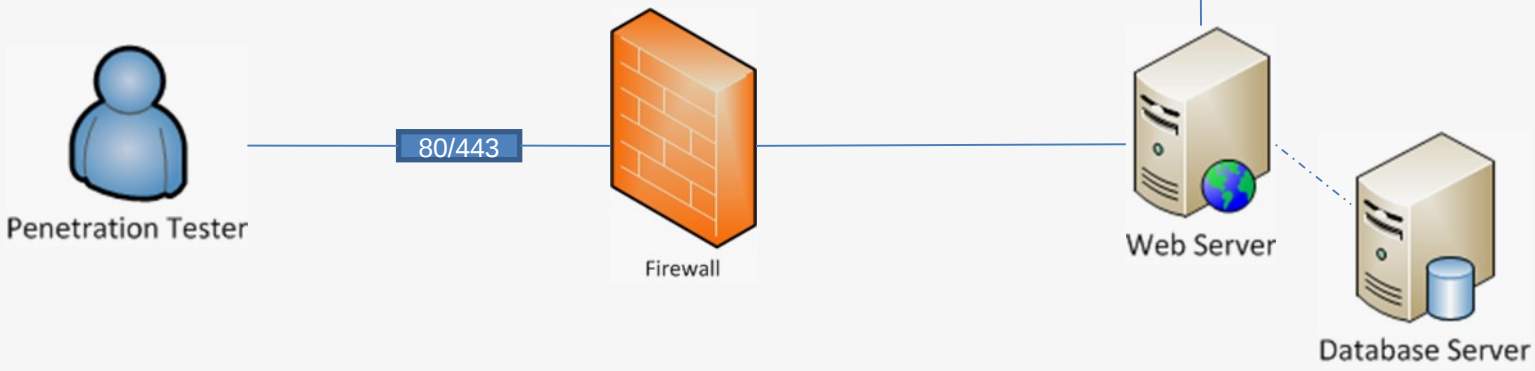




Web Applications

What a Penetration Tester sees





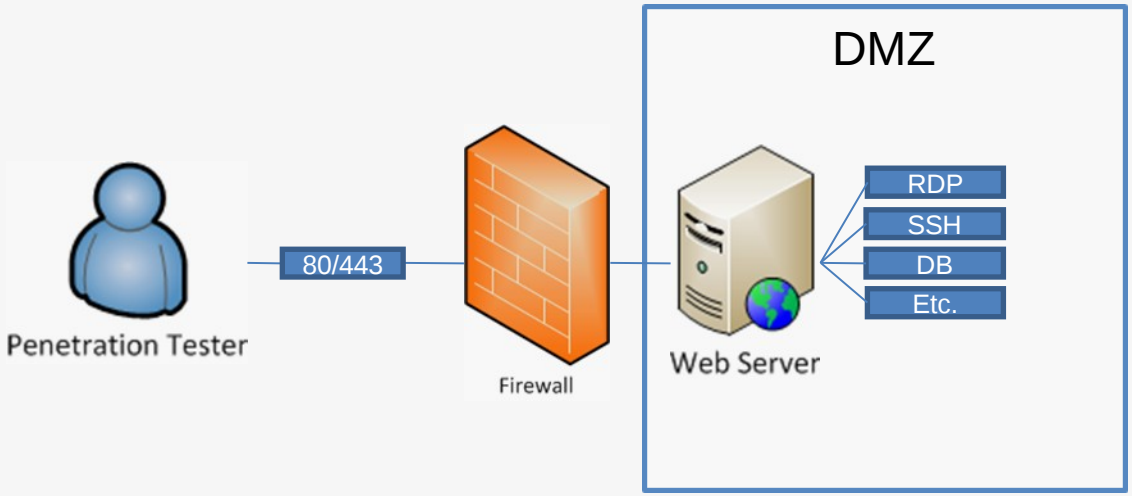
Firewall

A firewall is a software or hardware-based network security system that controls the incoming and outgoing network traffic by analyzing the data packets and determining whether they should be allowed through or not, based on applied rule set.

Web Application Infrastructure

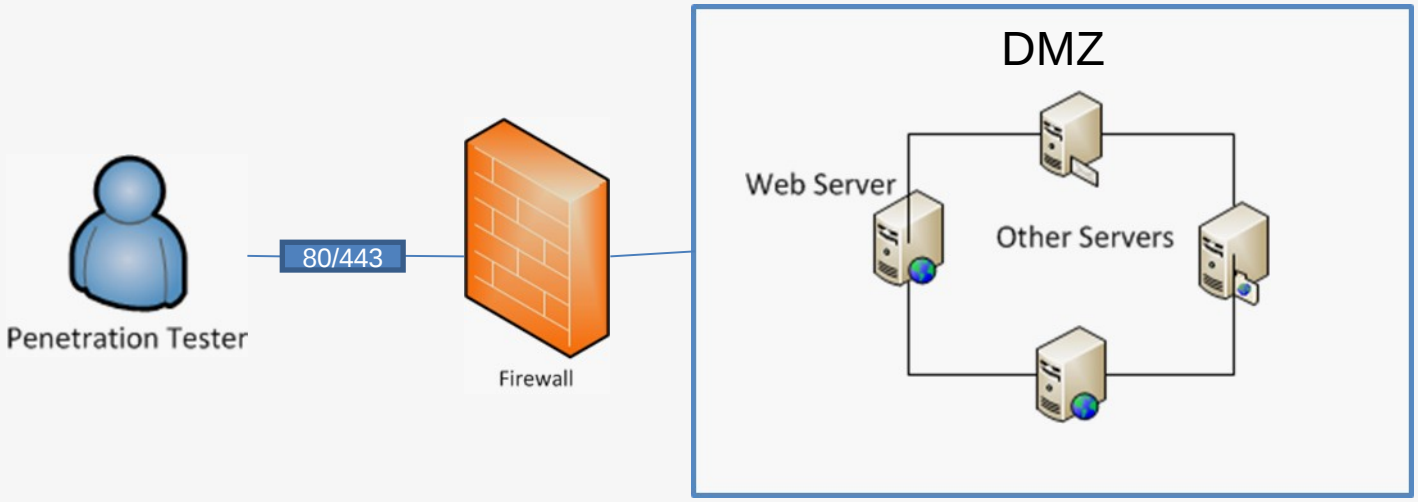
What a Penetration Tester can *"assume" ?*

The Web Server will have other services running



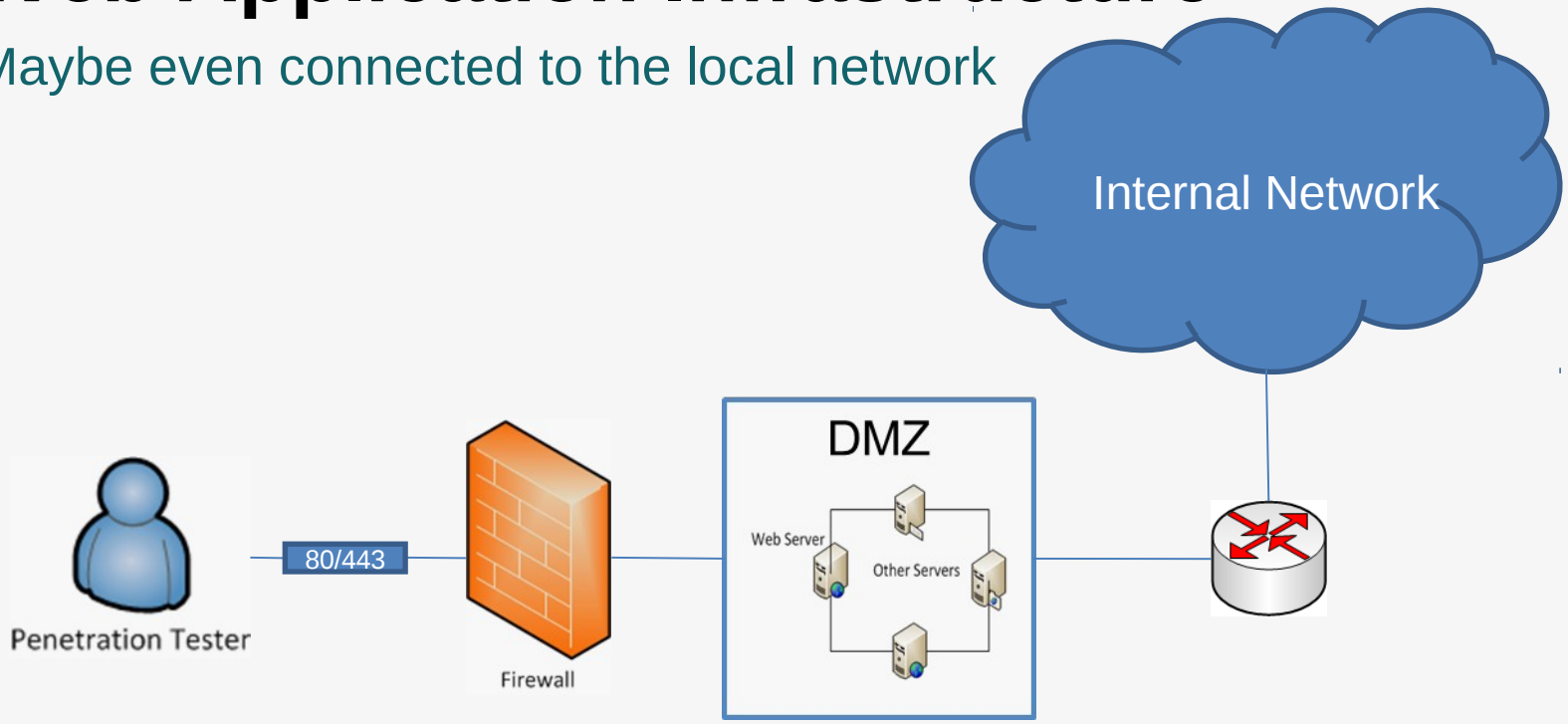
Web Application Infrastructure

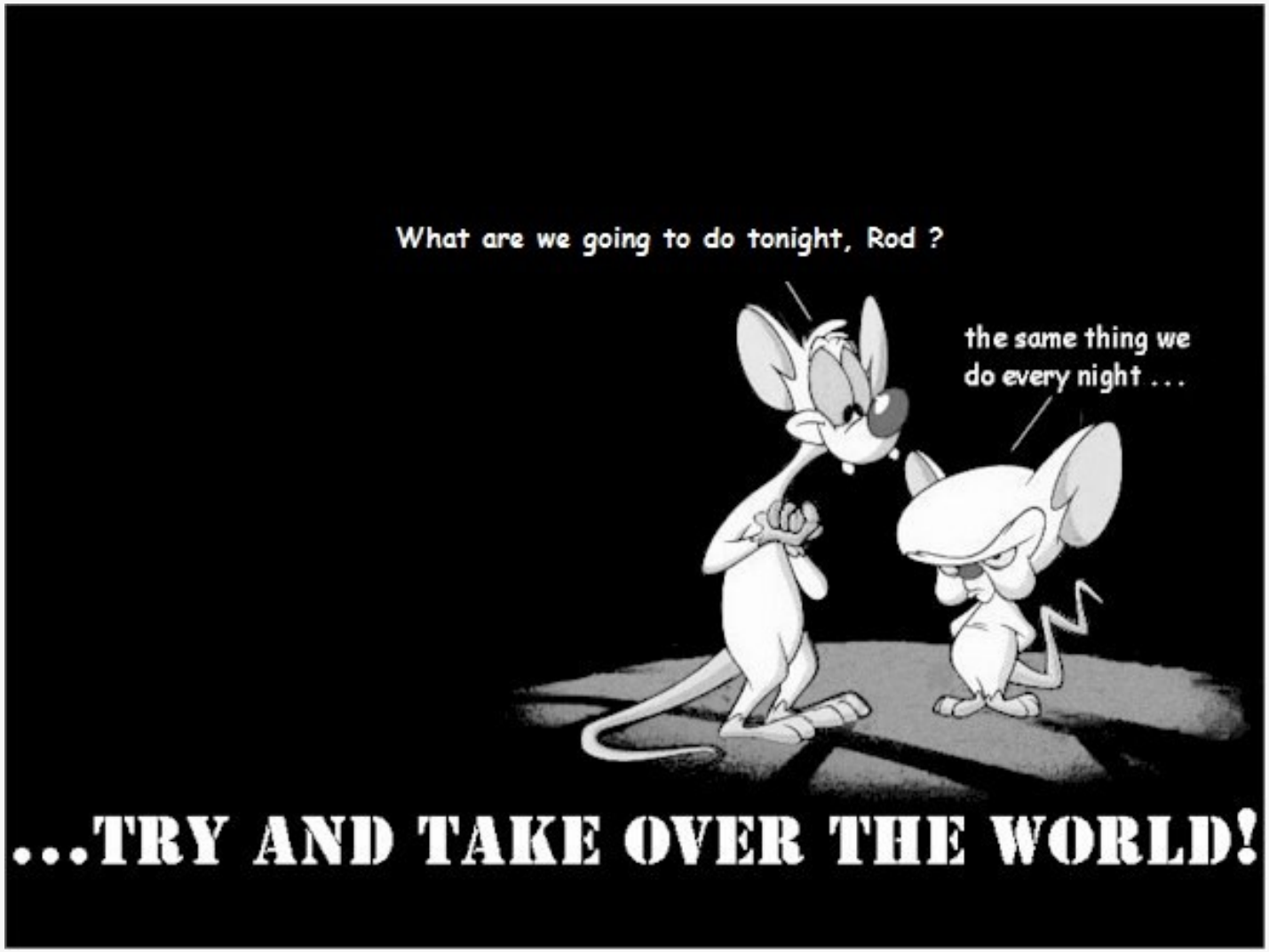
The Web Server might be connected to other hosts



Web Application Infrastructure

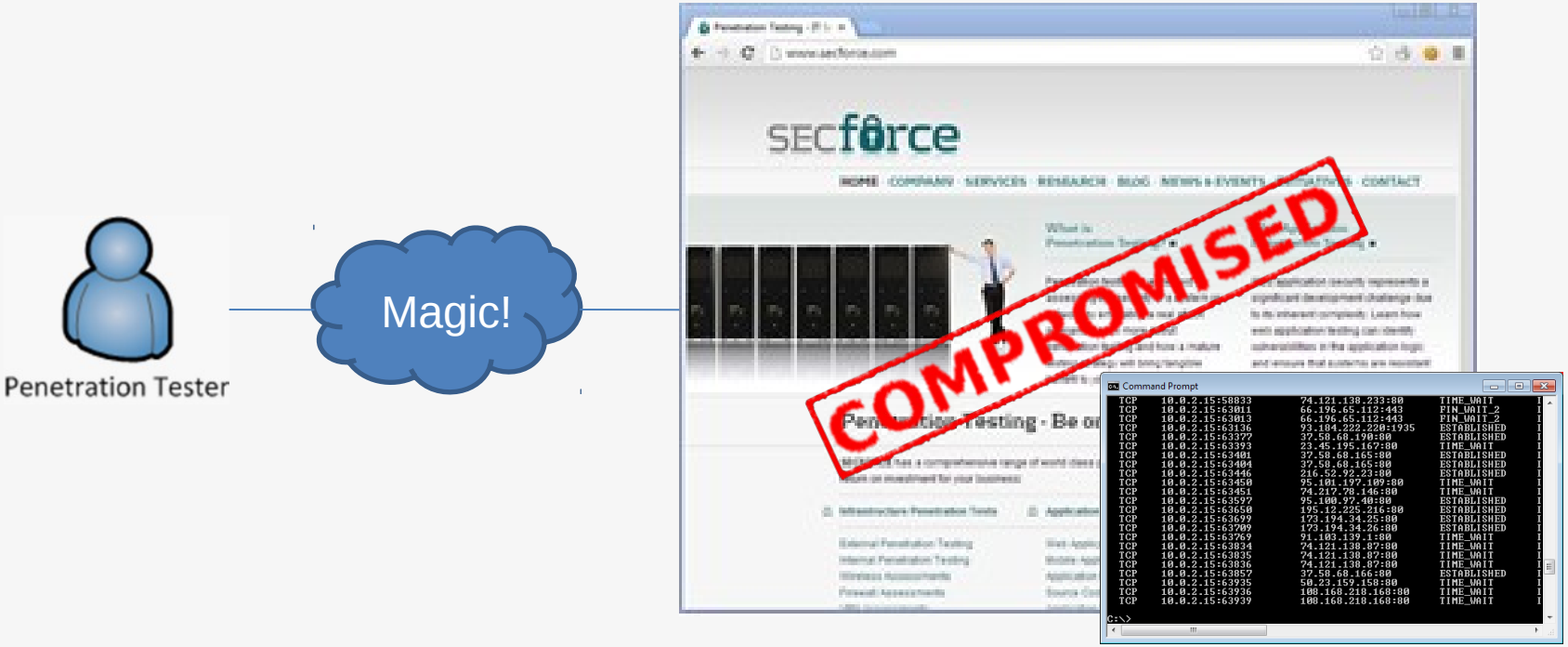
Maybe even connected to the local network




What are we going to do tonight, Rod ?
the same thing we do every night . . .
. . . TRY AND TAKE OVER THE WORLD!

The Goal!

"Don't worry, it happens to a lot of guys (and girls)"



Post Exploitation 101

Steps:

1.Upload meterpreter

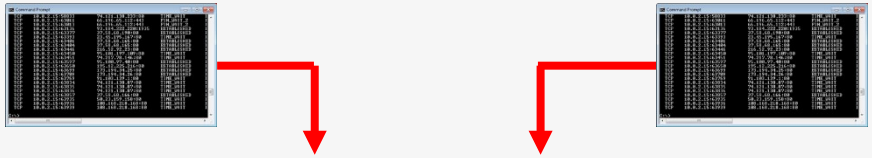
2.Run meterpreter

3.???

4.Profit

Post Exploitation

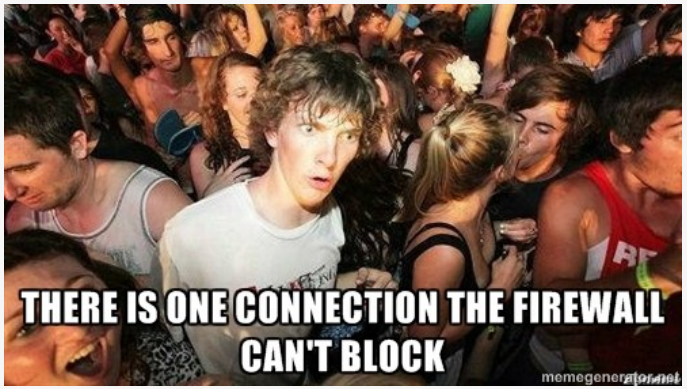
A *(*well configured*)* firewall, would block both incoming and outgoing connections to the internet from the webserver.



Post Exploitation

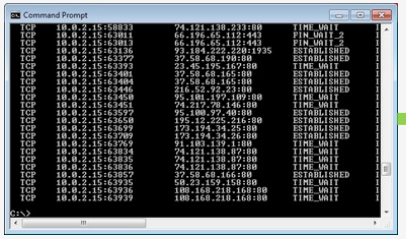
There is however one connection the firewall can't block

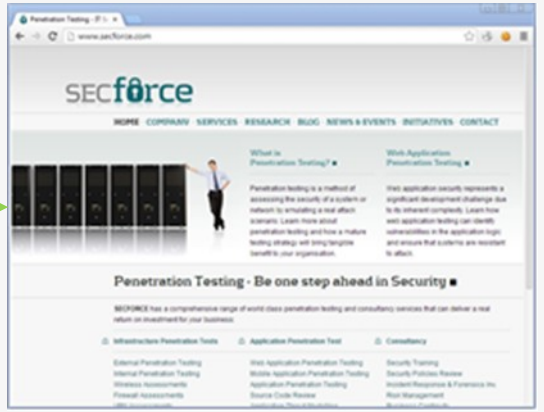
And this is to the webserver on ports 80 and/or 443 *



*typically

This will always be allowed

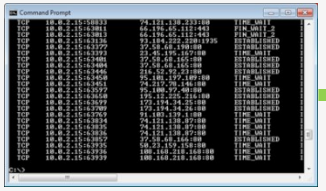




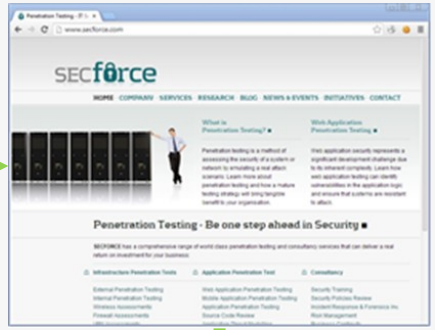
Idea

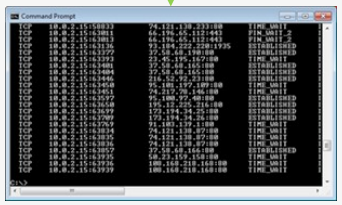
Use a web application to establish connections on the other end of the firewall

The theory



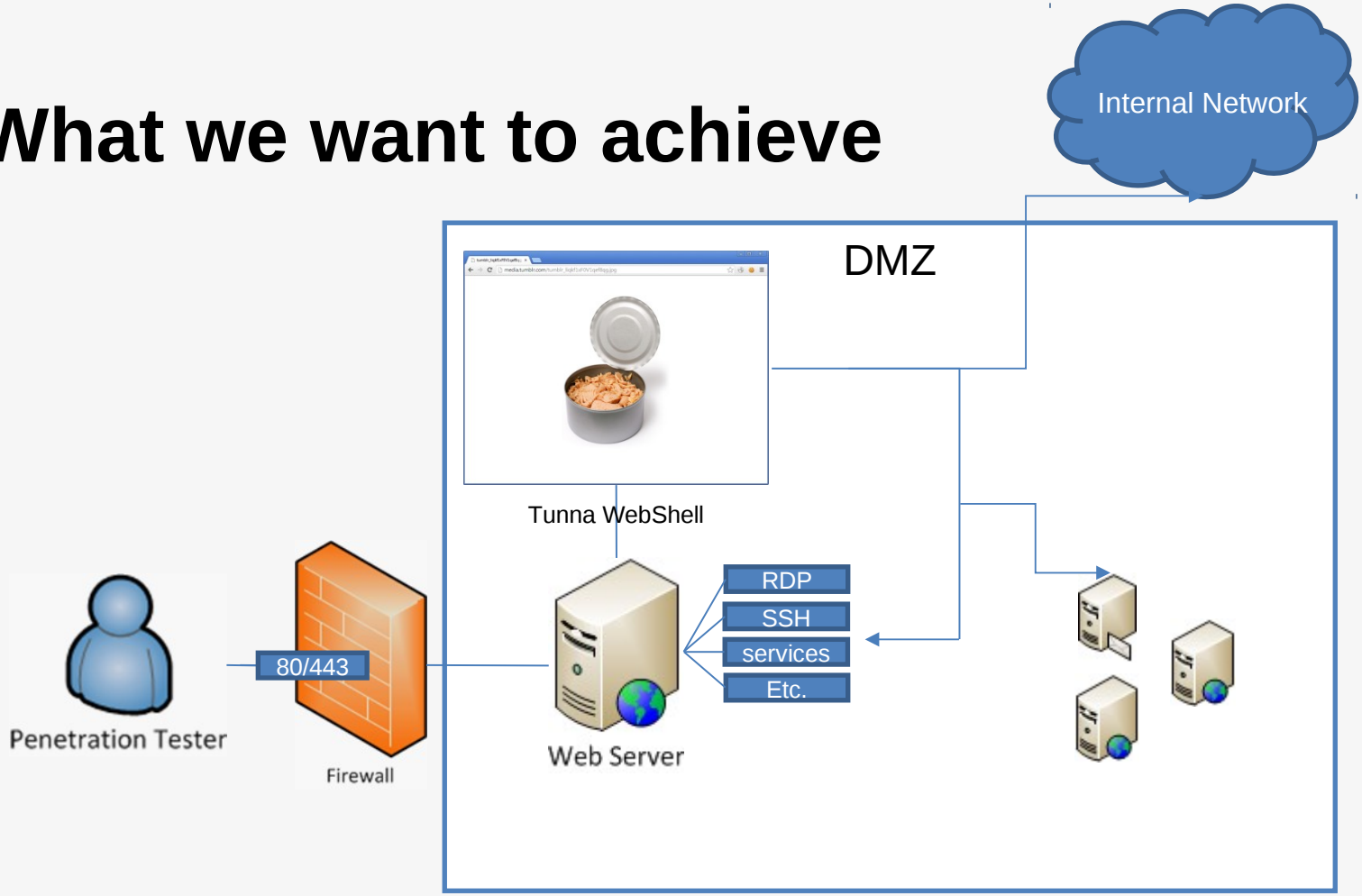
Meterpreter Handler





Meterpreter Shell

What we want to achieve

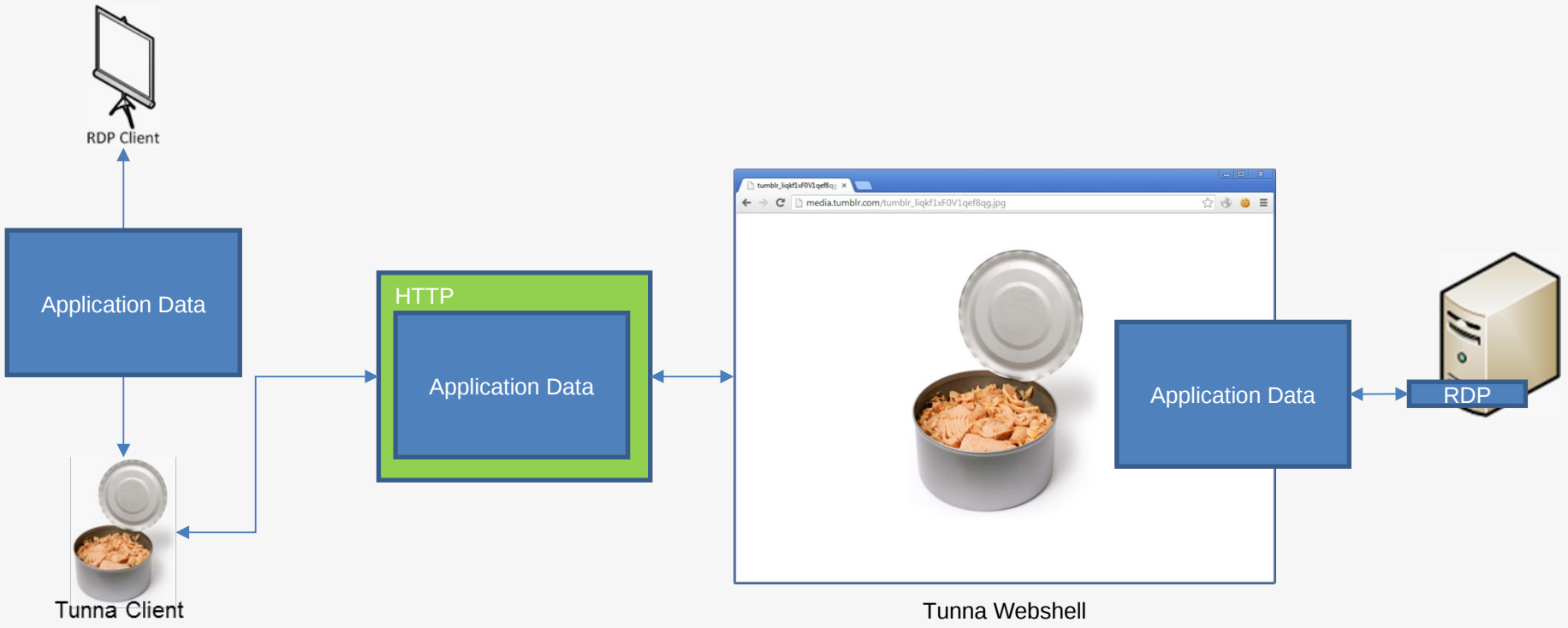


Using Tunna

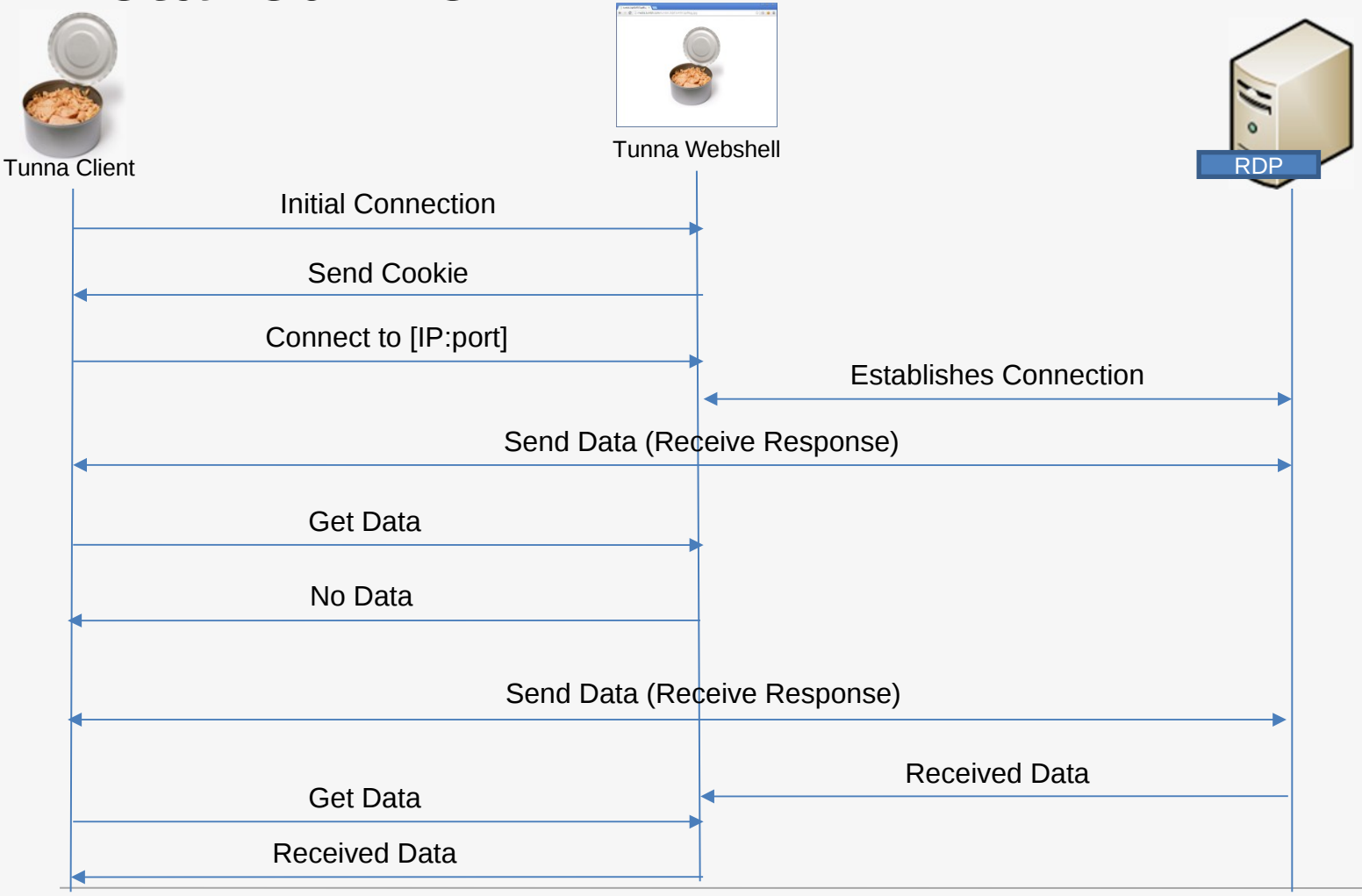
Once the “Tunna WebShell” has been uploaded to the webserver, the user can connect to any port the host can access on the internal network.

*Slide added becausefor the picture

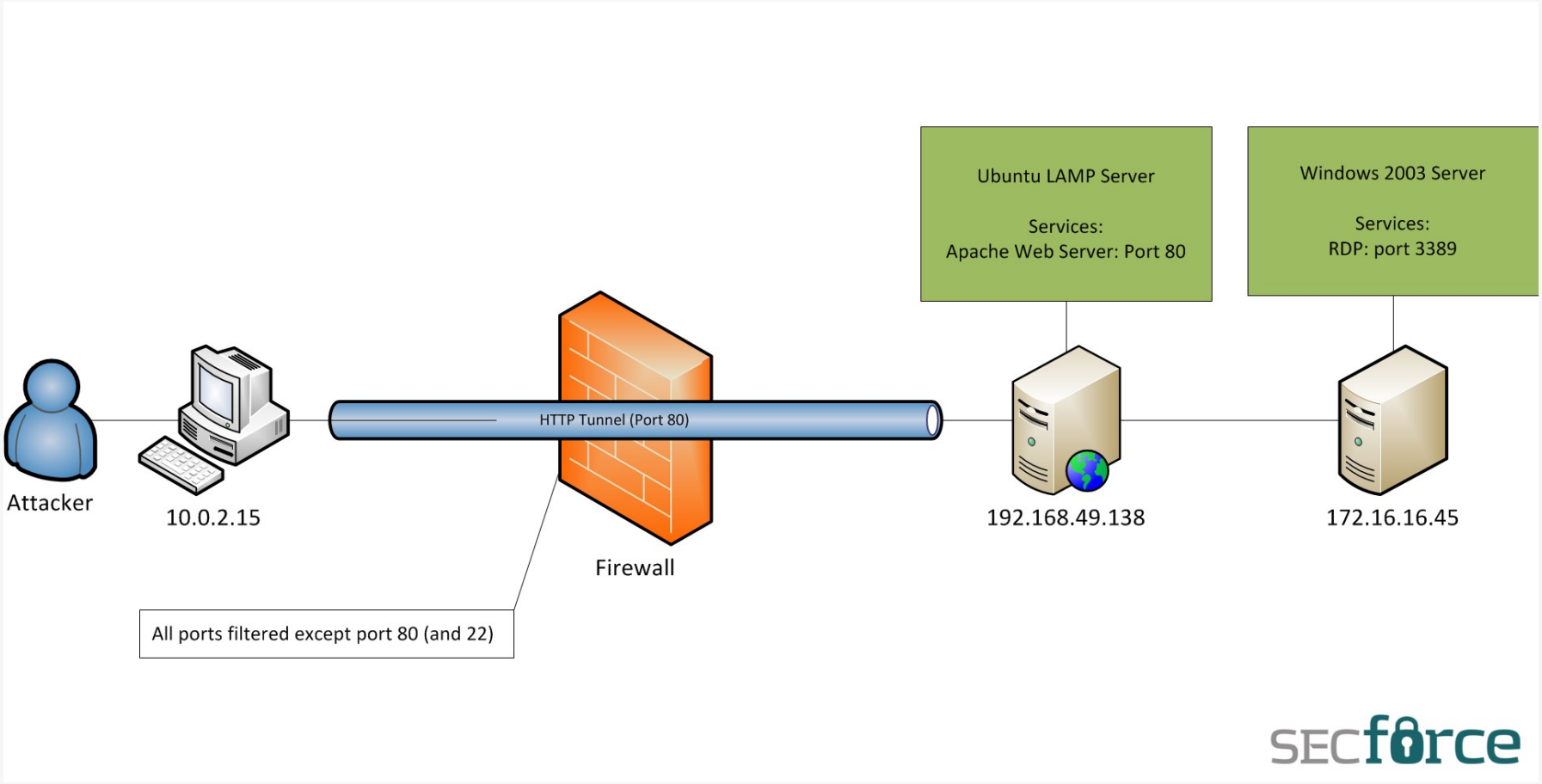
How Tunna works



Detailed View



Tunna RDP Demo



Tunna RDP Demo

HTTP tunneling with Tunna - webshell connecting to remote RDP

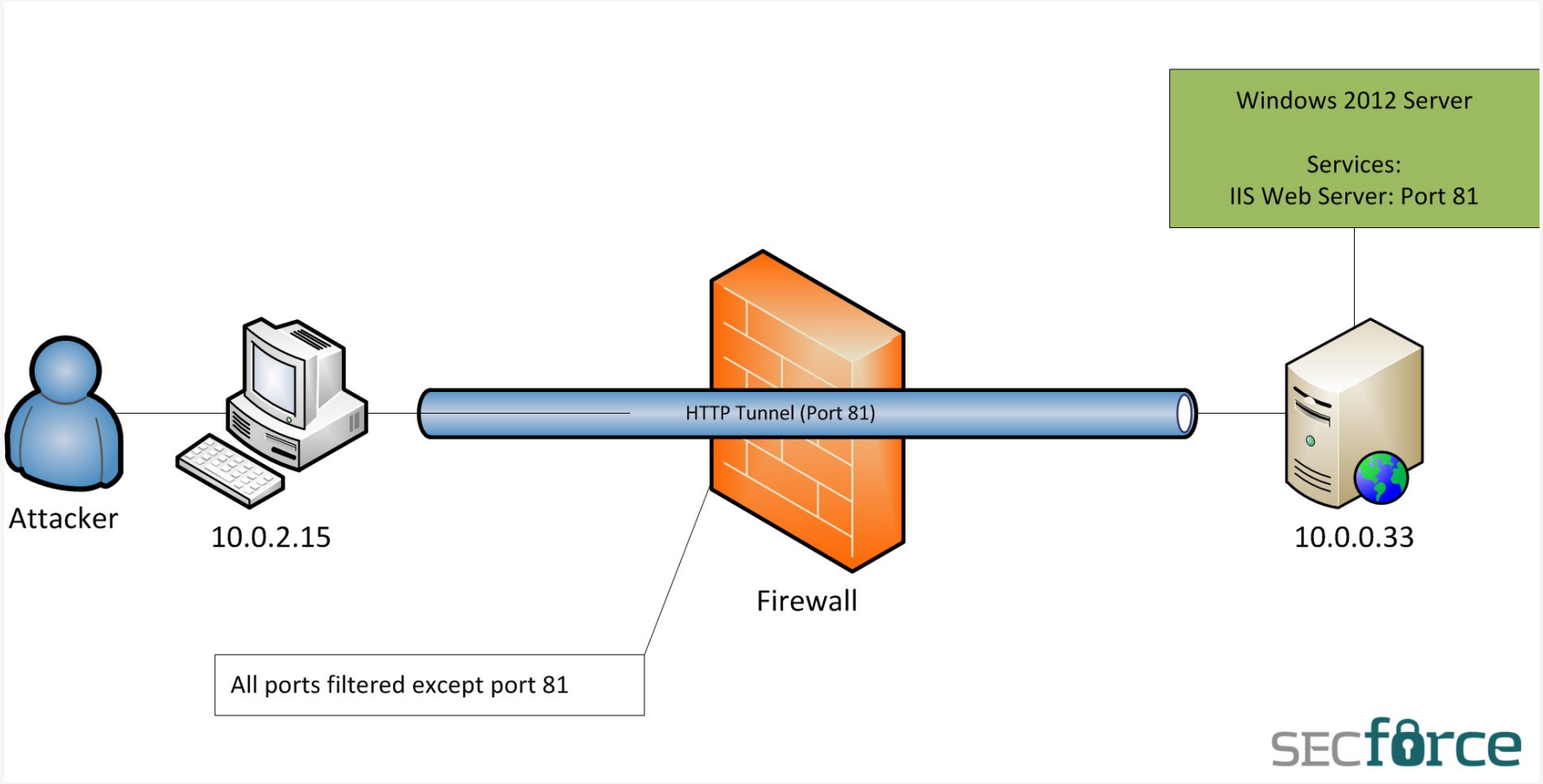
https://www.youtube.com/watch?v=Kqb1PGrkzVw

Making Things Easy

Tunna Metasploit Module:

- Creates a meterpreter listener that listens on a local port
- Uses “Tunna WebShell” to transfer meterpreter to the remote server,
- Executes it and
- Connects to it

Metasploit Demo



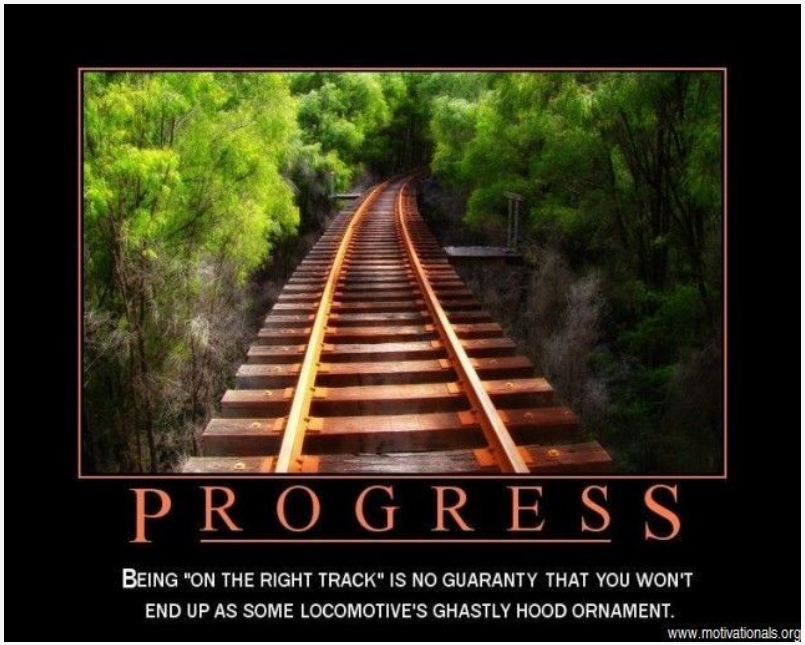
Metasploit Demo

HTTP tunneling with Tunna - metasploit module example run

https://www.youtube.com/watch?v=-Svxx7OVfQY

Tunna Version 1.1

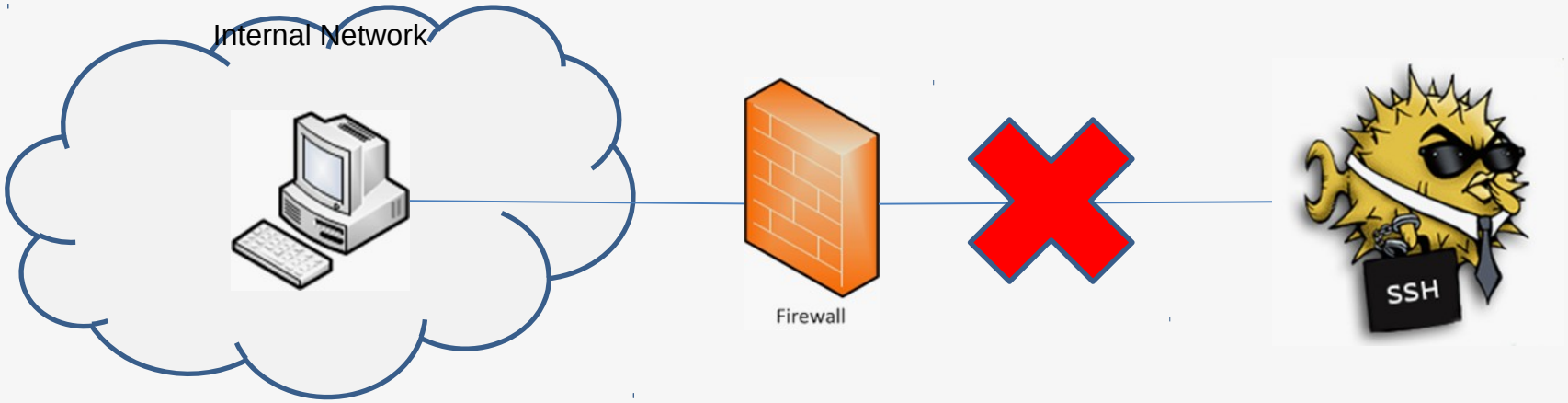
Opening a new can of Tunna



Breaking Out Tunna

The Problem:

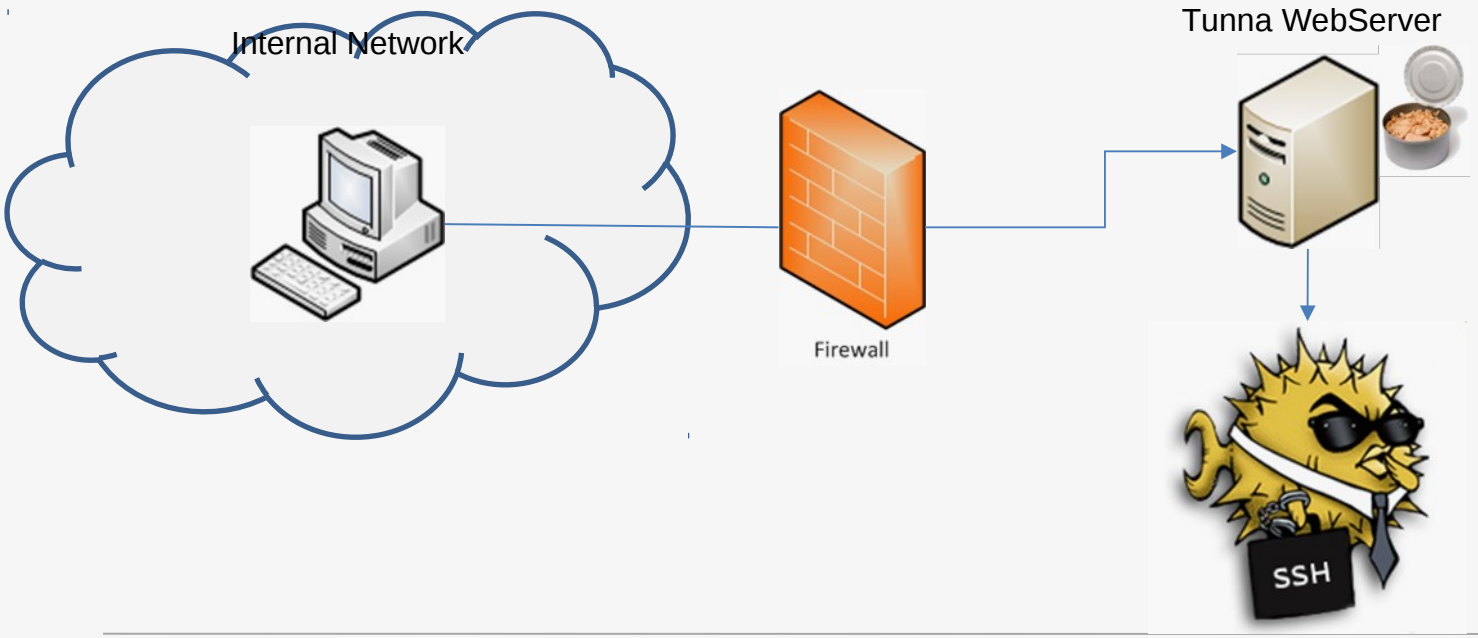
Internal firewall blocks certain services and/or sites



Breaking Out Tunna

Typically Internal firewalls block traffic based on the service or IP/DNS name of the remote host

Tunna can be used to pivot the connection to the remote host



Breaking Out Tunna

To ease this scenario a standalone “Tunna webserver” was developed. A webserver like Apache or IIS is not required.

Proxy support was also added to “Tunna Client” for situations where an internal proxy gateway is present. Tunna will use the internal proxy the same way the browser does and will channel all traffic through the proxy.

Limitations

The first version of Tunna had one limitation.

- It could only tunnel a single connection to a single remote service.
- A new Tunna instance was required for a second connection.
- *However, third party software like SSH or a meterpreter shell could be used along with Tunna to tunnel multiple connections*

Socking Tunna

Due to popular demand, the new version, Tunna (v1.1a) can be set up to be a local SOCKS proxy

Only SOCKS version 4a* is supported but works great for most scenarios!

**Note*: SOCKS BIND method is not yet supported*

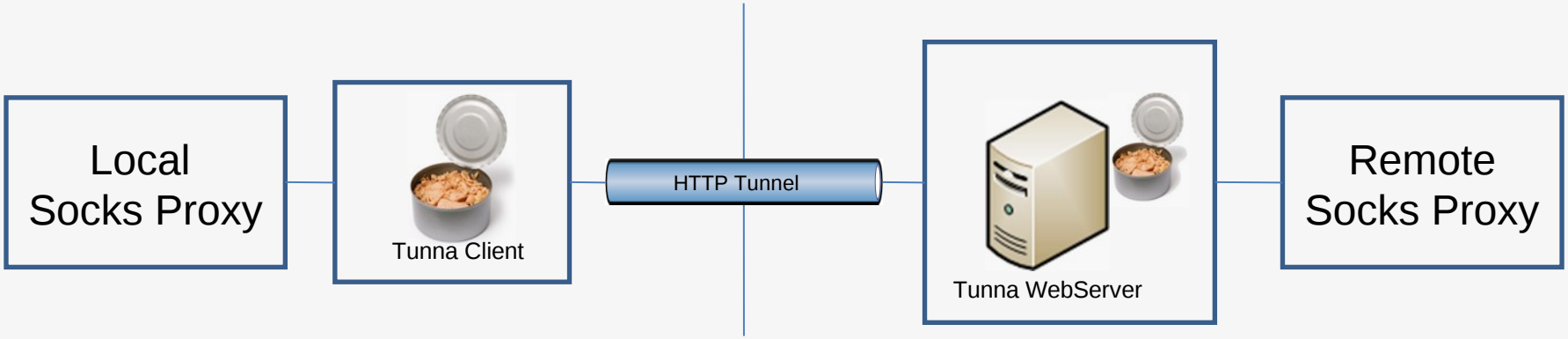
Split SOCKS 4a Proxy

The local applications connects to the local “Proxy Server” everything is transferred to the remote “Proxy Server” over a single connection

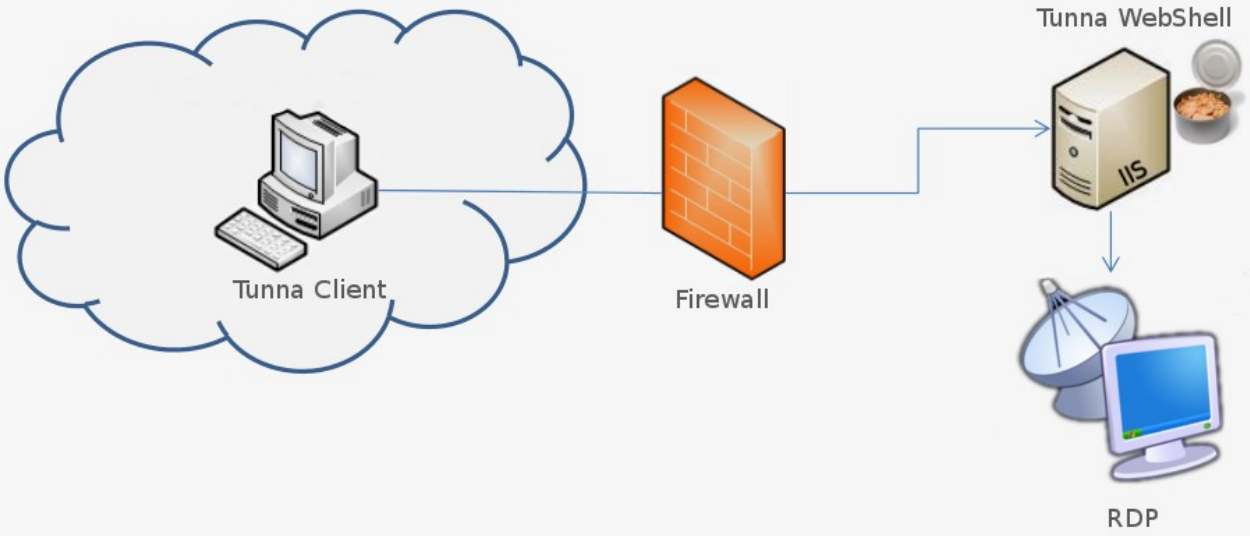
It works by tracking every connection but its transparent to the applications using it. It’s just like using a SOCKS 4a proxy.

SOCKS Implementation

- The applications connect to the local “Socks Proxy”
- Everything is forwarded to “Tunna”,
- Is transferred to the remote “Tunna Webserver” and
- Forwarded to the “Remote Socks Proxy”



Tunna SOCKS Demo



Tunna SOCKS Demo

HTTP tunneling with Tunna v1.1a using proxychains

https://www.youtube.com/watch?v=tyWTicaUD1k

Secondary Additions

Tunna Binaries for Windows are included in the new version (no need for python to be installed).

- Tunna Client executable
- Tunna Server executable

Settings.py file has been added to ease setting up the client

**Note: All Tunna client binaries or python scripts can be used with all the different webshells or the Tunna Webserver (binary or python script) the same way.*

⚠ Word of Caution !

Tunna generates a massive overhead for every TCP packet

Consequently, large amounts of traffic translate to large amounts of HTTP request.

This can lead to a Denial of Service condition where the webserver/network devices etc. will not be able to cope with all the requests*.

It is also recommended for Tunna webshells not to be used as a permanent solution.

Some functionality is still experimental.

**Tunna standalone webserver is not affected at the same level.*

Future Plans

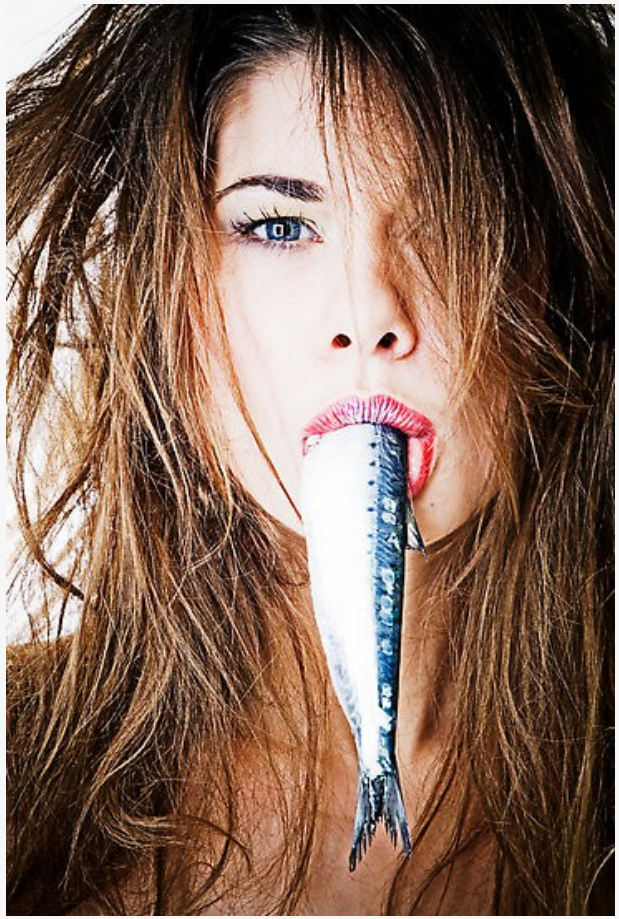
1. Add Authentication to Tunna
2. Add support for SOCKS v5
3. ???
4. World Domination

Tunna SUCKS!

… but it is still in development and is getting better with every release!

Thank you for listening!

…and watch this space:
http://www.secforce.com/blog/



**No animals were harmed during the making of this tool*