Putting the Tea back into Cyber Terrorism

SensePost Research

BlackHat Briefings
July 2003
Las Vegas USA

This project by
Roelof Temmingh [roelof@sensepost.com]

Index

1 Cyber Warfare / Terrorism - introduction

The difference between terrorism and war seems to be a thin line. In South Africa we see this very clearly. Our previous president, who is generally a nice guy, was called a terrorist by the previous government. Some of the more intense soldiers in service of the previous government are now called terrorists. Some would call an incident an act of war – others may call it an act of terrorism. For the purpose of this paper we do not try to distinguish between the two. The modus operandi remains the same: pick a target and destroy it.

For some time now, the effectiveness of cyber attacks has been widely debated. When limiting such attacks to those that can be launched from the Internet the following comes to mind:

- Denial-of-Service attacks
- Breaching network perimeters remotely (e.g. hacking in)

Both of these methods seem to be rather ineffective. While Denial-of-Service is not pleasant, it remains largely ineffective against internal network infrastructures. Organisations that do not heavily rely on the Internet for operations (e.g. military) or core business functions, this type of attack is not a major concern, as they can easily unplug from the Internet without adversely affecting their ability to conduct 'business'. The second attack could pose a huge problem for any single organisation, but chances are slim that a large enough number of companies within the same sector are owned. Even with the discovery of 0-day vulnerabilities, attackers rarely have a good understanding of *where* to attack.

With the creation of a worm, making use of a killer 0-day, it targets ALL hosts and not hosts specific to an industry, sector or country. Creating a worm, that utilises 0-day, to target a specific country is however an interesting concept.

In this paper we discuss another approach to cyber warfare/terrorism. All of the attacks referred to in this paper discuss attacks that could be launched remotely and over the Internet.

2 A rather nasty worm

2.1 Introduction

It is generally understood that worms can be much more advanced than most of their current manifestations. Worms generally break a few months after a new vulnerability has been found. Malicious coders slap together code that exploits this vulnerability and add a propagation method. The worm runs free in the wild for some weeks, where after infections decline steadily as administrators start fixing Internet facing servers. Some worms have the ability to target internal networks, which in most cases cause massive damage to internal infrastructure, even if they were not designed with a Denial-of-Service components in mind.

Administrators rushing to patch affected servers do a pretty good job in fumigating their own backyards. As soon as internal infections have been contained and the perimeter secured against the particular vulnerability, administrators generally don't waste too much of their time on it. The internal network is rarely kept adequately safe for the following reasons:

- New servers/workstations are added (without the correct patch levels).
- Servers are rebuilt with outdated software.
- Network segments that were not affected by the worm, for whatever reason, most likely remain vulnerable.

- Most worms exploit a specific vulnerability – and administrators tend to merely address that particular problem.
- Administrators have a tendency to rather spend resources on building a strong perimeter, in the form of anti-virus and content scanning technologies, in order to keep worms out than patch every internal host.
- Many of the internal networks can more directly be ascribed to “gross negligence” and mis-configuration, than to specific vulnerabilities themselves.

The focus of the security industry is slanted towards external perimeter security. Internal networks are predominantly soft targets. IT security can be compared to an igloo – hard on the outside and soft in the middle. In all the internal network assessments SensePost has performed we have always found, to some degree or other, “low-hanging fruit” vulnerabilities that allowed us full host compromise.

The following is a list of vulnerabilities found on just about every internal network:

- Microsoft IIS (5) Unicode / 2x decode
- Microsoft IIS (4) MSADC
- Microsoft IIS (5) .printer extensions
- Microsoft IIS (5) WebDAV
- Microsoft SQL with blank SA configured
- Blank local administrator passwords on Microsoft Windows hosts
- Apache Chunked Encoding
- OpenSSL < 0.9.6

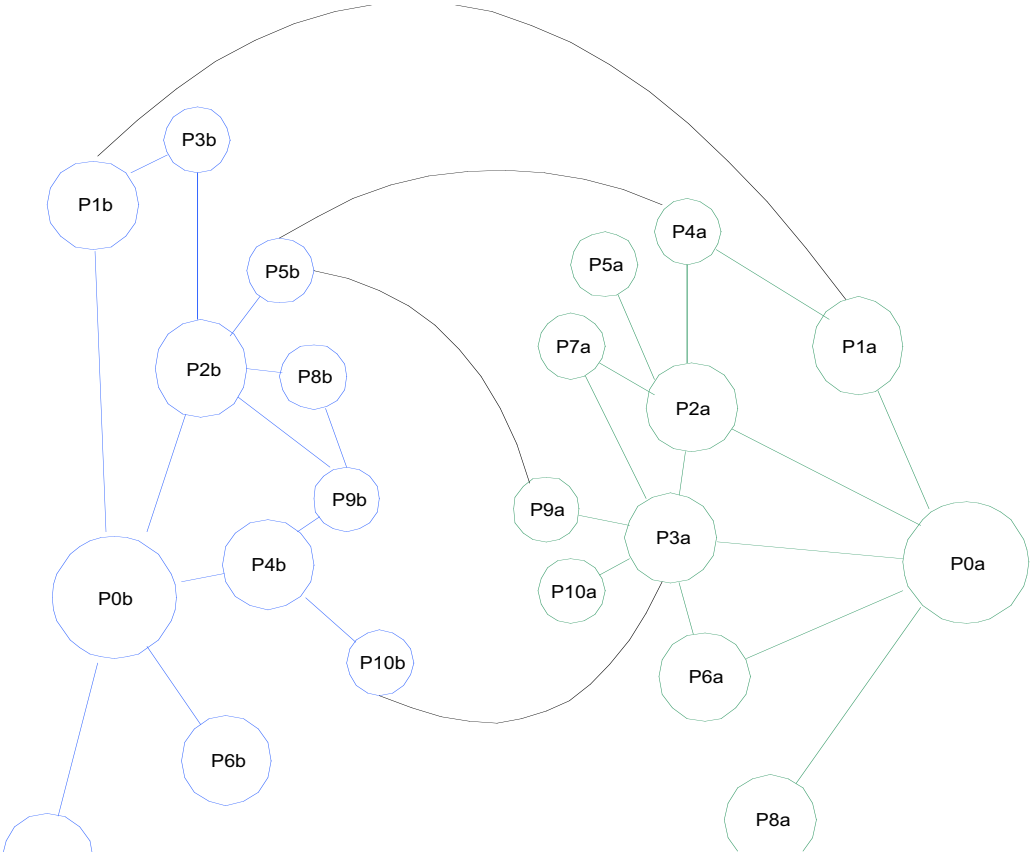
Many of the above vulnerabilities have been around for years, but there are very few administrators, in charge of managing large network spaces, that would be prepared to bet money on these problems not existing on their networks. All of the listed vulnerabilities would provide an attacker with remote command execution on the target host. Most people would agree that a worm based on all of the above mentioned vulnerabilities (and launched internally) could infect a great deal of hosts. Combined with a Denial-of-Service payload such a worm could have the ability to completely disable a large internal network.

When designing a worm that is to operate exclusively on an internal network, one has to keep the following in mind:

1. Targeting of new victims is markedly different from Internet-based worms.
2. The worm will propagate at wire speeds. Propagation in itself can cause Denial-of-Service and may possibly even hurt the worm’s spread.
3. The Denial-of-Service component of all worm instances need to be synchronised to a large degree (refer to previous point).
4. Inter-worm communications may be possible, as internal networks are most likely not properly segmented.
5. By performing EXE file infection hosts unaffected by the mentioned vulnerabilities can also be infected.

2.2 Inter-worm communication

A problem of special interest, as mentioned above, is the coordination between different instances of the worm. If one instance of the worm should start a Denial-of-Service attack it could prevent the worm from further disseminating into other parts of the network. Assuming that inter-worm communication is possible, a “message” type routing protocol between instances of the worm could be the answer to this problem. Have a look at the diagram below:



In the above diagram we assume that the worm has two different starting points – we call it patient 0a and patient 0b (P0a and P0b on the diagram). We also assume that worms will not re-infect hosts, but that an infected host will report itself as infected if another instance of the worm attempts an infection. We introduce the concept of a neighbour. A neighbour is:

- The host that infected you
- Hosts that you have infected
- Hosts you have been in contact with that have already been infected

The diagram above shows an infection in progress. Patient 5b has found that patient 4a and 9a has already been infected. They are therefore regarded as neighbours to 5b.

Patient 3a's neighbours are 10a, 9a (we infected them), 0a (she infected us) and 7a, 2a, 6a, and 10b (we found them to be infected when we started probing for other host to infect).

When a new infection occurs, the originating host broadcasts a message to this extent to all its neighbours. After the infection the newly infected host is added to the "neighbourhood". On hearing about an infection from any of the neighbours, the message is broadcast to all other neighbours. In this way all worm instances become aware of other instances. When not receiving new updates for a predetermined amount of time, lets say 10 minutes, we assume all possible targets have been infected. Now we can flatten the network by launching an all out DoS.

2.3 Denial-of-Service

Denial-of-Service attacks on internal networks are usually much more effective than their Internet equivalents. Flooding can occur at wire speeds. It is well known that routers and switches fell

over when previous worms (which contained no targeted Denial-of-Service components) penetrated perimeters. The recent one packet SMB Denial-of-Service attack (SMBKill) would have disastrous effects (ask anyone that was on the receiving end of a techie in their organisation joining the dark side). Flooding the network with bogus ARP replies, hijacking TCP connections (on unswitched networks) and DHCP exhaustion attacks are just some of the Denial-of-Service attacks that only really come to life on internal networks. It is beyond the scope of this paper to detail these internal Denial-of-Service methods.

As each of the instances of the worm knows about all the other instances, the DoS component of the worm can be smart enough to avoid disrupting other instances of the worm.

Additional Denial-of-Service components (internal to the affected host) could include:

- Inject 10 random bytes randomly into all Microsoft Office files as well as .zip (and possibly other) files. Database files should be targeted as well.
- If possible, change BIOS settings – possibly add a BIOS password (highly unlikely) or simply flash the BIOS.
- Show a pop-up box that displays a message such as "*You need to provide your sysadmin with the following string in order to re-enable your computer: 5V+g0! ^35Cvd93_ssd112X2*". This is used simply to flood helpdesks with telephone calls, effectively causing chaos and wasting resources within the IT Support areas that need to be concentrating on addressing the true problem. The sheer volume of calls may even crash phone systems.
- Determine if network devices (such as routers, switches, hubs etc.) are configured with default usernames and/or passwords. If so, change the administrative password. This part of the payload could be written to only support the top 3 types of network devices –e.g. 3com, Cisco (use commonly used passwords/usernames) and Bay. This prevents network administrators from applying ACLs to contain the infection.

2.4 Internal targetting

There are four means the worm could use to determine the boundaries of the internal network:

- Obtain the IP address and mask of the current host, from where the current subnet can be determined. A host with multiple interfaces is a bonus.
- Send SNMP queries, by using default and commonly used SNMP community strings, to all hosts on the current subnet. Extract the routing table and all interfaces. Large networks tend to run with routing protocols (such as OSPF) that yields lots of networks. Internal routers often still make use of default SNMP community strings.
- Traceroute to IP addresses located on known Internet based IP addressed, in order to record the route. Subnets can be determined by inspecting IP addresses encountered along the way. This method may not yield as many subnets as one would like, as firewalls along the path might be blocking ICMP.
- Ping IP addresses located one class C below and above the current subnet. For example if our subnet is 10.0.10.0/24 try 10.0.9.0 and 10.0.11.0. This method can be extended to include "brute forcing" as a last resort.

3 Delivery

A worm like the one described above would have a very limited lifespan on the Internet. Most vulnerabilities listed above do not exists in large numbers on Internet facing servers. The challenge is to deliver the worm in such a way that it is executed on a client located in the internal network. A very simple, yet highly effective, way to achieve this is to simply mail the worm to a number of people. Unlike the usual email-borne viruses the worm should not be attached to the

mail itself as such worms are quickly caught by content level filters. By emailing a single link to a downloadable executable are also not as effective, as intelligent content filtering devices pick up on these types of HTTP-based downloads as well. The use of encryption however bypasses this problem. When using URL obfuscation, as to appear that the EXE is located on an internal server, and SSL, as well a message that sounds as though it's from the company's marketing department (social engineering component) usually has the wanted effect.

You really can't blame a non-technical person working for COMPANY X for following instructions from *marketing@companyx.com* in an email with subject "*New <COMPANY X> screensaver – Click on Open once downloaded*" that contains a link to *https://intranet.company.com|document=%43%4020%39%2e6%31%2e18%38%2e%339/SS.exe*
.
What about all the pop-up windows that will appear complaining about site and certificate not matching? Users ignore it. Mail headers? Users don't read it. How about the system warning users that they are about to execute foreign code? Users tend to be intrigued by their company's screensaver –really…

The attack sounds very basic – where's the 0-day silent delivery remote execution for Outlook? Where's the buffer overflows? A buffer overflow might work on some specific versions. This method is so basic it works everywhere and with any email client. In many cases the most effective attack is really simple. The attack relies on the user's naivety – possibly the weakest link in computer security today.

To test our theory all members in an IT security team of a prominent South African bank were mailed, with the consent of their manager of course. When run, the executable we used extracted the username from the environment, opened an invisible IE browser and navigated to a site under our control, calling an HTML file with the username as parameter (this was butchered from Setiri's code – see BlackHat USA 2002).
The findings were as follows:

- 13 people mailed
- 8 people downloaded the EXE (60%)
- 5 people executed the EXE (38%) (one person executed 3 times...:)

As the EXE is virulent only one person needs to download and execute – the worm will find its way to all internal vulnerable hosts. If 5 members of an IT security savvy team in the financial sector executed an in-your-face EXE, how many marketing, sales or management type people would do the same?

4 Targetted delivery

How do we find email addresses for a specific company? Ever tried finding someone's email address when you know they are working for company XYZ.com? As usual Google is your friend. Scrape Google for *+@XYZ.com –www.XYZ.com* – and find everyone at XYZ.com that ever posted anything to a mailing list, a guestbook, or a newsgroup.

It's a no-brainer – and highly effective. Let's take a very arbitrary example – Hurriyet Newspaper in Turkey has domain *hurriyet.com.tr*:

```
# perl emails.pl hurriyet.com.tr
Received 83 Hits:
[bavci@hurriyet.com.tr]
[tturenc@hurriyet.com.tr]
[ecolasan@hurriyet.com.tr]
[yatakan@hurriyet.com.tr]
[dhizlan@hurriyet.com.tr]
```

[fsever@hurriyet.com.tr]
[rcaglayangil@hurriyet.com.tr]
</snip>

Mail 83 people at Hurriyet – and chances are very good that someone will download and execute their shiny new screensaver.

Other obvious choices for obtaining email addresses for a specific company is to actually respond to any of those Spam emails that promise to provide you with 200 trillion email addresses neatly compressed on a CD ROM – you'll be on their mailing list forever.

5 Footprinting – per country

Let us for now assume that the internal worm (as described in previous sections) is already developed. We also assume that we have a module that will extract email addresses per company from the Internet and will send email to everyone that it "farmed". In order for us to target a complete country all we need now is to find the domains for various companies within the country.

We start by looking at industries that will be hurt most by an IT blackout, or where the country is largely dependant on the existence of the company or department. The following sectors come to mind:

- Telecommunication companies (fixed line, GSM, satellite)
- Energy providers (electrical [hydro, nuclear, fossil fuel], oil etc.)
- Government departments (for obvious reasons)
- Military (just because its more fun when it's a mil)
- Media providers (newspapers, TV, magazines, online content providers)
- Financial services (banking, insurance, stock exchanges and reserve banks)
- Prominent businesses (where, for example, 60% of the country's GDP is linked to a single company)
- Medical services (hospitals, clinics, medical research companies)

5.1 Private sector

Trying to dynamically create a list of domains for each country (per sector) failed in every instance. Many sites try to provide "global" directories. For example Google does it with their Google directory, there is the Open Directory Project (www.dmoz.org) and a couple of others. These promise the ability to automatically determine all the companies/organisations per country and per sector. The problem with these directories is that they are very "noisy". An entry in DMOZ for "science and development" in South Africa lists CSIR (a research giant in South Africa), but also the Treehaven Waterfowl Trust. Adding the Treehaven Waterfowl Trust to a list of possible targets (for Cyber Warfare) really doesn't get us anywhere, other than a very confused administrator who's probably more interested in the breeding habits of waterfowl than IT security. Most directories are also specific to the US.

How do we find domains for each of these sectors per specific country? In most of the cases we easily find a specific directory of companies for each sector in each country. Look at http://www.cellular-news.com/coverage/all_networks.php for instance (full list of links will be provided in an Appendix). With enough cutting, pasting, *awk, sed* and *perl* we can easily massage the domains into a database. Some directories are more difficult to mine – in these cases we can query the directory in real time. Querying the directories in real time has the advantage that the data is always up to date. It however has the disadvantage that we are now dependant on the directory being up and running.

Getting prominent businesses per country is not difficult (Business Day provides a nice list of the top 1000 companies worldwide). Getting the domains associated with the company is a bigger problem. By combining lists of companies we can easily end up with 2000+ companies across the globe. Manually finding their associated domains is certainly an option, but if you are lazy like me you would probably try to do it programmatically. American companies usually use companyname.com. How about other countries? Did the company manage to register the .com?

So here is the challenge: as input the country and company name, and as output the domain. A possible way to achieve this as follows:

- Find the TLD associated with the country
- Remove strings like LTD, Holdings, PLC etc from the company name
- Do a Google search for <*company name*> *site:TLD* (will search for pages Containing the company name in the specific TLD). Hope that Google's PageRank will do a first round filter.
- Extract all the URLS
    - If the URL's domain name match the company name directly save it
        - Example: Telstra – Telstra.com.au
    - If the company name contains several words – try the abbreviation
        - Example: National Australian Bank – nab.com.au
    - If the abbreviation is located within the domain name
        - Example: Deutsche Telekom - dtag.de, Japan Airlines – jal.co.jp
    - Remove the spaces from the company name and see if we have a match
        - Example: Banco do brasil - bancodobrasil.com.br
    - If the company name contains more that one word – if any word match
        - Example: Agilent technologies - agilent.com
    - Create a string "window" of 3 characters – slide the window over the company name, and count how many matching instances were found on the URL's domain. Calculate the average for all the domains found, and only report on domains higher than the average (this is basically a bastardized correlation function). This technique is used to match most fuzzy matches between domain name and company.
        - Example: Cocacola enterprises - cokecce.com
    - Do the same – but with the abbreviation
        - Example:Kansai Electric Power -kepco.co.jp,Skandinaviska Enskilda Banken - sebank.se
    - If Google only returns one match – extract the domain from the single match.

There are probably many more tricks one can use the automatically determine the domain name from the company name. How do we find the domain name if the company is located in Israel for instance and have a .co.il domain, but their main site is located on the .com TLD? No problem – we have a handy script that will extract email addresses from the Internet – if the main domain is the .com chances are good that more people would be mailing from the .com TLD than the .co.il. Do a scrape on both and see who gets the most hits.

Using this technique we can mine the Internet for country:company name combinations and easily, quickly and rather accurately determine the domain name. The technique is obviously not foolproof – but it beats surfing 2000+ sites..:)

5.2 Government and military

Most government and military networks have subdomains. The idea would be to obtain as many of these subdomains, in order to target the subdomains separately. By looking of the output of our Google email scraper for gov.za (for example) we see the following (extract):

[domestics@uif.gov.za]

[alexio@dacst4.pwv.gov.za]
[janetd@gpg.gov.za]
[schuttet@dwaf.gov.za]
[draller@premier.kzntl.gov.za]
[karinh@gpg.gov.za]
[cwipser@capetown.gov.za]
[pgrobler@pawc.wcape.gov.za]
[makhathinim@durban.gov.za]
[celeste@statelib.pwv.gov.za]

It's easily spotted that we are dealing with separate subdomains – *uif, pwv, gpg, dwaf, kzntl* and *wcape* – all within the .gov.za domain. If there are more subdomains within the *pwv* domain we can start again – this time with *pwv.gov.za* as input. The process can be repeated until no more new subdomains can be found.

Not all governments use *.gov.TLD* as their sub-TLD. The following table shows the TLD, the country and the government sub-TLD for countries that do not use *.gov.TLD*:

TLD	Country	Government sub-TLD
.at	Austria	gv
.bj	Benin	gouv
.ca	Canada	gc
.ch	Switzerland	admin
.es	Spain	map
.fr	France	gouv
.ga	Gabon	gouv
.gt	Guatemala	gob
.jp	Japan	go
.kr	South Korea	go
.mc	Monaco	gouv
.mx	Mexico	gob
.ni	Nicaragua	gob
.nl	The Netherlands	overheid
.no	Norway	dep
.pa	Panama	gob
.pe	Peru	gob
.pr	Puerto Rico	gobierno
.ro	Romania	guv
.sn	Senegal	gouv
.sv	El Salvador	gob
.th	Thailand	go
.ua	Ukraine	go
.ug	Uganda	go
.uy	Uruguay	gub

Not all countries have their government networks nicely named within a sub-TLD – in such cases effective government network mapping cannot be done.

The same principle works on military networks – there are however fewer countries that make use of the *.mil.TLD* sub-TLD. In many cases the military DNS space is located within a government DNS space – Singapore for instance is using the *mindef.gov* domain for their Ministry of Defence. Hereby a table of known non-standard military sub-TLDs:

TLD	Country	Military

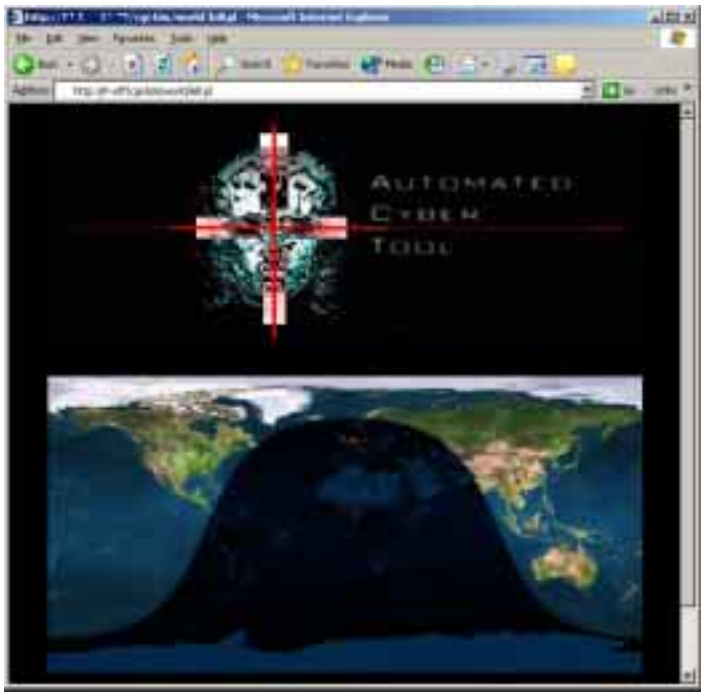
		sub-TLD
.at	Austria	bmlv.gv
.au	Australia	defense.gov
.be	Belgium	fmil
.bg	Bulgaria	md.government
.ca	Canada	dnd
.ch	Switzerland	armee
.co	Colombia	mindefensa.gov
.cz	Czech Republic	army
.de	Germany	bundeswehr
.dk	Denmark	sok
.eg	Egypt	mmc.gov
.es	Spain	mde
.fr	France	defense.gouv
.gr	Greece	mod
.hr	Croatia	morh.tel
.hu	Hungary	h-m
.il	Israel	idf
.it	Italy	difesa
.jp	Japan	jda.go
.kr	SouthKorea	mnd.go
.lb	Lebanon	lebarmy.gov
.lt	Lithuania	kam
.lv	Latvia	mod.gov
.mn	Mongolia	pmis.gov
.nl	The Netherlands	mindef
.pt	Portugal	mdn.gov
.ro	Romania	mapn
.ru	Russia	rian
.sg	Singapore	mindef.gov
.si	Slovenia	mo-rs
.sk	Slovakia	defense.gov
.th	Thailand	mi
.uk	Britain	mod
.uy	Uruguay	armada.gub

The current real time mapping of military DNS space is much less effective than the government DNS space.

6 Putting it together

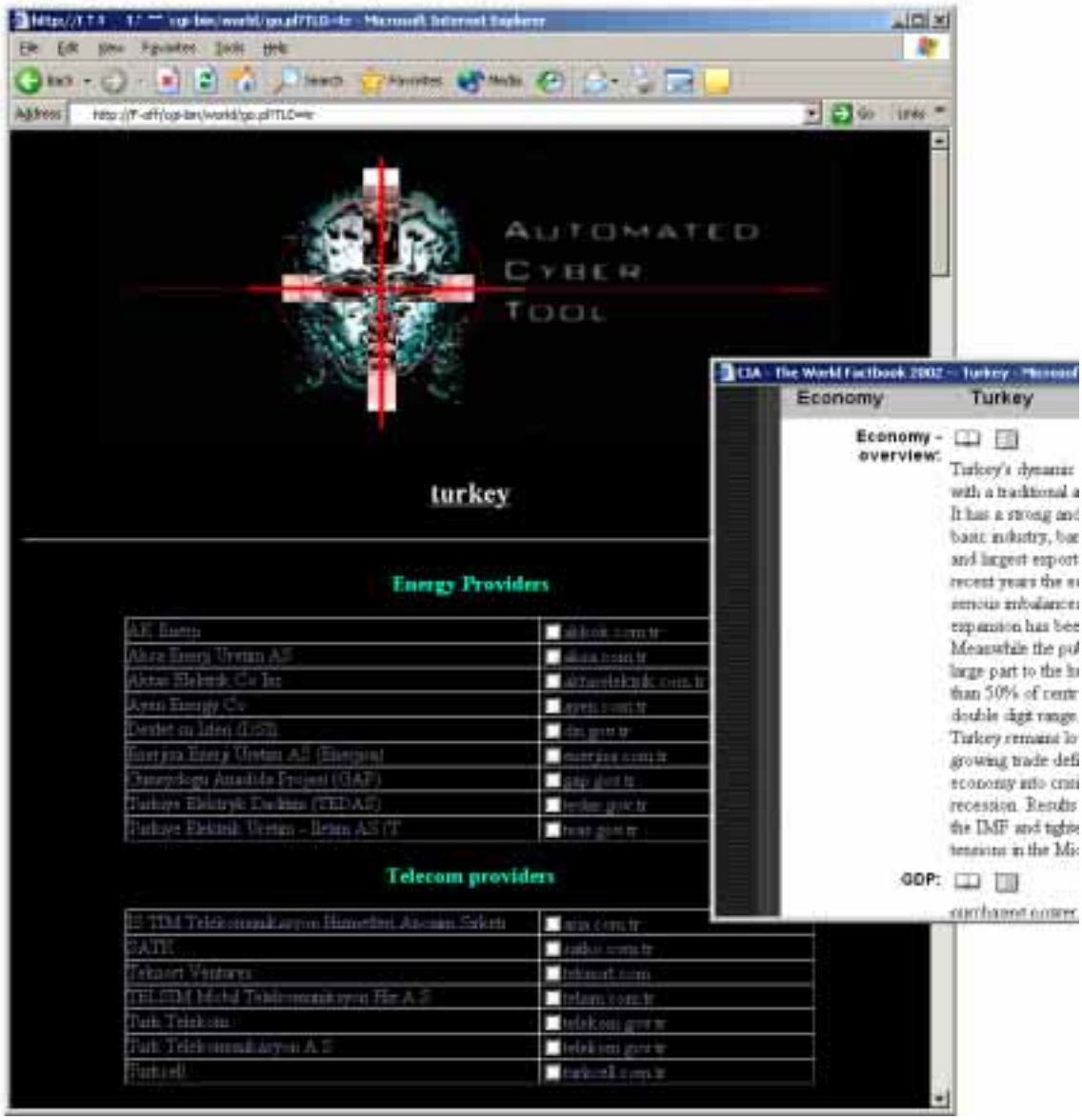
What good would all these modules be if we can't tie them together in a GUI that looks like a scene from the movie "Hackers"? So on Keith from BlackHat's request we built a GUI (hi Keith – you rock!). The following are screen shots from the GUI – the GUI is so easy that very little explaining is needed.

The first screen is a real time sun map – the idea here is that you would want to know where it is day time (normal people don't read their mail at night):



After selecting the continent where the target country is, a clickable map of the continent is displayed:

Clicking on a country leads to a screen that shows the domains of companies/organisations within each sector. We click on…Turkey (no hard feelings guys). Clicking on the country's name pops up a CIA world fact book entry for Turkey – with the "Economy" part in focus.

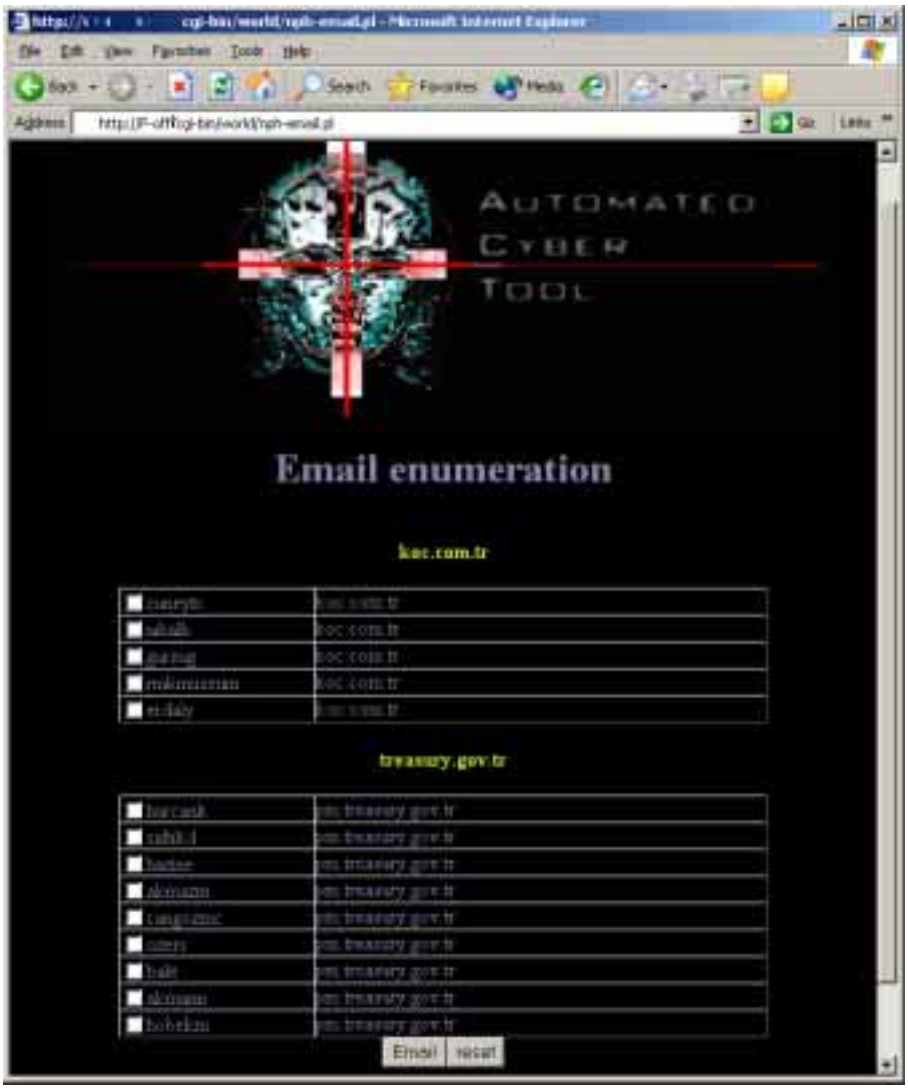


ACT lists:

- Energy providers (from database)
- Telecom providers (from database)
- Newspapers (real time from directory)
- Prominent businesses (from home grown database – see private sector)
- Government (real time)
- Military (real time)
- Financial services (real time from directory)

Any number of domains can be selected (by checking the tickbox). By default none are selected – but it is envisaged that, when the system is operational, the default will be to have all selected. The user proceeds to click on the Email button at the end of the form. ACT will now perform a

Google scrape on all the domains that were selected and display the results. For the purpose of this document only two domains were selected – one from the “Prominent Businesses” and one from Government. The following emails were scraped:



Again, by default the email addresses are not selected, but in a working copy the default will be to select all. We choose two addresses and click on “Email”. The system proceeds to spoof email to the person as per the “Delivery” phase of this document:

7 Breaking in from outside

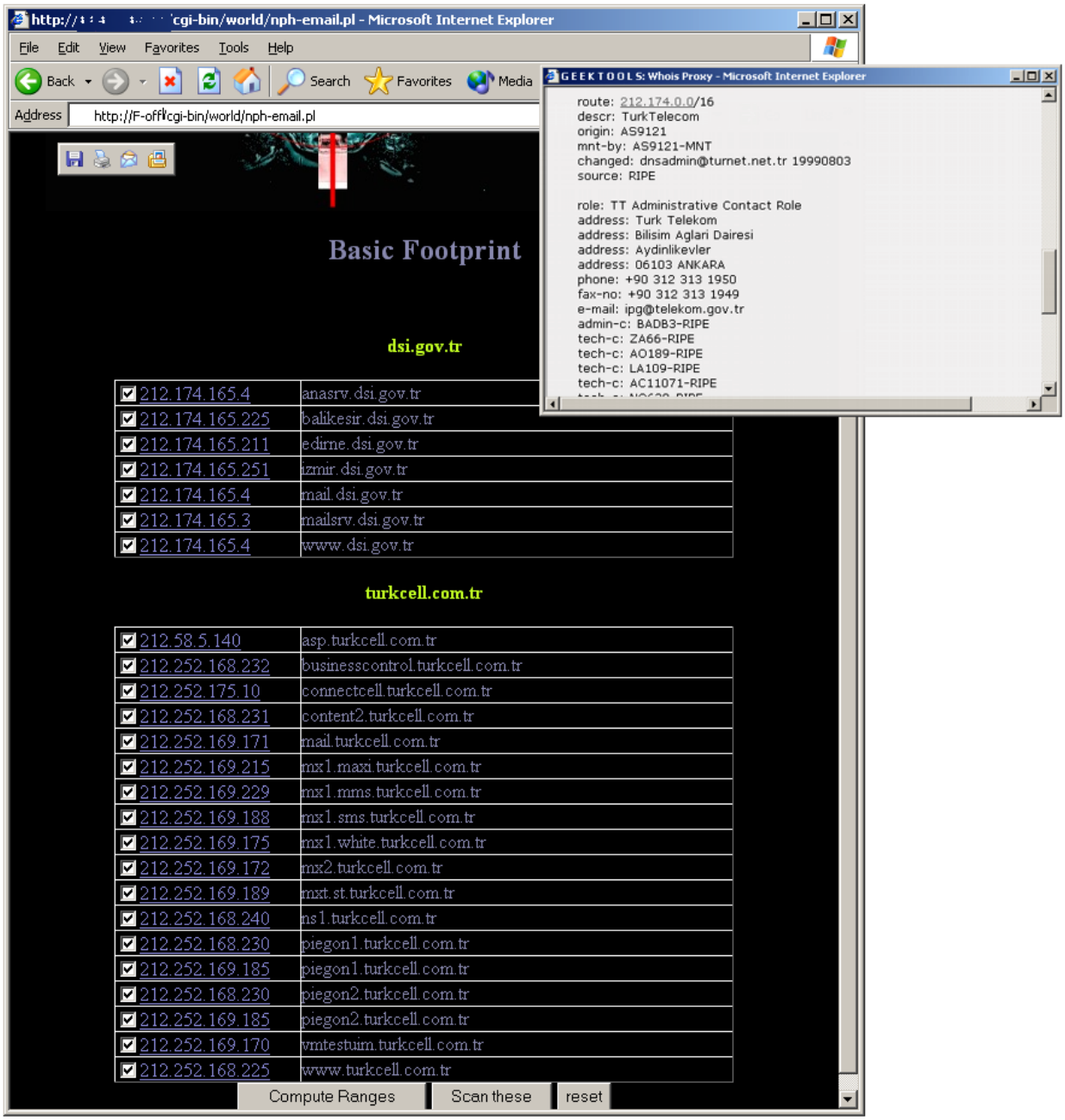
People normally associate cyber terrorism with hackers that break into computer networks and shut down the electrical system (for instance). Experience, and many studies, tell us that this is either not feasible as control systems are kept apart from IT networks, or that such a single incident will not create the amount of impact an attacker would want to accomplish as there are hundreds of energy providers in the US for example. A better idea, or not as bad, would be to look wide (country-wide) and not deep. Search for vulnerabilities that can quickly be exploited reliably and without effort.

After having collected all the domains for the different sectors of each country we decided, as a kind of afterthought, to use the tools that we have already developed (see BlackHat 2002 Singapore and BlackHat 2003 Windows Seattle talks and papers) to quickly do very basic footprinting on each domain that we found. We also added the ability to expand the IP numbers to ranges (via “reverse” traceroute), to do a mini portscan, extract banners, and build a “low hanging fruit” vulnerability finder (it’s NOT a scanner – it’s badly hacked PERL).

The idea was to create something that will quickly look at the IP space of many domains and determine if any of the perimeters can be breeched with ease. As most networks these days only allows HTTP and SMTP across the perimeter, the LHF (low hanging fruit) finder will test only for common problems found on Microsoft’s IIS. The tester was built to determine if the vulnerability really exists – that is – not to determine if, for instance, MSADCS.DDL exists; but if it can be exploited.

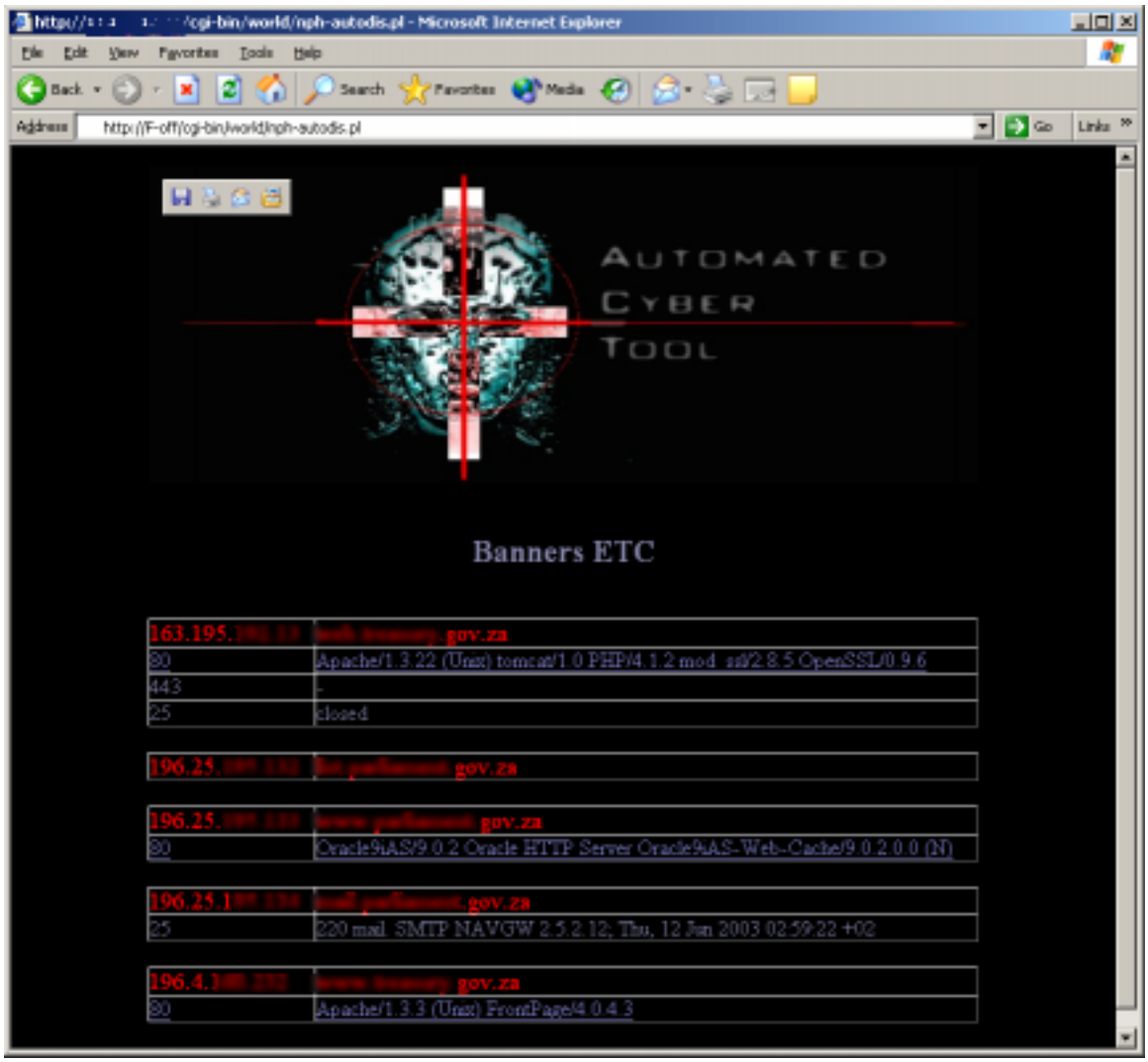
The following screenshots shows how this is done in ACT:

We stick with Turkey – and do a basic footprint on just two domains – that of GSM provider *Turkcell* and government department *DSI*. Clicking on an IP opens a small window that points to the GeekTools whois proxy – so that we can verify the range and target:



Interestingly we find from GeekTools whois proxy that government department DSI is located in a class B assigned to Turkish Telecom.

Not wanting to piss off the Turkish government, we decided to rather move a little closer to home by looking at some of the local South African government's departments. We proceed to do a mini port scan on the IP numbers selected by clicking on "Scan these".



Not wanting to get into too much trouble with the ZA Gov, we stop at this stage. Note the Apache version and OpenSSL version on the first host. Notice that the IPs and DNS names have been blurred out..:)

Links from HTTP ports on this screen would lead directly to the LHF finder – which will not be shown here.

As far as the rest is concerned, you hopefully get the picture…

8 Conclusion

First off we would like to apologise to any parties we may have offended by using examples in which they are the “supposed” target. We attempted to approach this as cautiously and respectfully as possible, while still providing the reader with a real-world feel.

Having paid the ferryman, we hope we put the Tea back into Cyber Terrorism. No sugar and a spot of milk please!!