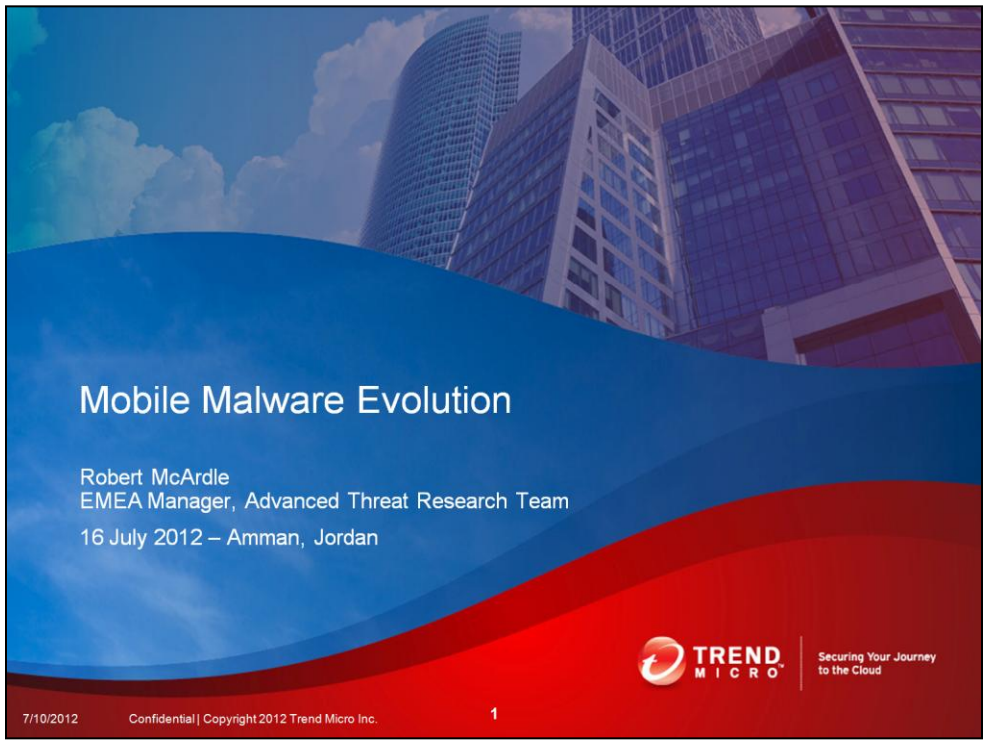
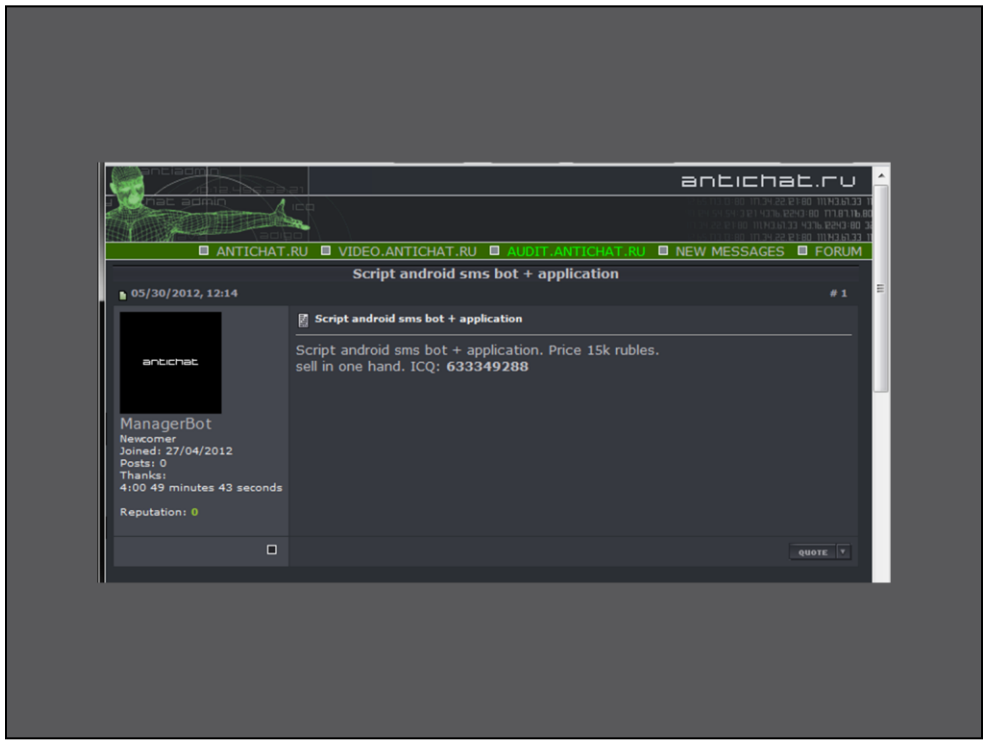


[BRIEF INTRO OF MYSELF AND TEAM]

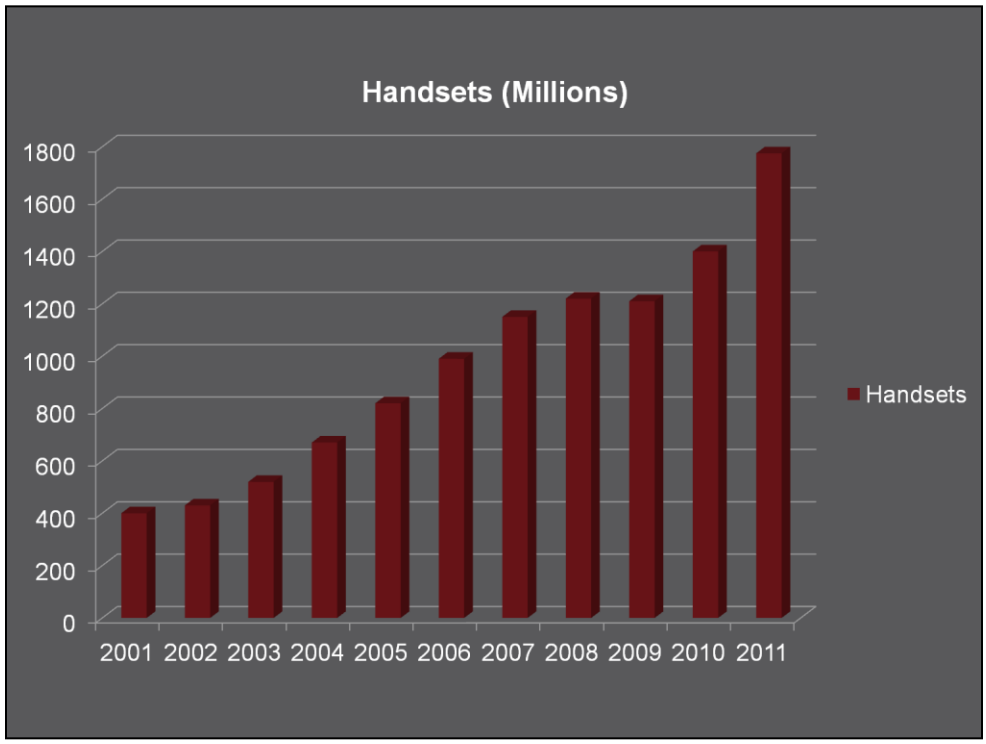
[BIT OF PHONE GEEK – THIS IS MY 20TH PHONE TO DATE, AND SINCE 2006 THEY HAVE ALL BEEN SMARTPHONES]

Let me start by saying that mobile malware is no longer a theoretical issue…



… it is now a mainstream form of attack, and is actively being discussed and traded on Cybercrime forums. Here is a translated version of a post from the forum AntiChat.ru – where it is being sold for a little over €350.

Traditional Malware itself is 26 years old this year, and of course mobile malware is much younger. It is however evolving much faster than its windows cousins, largely because it is being driven by professional cybercriminals instead of people doing this as a hobby. This rise in mobile malware is also being fueled by the rise in sales of mobile devices – especially smartphones.



Here you can see a chart of mobile phone sales over the last 12 years. Last year a staggering 1.7 Billion handsets shipped worldwide – that's enough handsets for a quarter of the worlds population. The majority of them are not actually smartphones. Even still the amount of smartphones sold is huge. Between now and our coffee break at 10:40 there will have been over 50,000 Android phones activated.

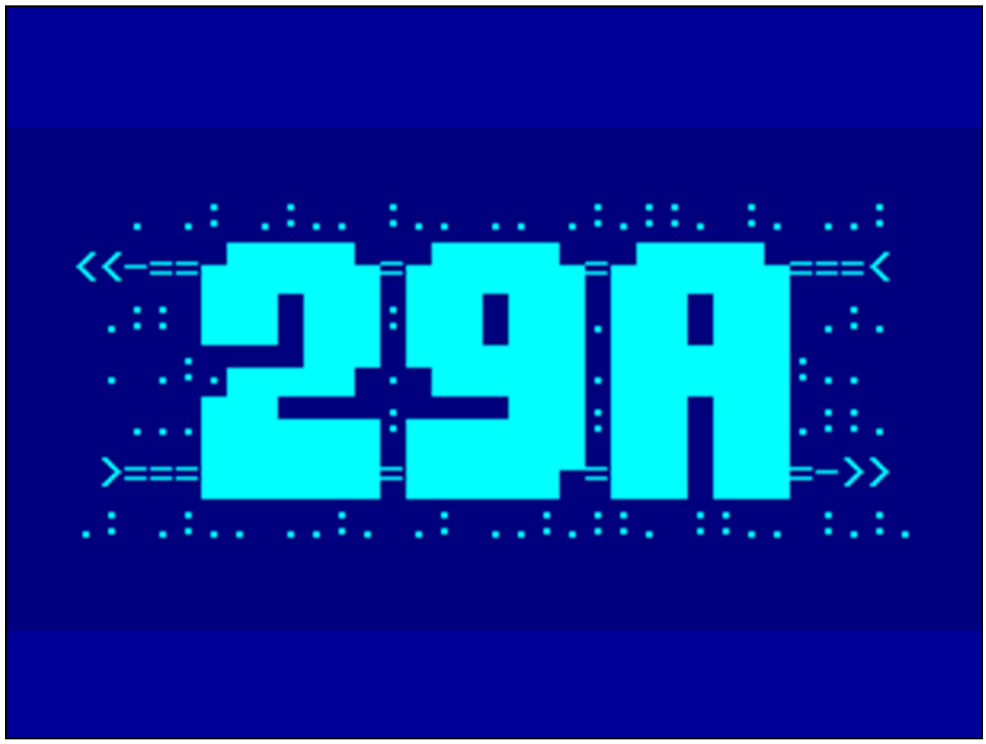
And what about Apple – they made 13 billion profit in the last quarter of 2012. To put that in perspective that makes them 3 times the size of Microsoft.

So its pretty clear the world is changing – but to understand how modern mobile malware works, we need to go back to the start..

Lets look at 2004, which is scarily 8 years ago at this point..

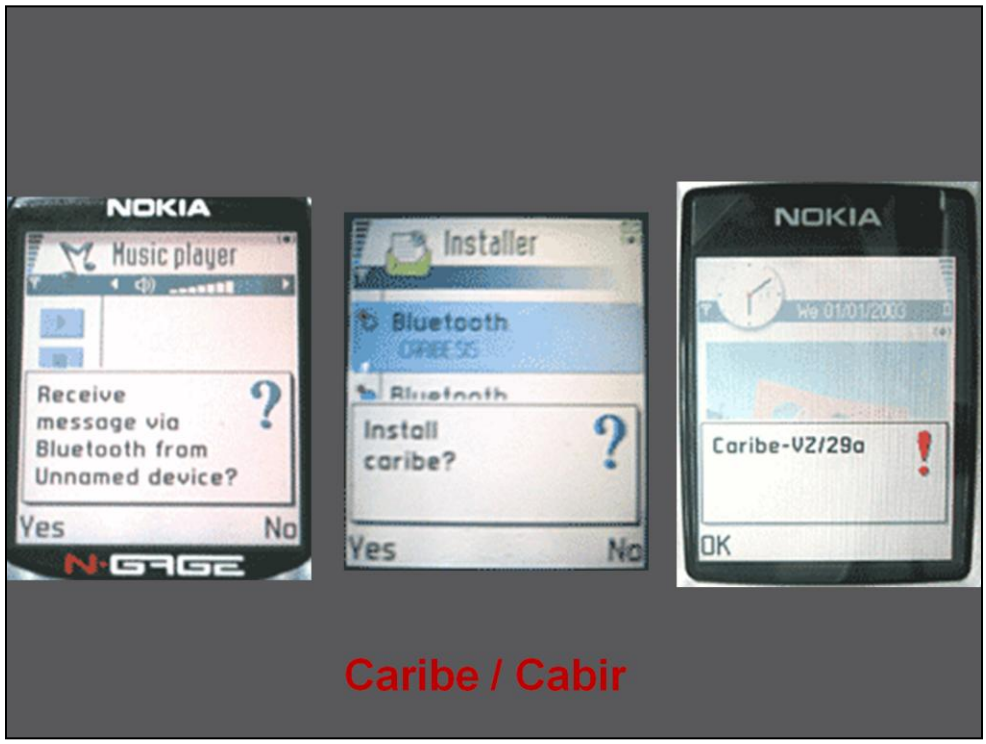
Back then this is what a “Smartphone” looked like, and Nokia’s Symbian operating system was completely dominant.

IMG: Nokia 6620 running S60



In the same year the famous malware writing group 29A became interested in the idea of mobile malware. This well known group where also responsible for some of the most innovative malware ever released including the first 64bit malware for Windows Vista, which they created even before the OS had been released by Microsoft. The groups aim was to create Proof of Concept malware for non-standard OS.

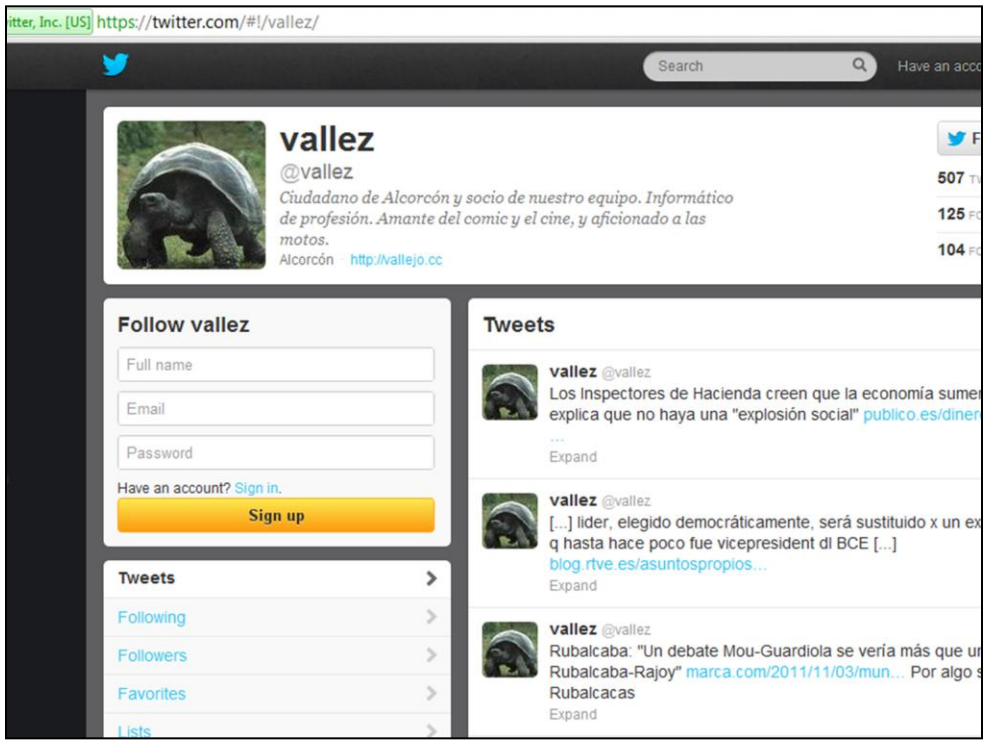
Incidentally does anyone know where the name “29A” comes from? It is hexadecimal for 666 – which is associated with the antichrist and the devil.



On June 14th 2004, one member of 29A known as VirusBuster sent a file called Caribe.sis to all of the major AV vendors. On first inspection this file was unusual compared to the normal Windows samples vendors where used to receiving. It has clearly been created for the Symbian OS. Also most reverse engineers were familiar with reversing intel based executables, using x86 assembly – whereas Symbian uses ARM.

It did not take long however for researchers to discover that this was the first mobile malware. Once installed it would display a message on the screen, and set itself to autostart with the phone. It would then look for all available Bluetooth devices within range, and send a copy of itself to them.

By the way – every AV vendor have slightly different names for the malware I'll discuss today, so I'll use the most common names instead of Trend Micro specific ones.



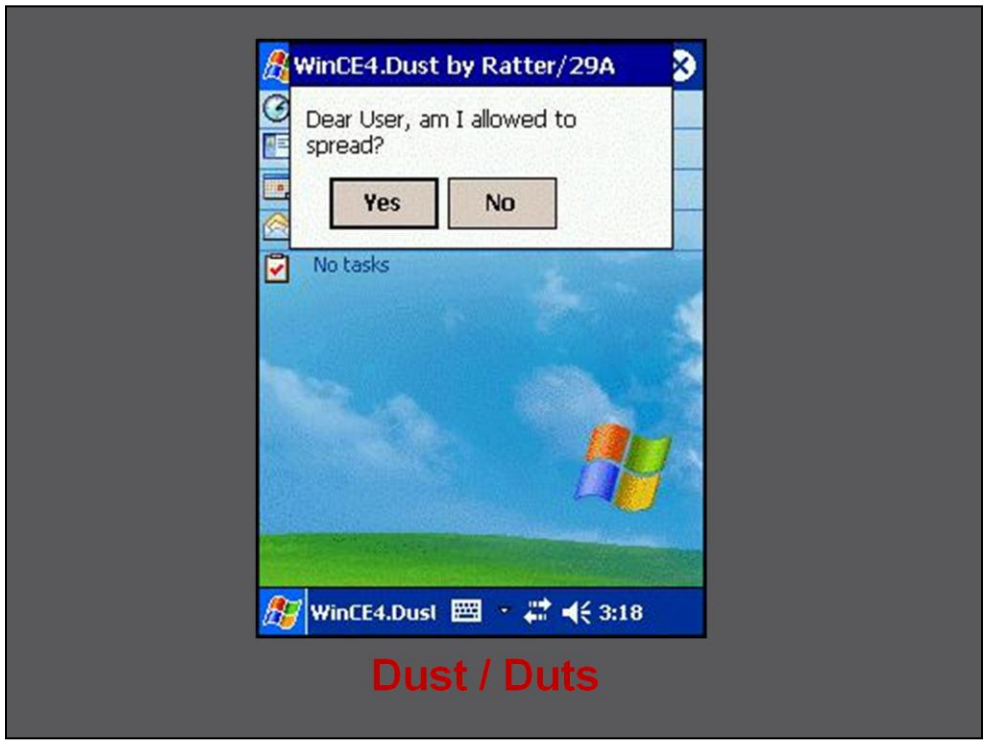
Caribe was developed by a Spanish malware writer who uses the handle Vallez, but whose real name is Javier Vicente Vallejo. He claimed that the time was right for mobile malware for 3 main reasons

-There was a single platform that was popular enough to target (Symbian). This is similar to the what we saw in the early days of Windows malware

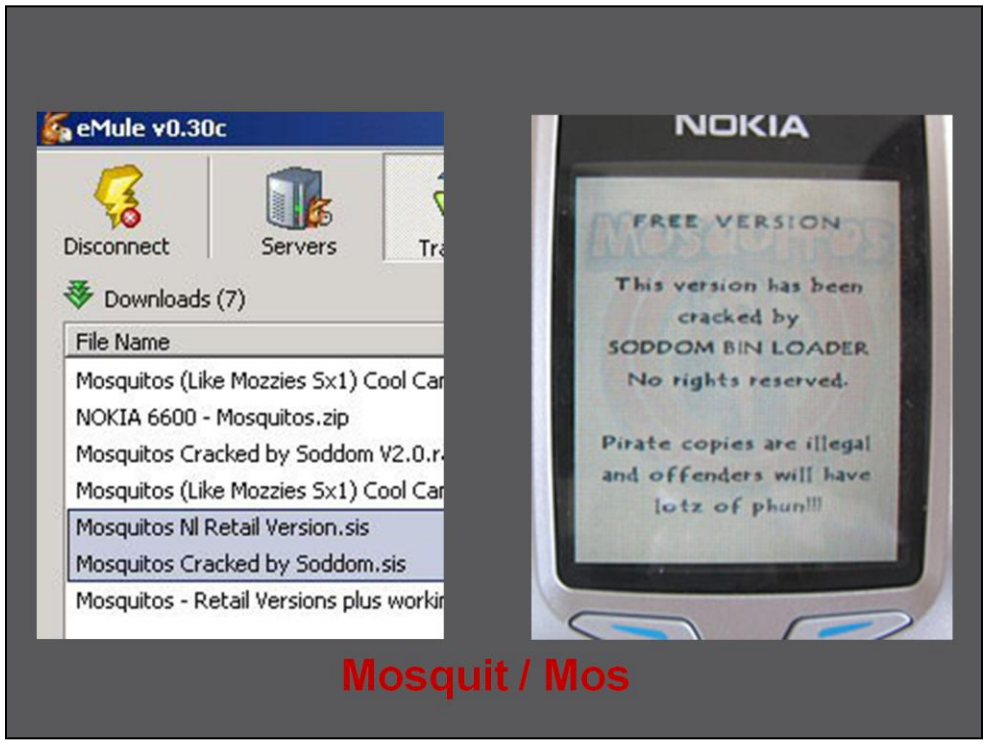
- Development tools were readily available. Symbian and Nokia had some very good SDKs

-There are a number of security issues in the platform

This malware got a lot of attention, both among the security industry and among hackers – who eagerly waited for the 29A group to release their next online magazine which would include the full source code. This magazine was released later that year.

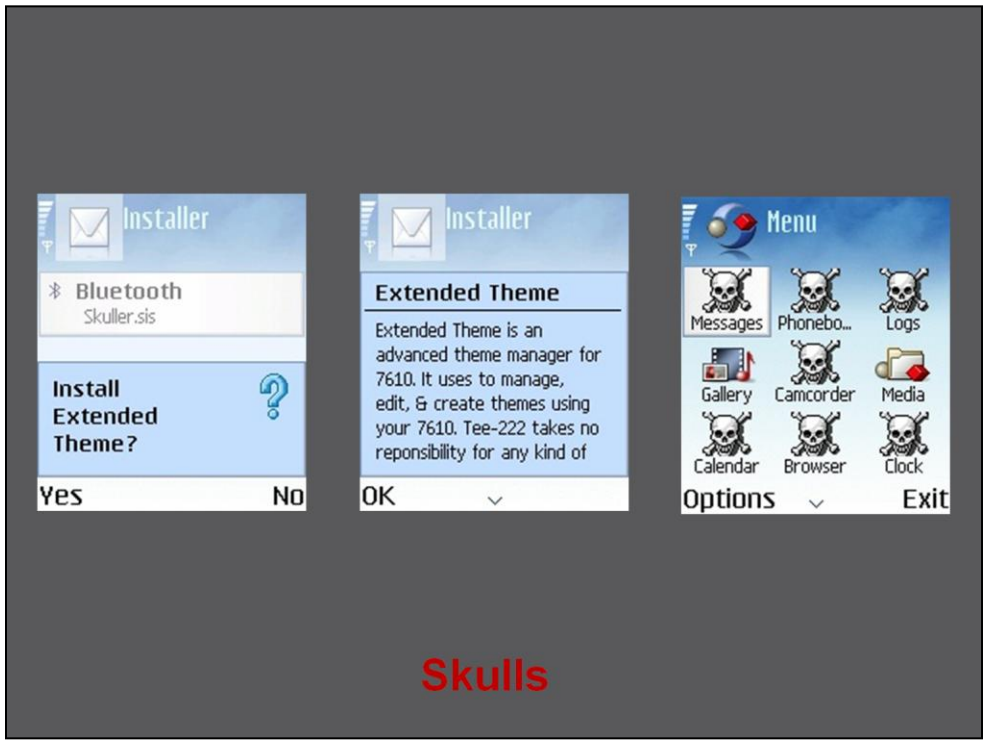


In the mean time however, mobile malware continued to evolve. Only a month later, in July 2004, another member of 29A released the first mobile malware for PocketPCs. Unlike Caribe which was a worm, Dust was a File infector – spreading to all exes in the same folder which were larger than 4K. Ultimately however PocketPC malware did not catch on – much like the Pocket PC itself.



Worryingly it did not take long for the first financially motivated malware to arrive on mobile platforms, in fact only 2 months after Caribes release, in August 2004 – a new Trojan called Mosquito arrived. This malware was a cracked version of a popular game called Mosquitos, but it contained a trojan inside. The attacker spread his cracked version, using P2P networks such as eMule.

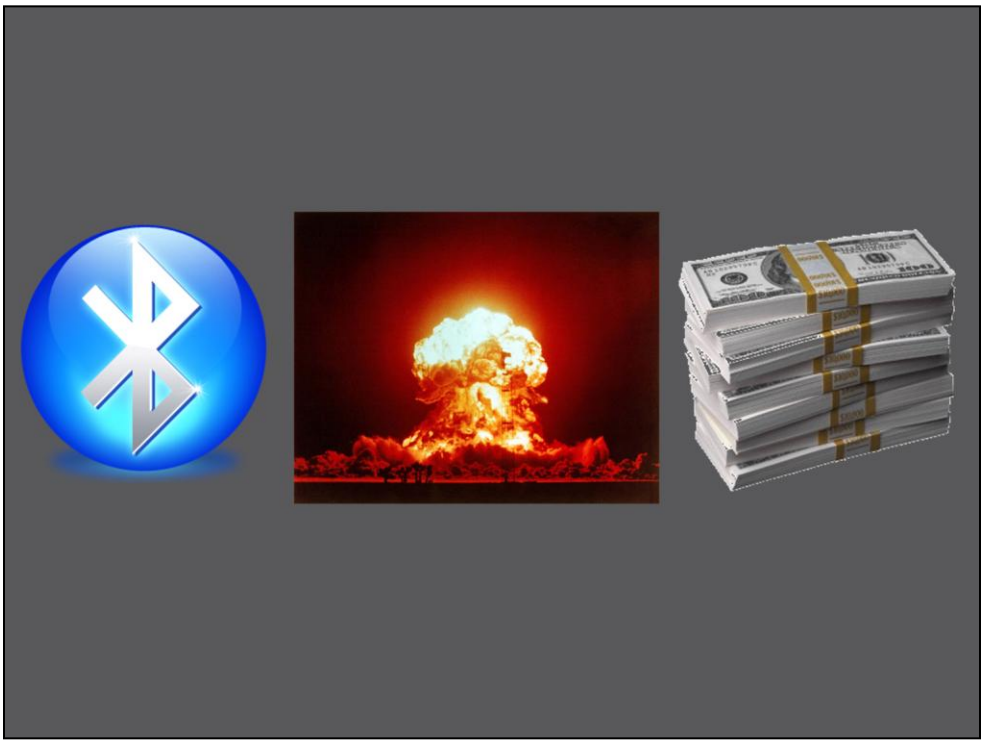
When executed, the Trojan displayed a message to the user – before running the cracked game. However it also sent an SMS message to a premium rated service without the user knowing – in turn earning money for the malware author. This exact same technique is still in very regular use today.



Towards the end of 2004 a new Trojan, known as Skulls, became the first of a what would become the largest family of mobile malware for the next 1-2 years. This was the first malware to take advantage of a security flaw in Symbian that allowed any application to overwrite system files with their own files, without prompting the user. This was actually a feature of Symbian, as opposed to a flaw – as it allows system files to easily be updated.

Skuller however used this technique to effectively delete all applications on the phone. It also replaced all of the application icons to a scull and crossbones, which is where this malware gets its name. To make matters worse, if the victim switched off the handset, it would stop working when it was turned back on again. This type of purely malicious Trojan became very popular among early mobile malware writers.

So by the beginning of 2005 there were three main approaches to mobile malware:



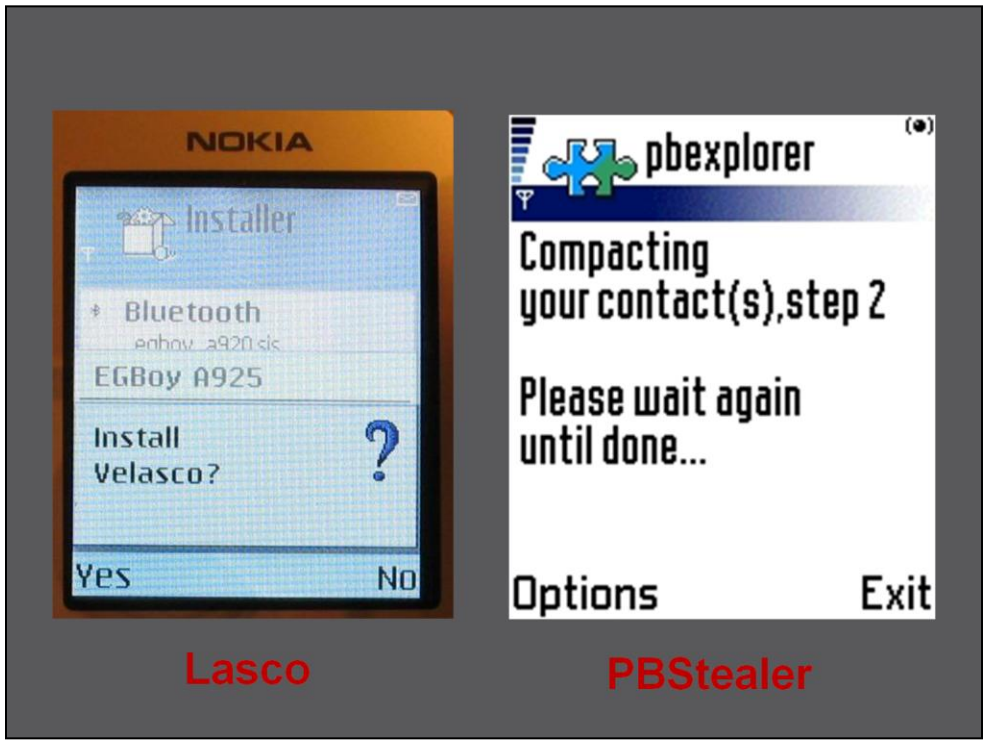
Firstly there were worms that spread via Bluetooth, such as Caribe

Secondly we had purely malicious, destructive Trojans – such as Skulls and its variants

And lastly we had Financially motivated Trojans, such as Mosquito

```
#include "general.h"

#include "CaribeInstaller.h"
#include <aknapp.h>
#include <e32std.h>
#include <e32base.h>
#include <e32def.h>
#include <f32file.h>
#include <bautils.h>
#include <eikenv.h>

#include "file.h"
//#include "sisheader.h"

#define AUTOSTARTABLE "C:\\SYSTEM\\SYMBIANSECUREDATA\\CARIBESECURITYMANAGER\\CARIBE.APP"
_LIT(Autostartablestr,"C:\\SYSTEM\\SYMBIANSECUREDATA\\CARIBESECURITYMANAGER\\CARIBE.APP");
#define AUTOSTARTABLERSC "C:\\SYSTEM\\SYMBIANSECUREDATA\\CARIBESECURITYMANAGER\\CARIBE.RSC"
_LIT(Autostartablerscstr,"C:\\SYSTEM\\SYMBIANSECUREDATA\\CARIBESECURITYMANAGER\\CARIBE.RSC");
#define AUTOSTARTABLEPATH "C:\\SYSTEM\\SYMBIANSECUREDATA\\CARIBESECURITYMANAGER\\"
_LIT(Autostartablepathstr,"C:\\SYSTEM\\SYMBIANSECUREDATA\\CARIBESECURITYMANAGER\\");
#define RECOGFILE "C:\\SYSTEM\\RECOGS\\FLO.MDL"
_LIT(Recogfilestr,"C:\\SYSTEM\\RECOGS\\FLO.MDL");
#define RECOGFILEPATH "C:\\SYSTEM\\RECOGS\\"
_LIT(Recogfilepathstr,"C:\\SYSTEM\\RECOGS\\");
#define SISFILE "C:\\SYSTEM\\SYMBIANSECUREDATA\\CARIBESECURITYMANAGER\\CARIBE.SIS"
_LIT(Sisfilestr,"C:\\SYSTEM\\SYMBIANSECUREDATA\\CARIBESECURITYMANAGER\\CARIBE.SIS");

unsigned short DOCRC16(unsigned short crc,void * array,int size);

CaribeInstaller::CaribeInstaller()
{

}

CaribeInstaller::~CaribeInstaller()
{


}

/*****************************************************************
```

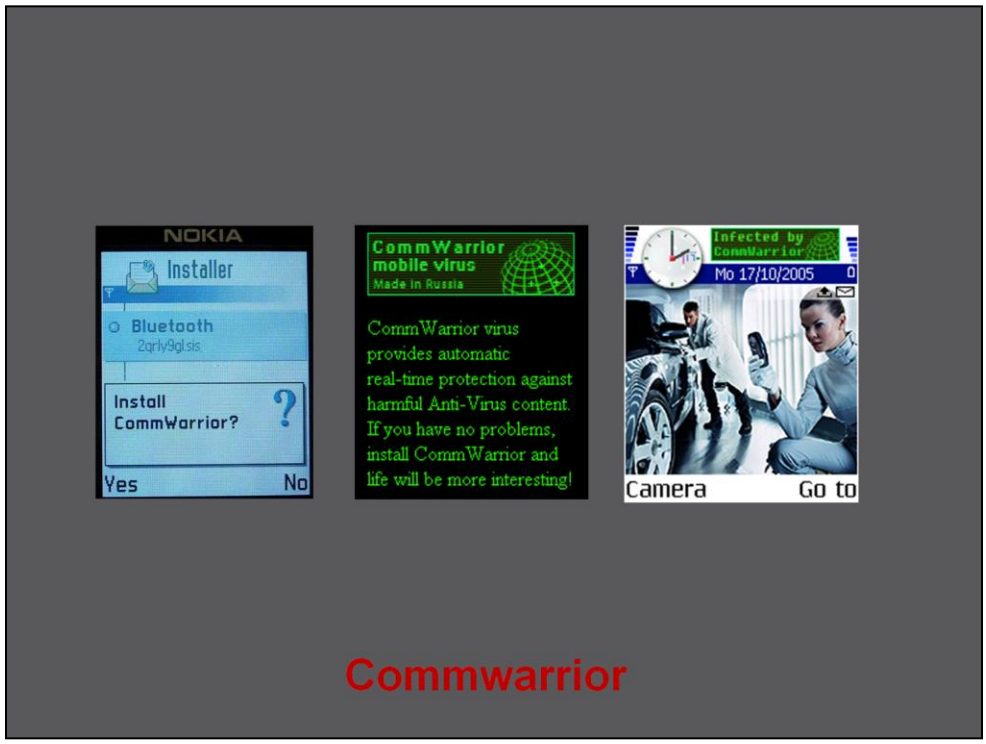
At the end of 2004 however, 29A released the source code to the Caribe worm. This in turn led to a lot of spinoff projects, and a similar situation occurred with the Skulls malware.



Lasco, like Caribe, could spread over Bluetooth. It was originally a series of simple modifications to the Caribe source code made by a Brazilian Hacker called Marcos Valasco. He submitted these modified versions of the malware to AV companies, who repeatedly detected them as Caribe variants – as there were no major differences. Needless to say the virus writer was not very happy about this, so he decided to add new functionality to the worms. The Lasco malware consisted of a modified Caribe that also worked as a file infector for SIS files – the install files for Symbian applications. This was a pretty significant step forward in mobile malware, however for some reason this technique thankfully never really caught on.

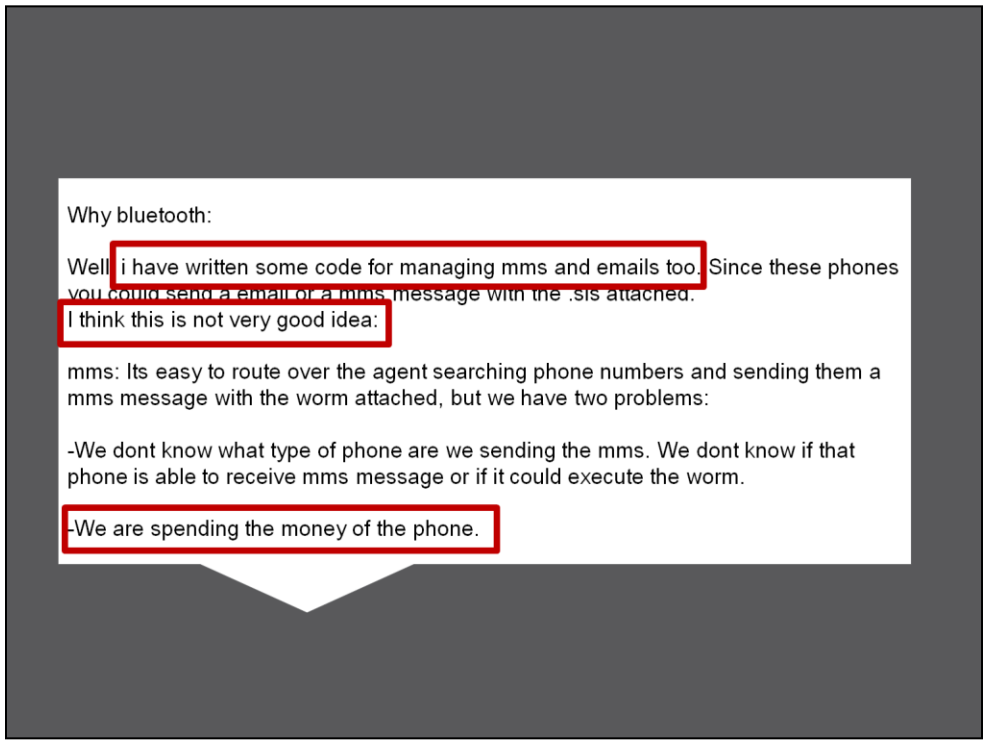
Another modified version of Caribe was used in the malware known as PbStealer, which was the first mobile Spyware and which originated in Asia. PBStealer pretended to be compact the users address book. However PBStealer worked by sending the entire contents of the phone phonebook to the first device within Bluetooth range. Of course this meant that the attacker needed to be around 10 foot away from the victim, but it was an interesting development none the less.

On top of these Caribe was widely becoming used as a means to spread other malware, such as Skulls.



The second major breakthrough in mobile malware came in March 2005 when a hacker known as e10dor unleashed a new creation known as Commwarrior.

Like caribe this was a worm, and could also spread over Bluetooth. However Commwarrior also had the ability to spread over MMS message – greatly increasing its potential victims. Bluetooth worms are restricted to only infecting victims within a short distance of each other, MMS can spread from country to country in a single hop.



Interestingly Vallez the author of Caribe had already thought about using MMS messages, as can be seen in the Caribe release notes – but choose not to as this would incur a cost on the victim. The author of Commwarrior had no such objections

Also commwarrior was very smart about the way that it choose to spread. During day time hours it would concentrate on using Bluetooth to spread, when the victim was most likely near other phones. At night time it would then send MMS messages to all of the users contacts.

Ultimately there were not many variants of the Commwarrior malware, as the source code was not released for at least a year, and even then was hard to find. It was however quite active in the wild, with infections reported in at least 20 countries.

In addition to worms, 2005 also saw the arrival of a bunch of new destructive trojans. Many of these were simply variants of the technique used by Skulls, such as Booton

Cardblock was a family of Trojans that would encrypt the phones memory card with a random password, effectively rendering it unusable

Blankfont did exactly whats it name describes, changing the default font to one with no letters – making it very tricky to use the phone

Another interesting malware was Cardtrap – this was a destructive trojan similar to Skulls, but which also placed Windows malware on the memory card – so that it would automatically run when inserted into a Windows PC – an example of a cross OS malware.

So what came next? Was there an explosion of malware?

Well strangely, no there wasn't

You might remember earlier we mentioned 3 things needed to properly support malware – a single platform popular enough to attack, development tools and security issues. During 2006 this first condition no longer existed. Over the period of these two years Symbian's stranglehold on the market was gradually eroded, until it ended up with less than half of the smartphone market. Windows Mobile 5 and 6 were first to cut into that market share, pulling out about 20% in total.

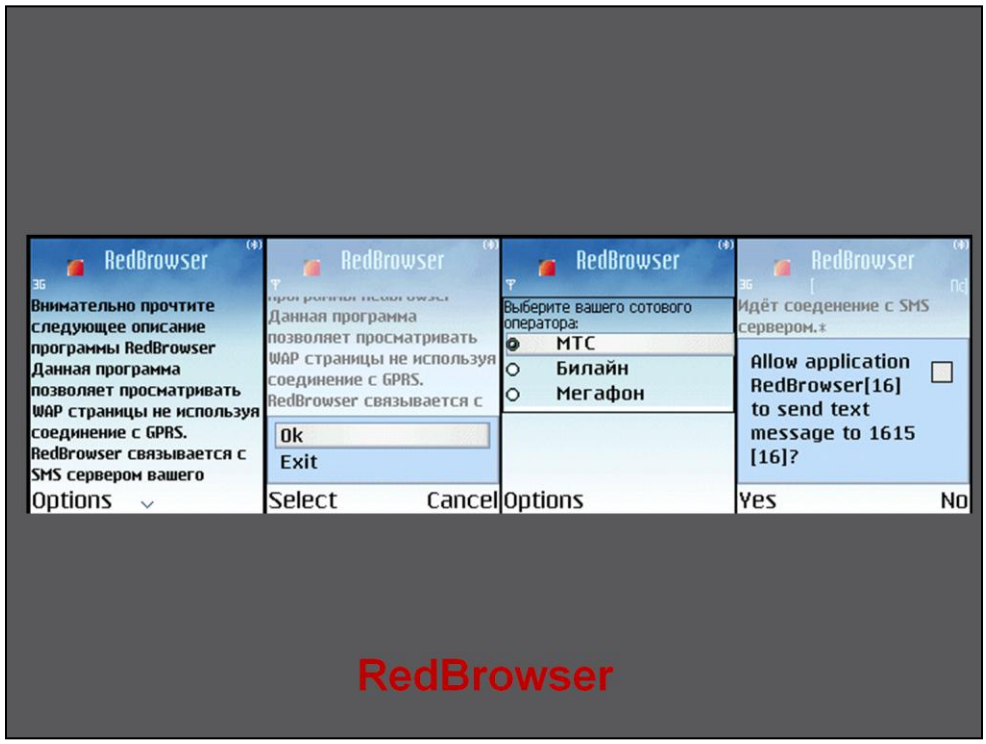
Next along came RIM, whose Blackberry platform was very popular in the US, and among Enterprise customers everywhere

And of course this happened…

The mobile market had once again become quite fragmented, so writing malware for one OS no longer made much sense.

Of course that is not to say that there was no development of mobile malware during this time – in fact the number of malware families pretty much trebled, but most of these were simple evolutions of the malware we have already talked about. There was one big difference however – criminals had found one attack vector that would work regardless of the OS involved…

Java, or J2ME (Java 2 Micro Edition) to be precise. All of these various platforms – Windows, Blackberry, Symbian – all of them can easily run Java. To add to that Java is quite easy to develop.



From early 2006 onwards, and right up to today lots of malware families arrived that all did one thing – send SMS messages to high cost numbers, and Java was the language of choice.

The Trojan that started all that was Redbrowser. This Trojan which both came from and targeted Russia, pretended to be a application that would allow the victim to surf the web over SMS. So if your phone did not have internet access, you could simply put www.google.com into Redbrowser and it claimed it would use cheap heavily compressed SMS to download and display the page. In reality of course it repeatedly sent SMS message to a very high cost premium number, costing the victim quite a bit of money.

Following on from that Trojan we see many many more malware variants over the next year or two using the exact same trick. Most of these trojans were propagated on P2P sites, mobile ringtone sites and so on – and were quite successful in some countries. Also in the vast majority of cases these malware were targeting Russian users – as they all used shortcodes that would work only in Russia.

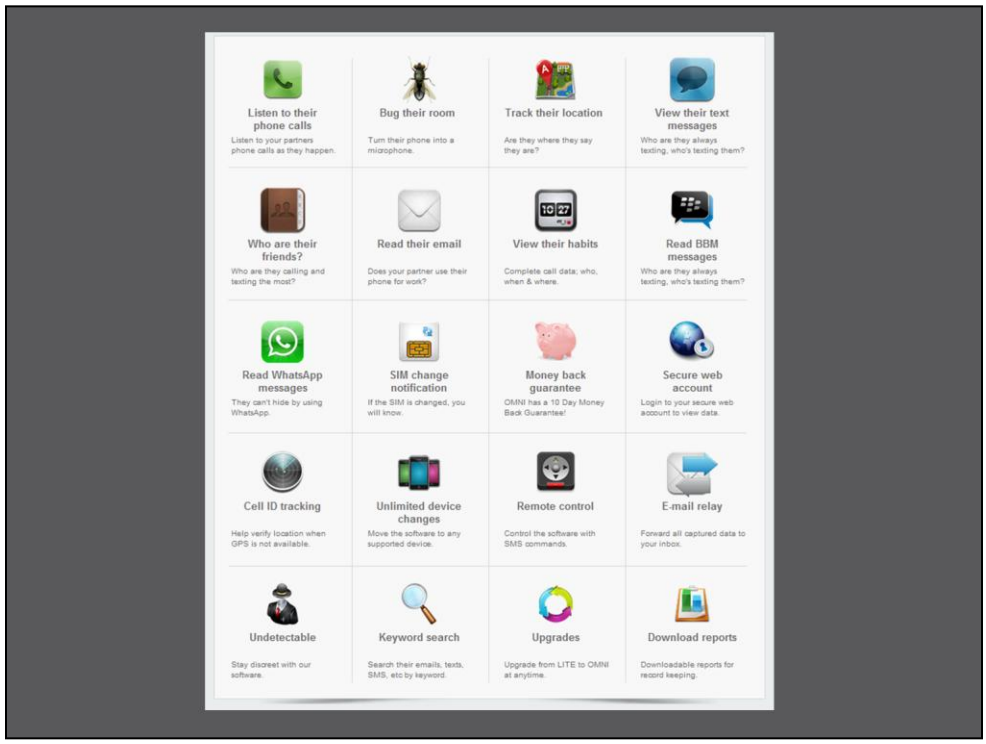
There was one notable exception to this – several variants of the Flocker family started to send SMS messages to the three digit number 151, all containing the same format – TP <12 numbers> <4 or 5 numbers>. Three digit numbers are not used in Russia

It turns out that an Indonesian mobile provider offer a service where by people with a specific type of SIM card could wire money from their account to another users account (who had the same type of SIM card), all using a SMS. Can you guess what the format of these messages had to be? Well they had to start with TP ☺

In this case it looks like someone had simply modified an earlier version of Flocker, which had historically targeted Russia victims.

There was one other very notable malware coming out of this period in time – Flexispy.

Flexispy is the first commercially available Spyware product for mobile phones – originally released for Symbian, it is now available for Android, Apple, Blackberry, Windows and Maemo as well.

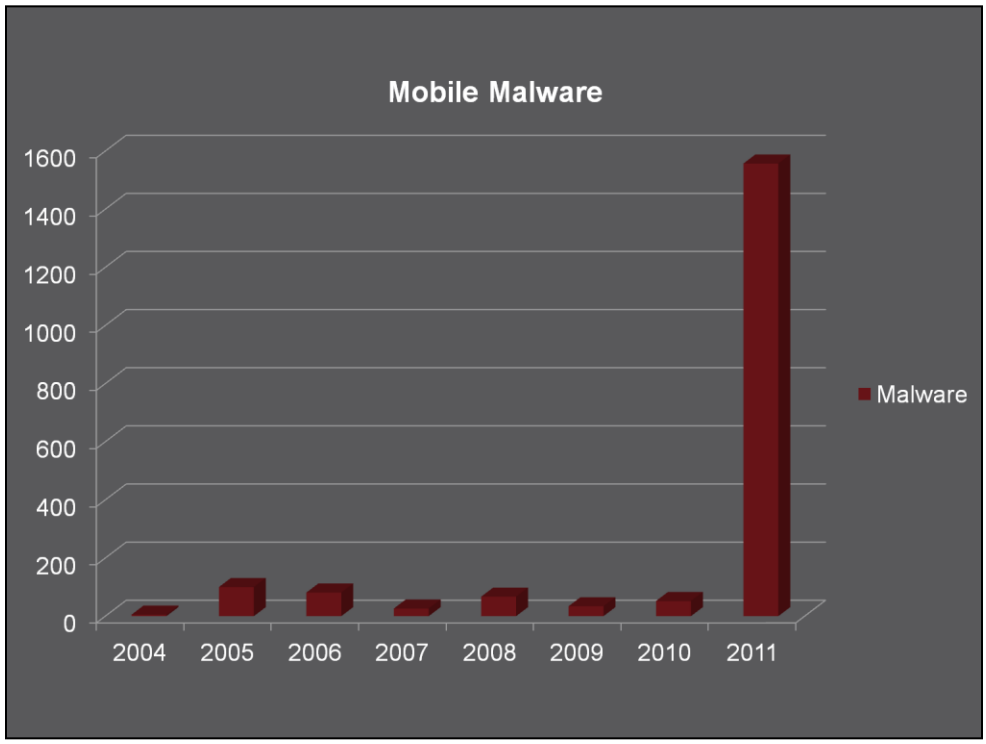


Flexispy offers a range of quite advanced features

- Reading SMS, Emails, IM, Address Book
- Geolocation tracking, use the phone as a bug
- Intercept phone calls, Complete call logs, who do they call most

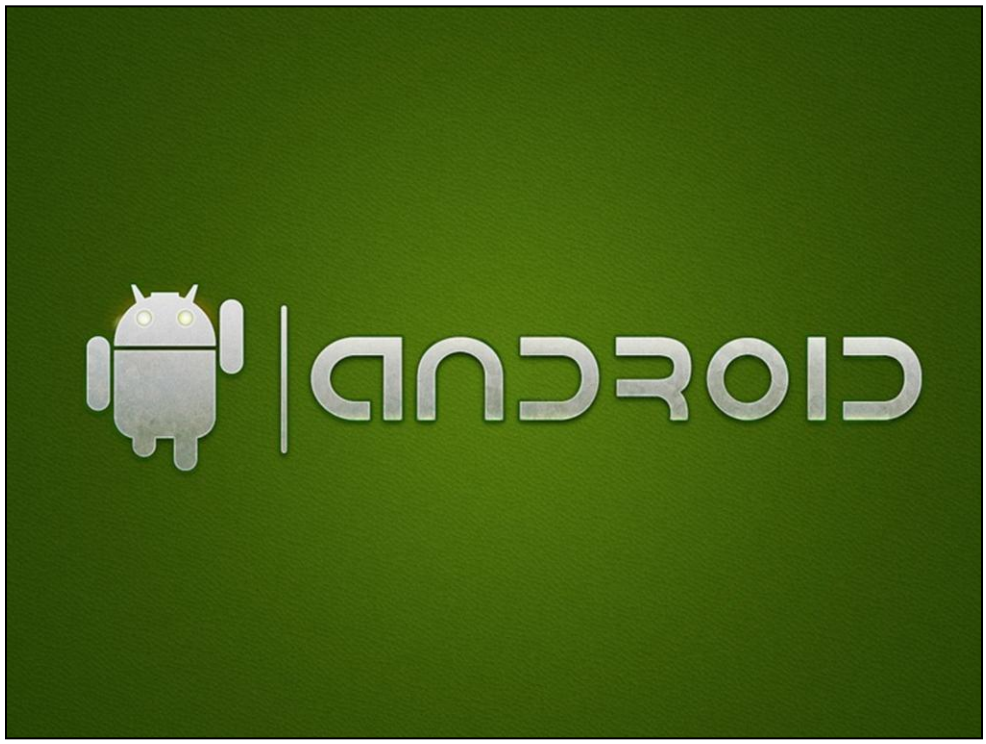
Marketed as an app to be used to monitor a cheating spouse, Flexispy is however far more devastating when used in a targeted attack. Picture this scenario – the attacker succeeds in installing Flexispy on the mobile of an executive from the target company. He checks the execs calendar and notices that he has an upcoming board meeting. Once the board meeting starts, he simply actives the microphone on the device, and has it call back to the attacker. Now the attacker can listen in on the entire meeting – and there are no visible signs on the handset itself to indicate that a corporate espionage is taking place.

All of that takes us up as far as 2009, and that's when everything begins to change



Over the next 3 years the volume of mobile malware is going to grow exponentially, and the real era of mobile malware will have arrived.

So what lead to this explosion?



Around the start of 2009 the first Android devices started to hit the market. The Android OS development is primarily overseen by Google – however several other big players also take an active involvement – including Samsung, Intel, HTC and Motorola.

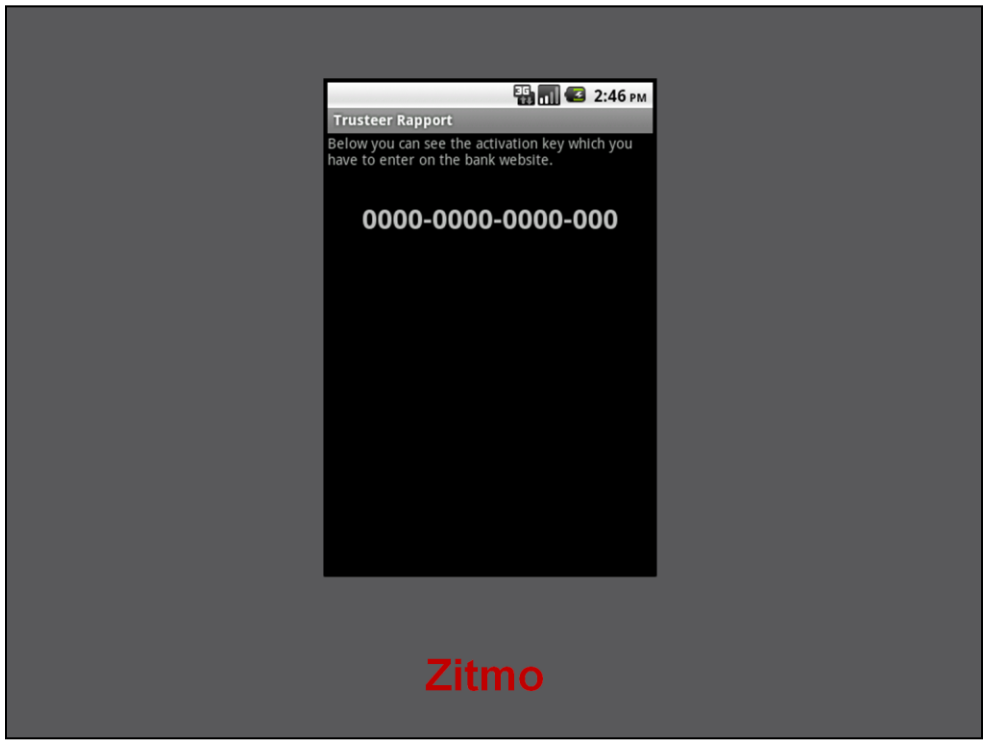
Since its launch in 2009 Android has been massively successful. Today in 2012 over half of all US Smartphones run Android, with Symbian running on 1% of them.

And just as ordinary developers flocked to make use of Android, so too did malware writers. Over the period of 2009 to 2010 Java malware and then Android malware rose to outrank Symbian in terms of malware variants.

And the overwhelming majority of these malware were financially motivated. The Era of experimentation and playing around was over – by 2009 the professional mobile cybercrime era had arrived.

At first these changes were subtle – the majority of mobile malware still used high cost text messages to generate money. The first Android malware, known as FakePlayer made a subtle change to that – by sending SMS messages to Adult related sites. This also helped to prevent victims complaining about their bills, potentially resulting in these numbers being blocked. Later another malware, this time for Windows mobile change the game again – by becoming the first mobile malware to directly phone high cost numbers.

The real era of mobile cybercrime started however, when Zitmo entered the scene.



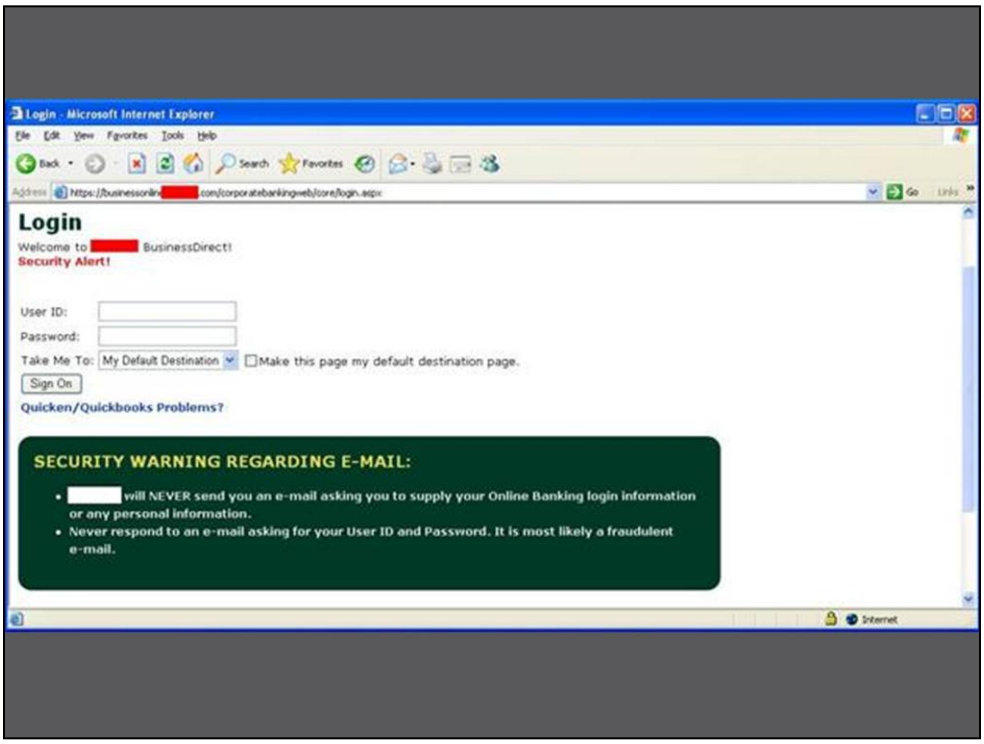
This is Zitmo. Its really not much to look at is it. However – it is the single most important mobile malware to date. Incidentally this is the Android version of the trojan – it is also available for Symbian, Windows Phone and others.

So what does Zitmo do? Well – it intercepts SMS messages, and sends them to the attackers numbers. Big Deal you think – Spyware like Flexispy has had that ability for years. The difference here is that Zitmo does not intercept every SMS it ONLY intercepts SMS messages coming from Banks

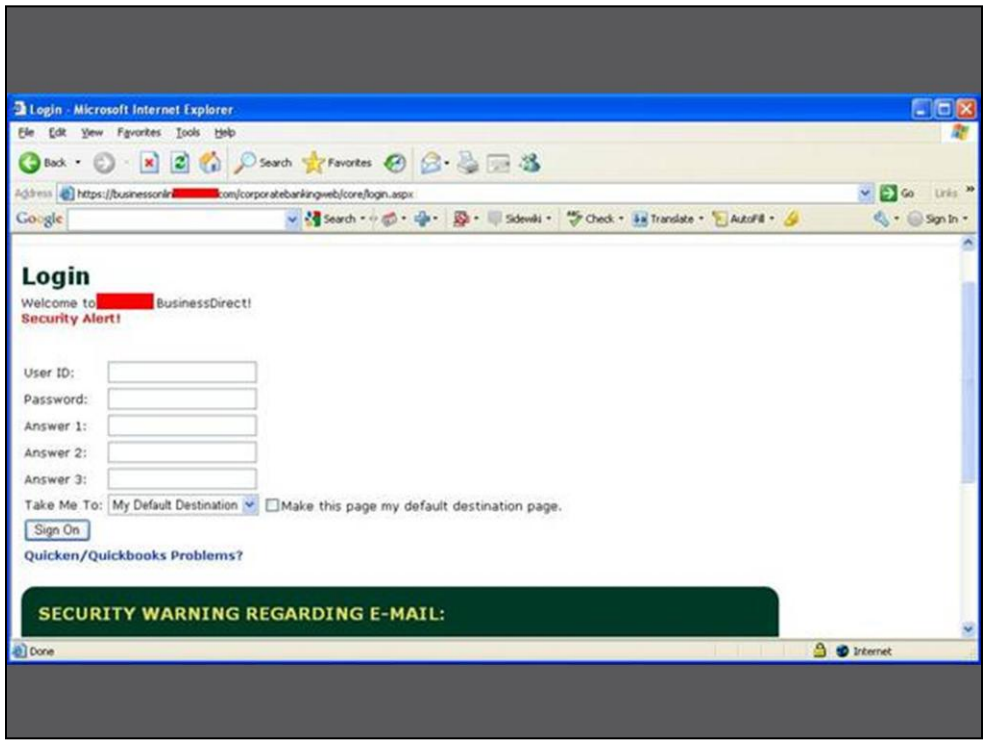
Lets step through an example attack.

Who here is familiar with the Zeus malware? (Show of hands – and ask to describe).

Lets look at how Zitmo works in the course of standard Zeus attack



First the victim, who is already infected with the Windows or Mac version of Zeus visits their banking or financial page. Normally their login screen looks like this.



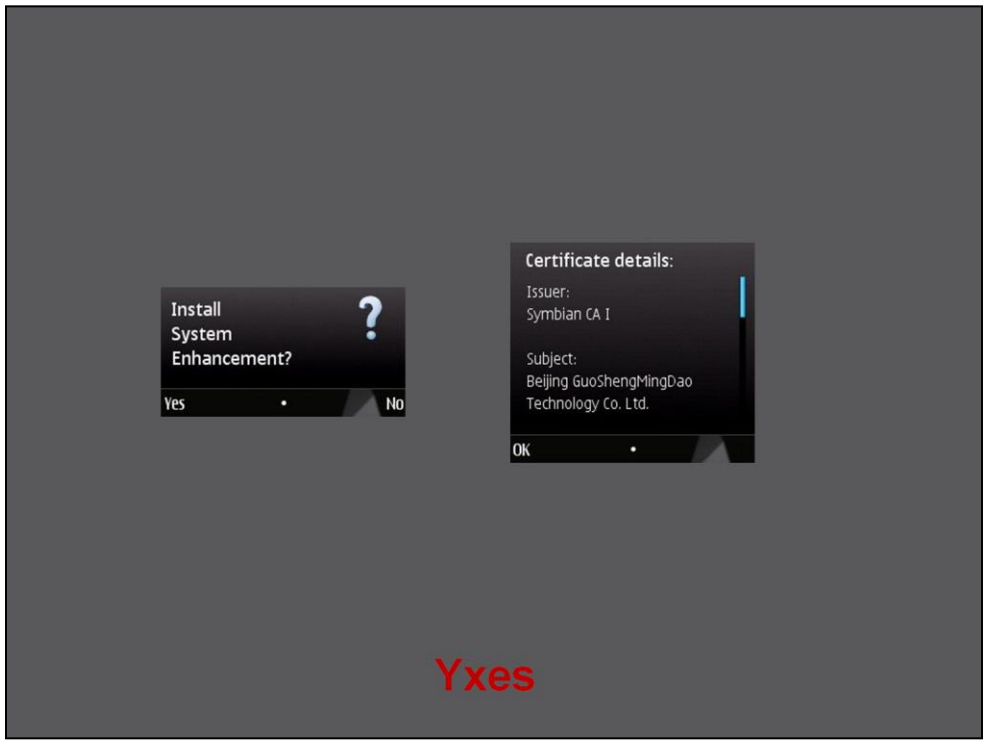
But when they are infected their login page will have different fields. These fields will be unique to each bank that is targetted.

Using this technique the attackers now have all the details they need to login to the persons account and create a transaction wiring money to their own accounts. There is a catch however – some banks will now send a text message to the users phone which they need to enter in the website to confirm a transaction.

To get around this the attacker first discovers the victims phone number (normally by injecting an additional field on the bank login page). Next the SMS a link to that user telling them to download an app that is required by the bank for security reasons – but this is of course Zitmo.

Once Zitmo is installed on the victims machine the attacker simply logs into the victims account and creates a transaction. A SMS is sent to the victims phone, which is intercepted and sent to the attacker. Now he simply enters this number on the banking site, and walks away with the money.

Banking criminals are among the most professional of all Cybercriminals, so when they become involved in mobile malware it is a clear sign that Cybercrime is targeting mobile in a big way.

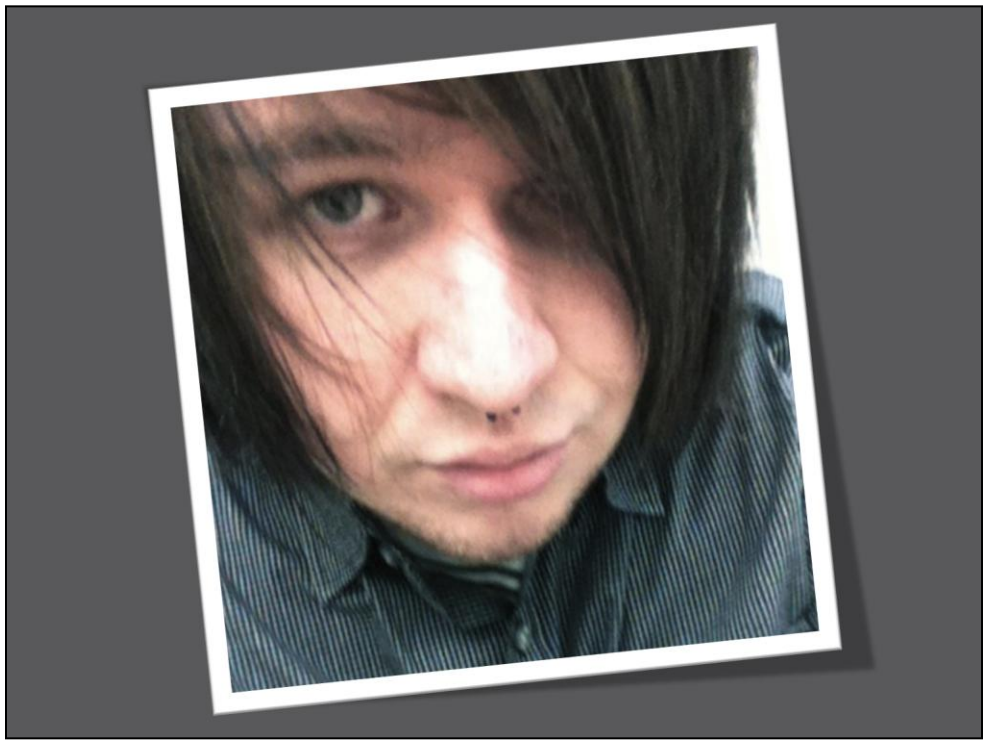


Of course not all malware from 2010-2011 where Android malware – there was still a healthy amount of Java and Symbian malware being released.

Probably the most notable Symbian malware for this period was Yxes

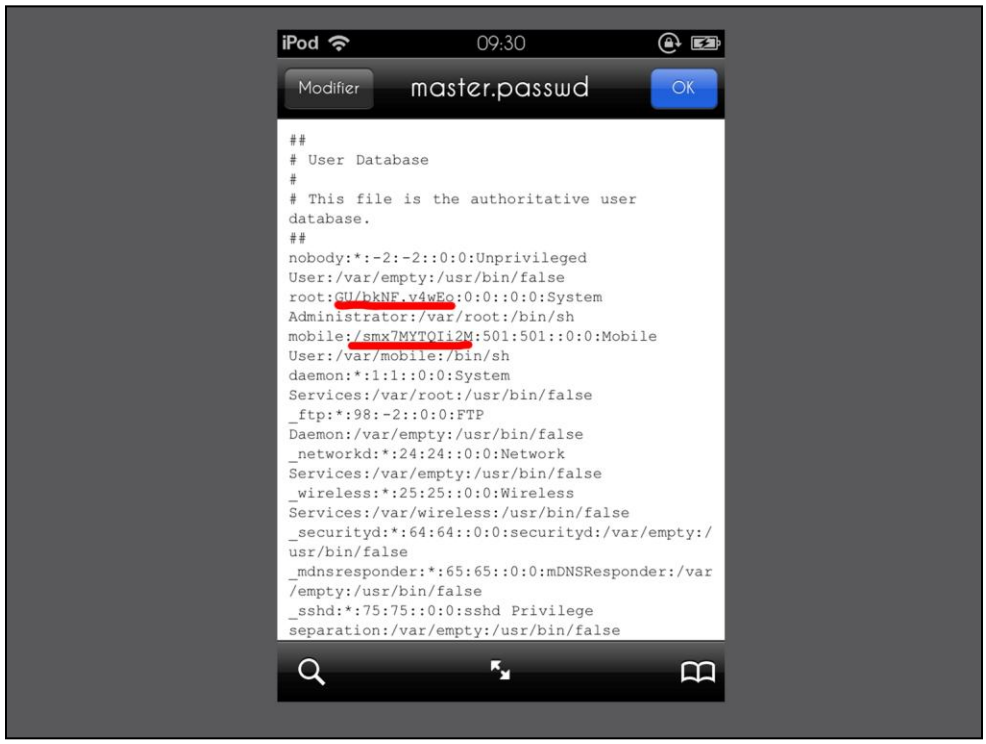
Like most mobile malware – there is not much exciting to see in the visual parts of the app itself. Also the main behaviour of this malware is not very interesting – its simply a SMS trojan. It also had a valid digital certificate which is important for installing on later Symbian versions.

However what is very interesting about Yxes is that it is capable of downloading new SMS templates from a remote server. Effectively this make Yxes one of the first mobile phone botnets. Actually the first mobile botnet was the Booton Symbian malware from 2004. However it was only after the arrival of the iPhone in 2007 that our phones really were “always on internet devices”, which makes mobile botnets much more successful.



Speaking of the iPhone – let me introduce someone to you. This is Ashley Towns from Sydney, Australia. Ashley is a 21 year old student and novice programmer.

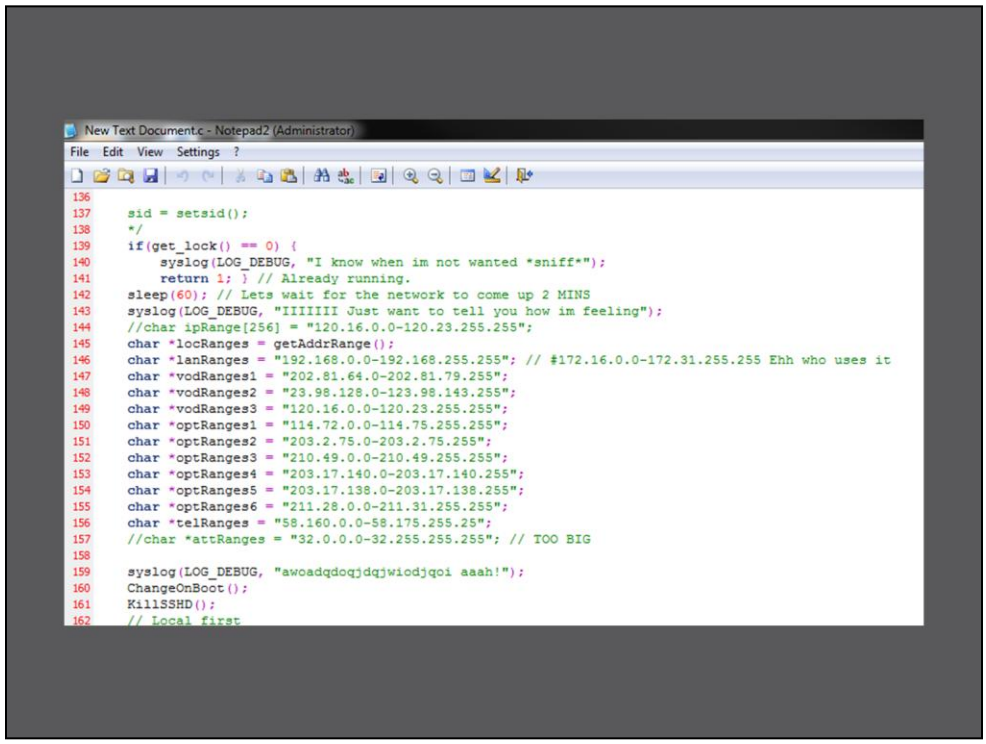
After getting buying an iPhone Ashley, like many people, decided to jailbreak the device. Jailbreaking is the process of removing the limitations Apple impose on devices running iOS – essentially it gives you root access on the phone. Normally this is carried out by some form of software or hardware exploit. Unlike many people however, Ashley decided to actually look at the modifications to the phone during the jailbreaking process.



One thing that Ashley noticed is that jailbreaking an iPhone add an SSH Server running on the phone, to allow a power user connect to the phone and directly interface with the underlying operating system.

Next he took a look at accounts that were on the device and discovered the password hashes stored in the master.passwd file. Simply cracking this hash showed him that the default password was “alpine”. The problem is Ashley was one of the very few people who had both jailbroken their phone, and bothered to look at those changes in depth. Almost everyone else with a Jailbroken iPhone was now running an SSH server with a default password.

Ashley decided to take advantage of this, and coded up a worm that would spread from phone to phone using the default password.



First an infected phone would scan the network the phone was on, and then also address ranges belonging to Optus, Vodafone, Telstra and other Austrailian profiders looking for iPhone with SSH running. It also picked 20 IP addresses at random to target outside of these ranges, so in theory could spread internationally.

And what would it do when it discovered a vulnerable iPhone? It would login, copy itself over and execute itself – spreading from this new victim. The malware payload also had one other payload, which is what it is most well know for...

It changed the background of all infected phones to a photograph of 80s singing sensation Rick Astley, whose music video for the song “Never going to give you up” has lead to the internet phenomenon known as Rickrolling.

The aftermath of this worm has led to several things

-Ashley Towns was hired by an Austrailian mobile phone application company.

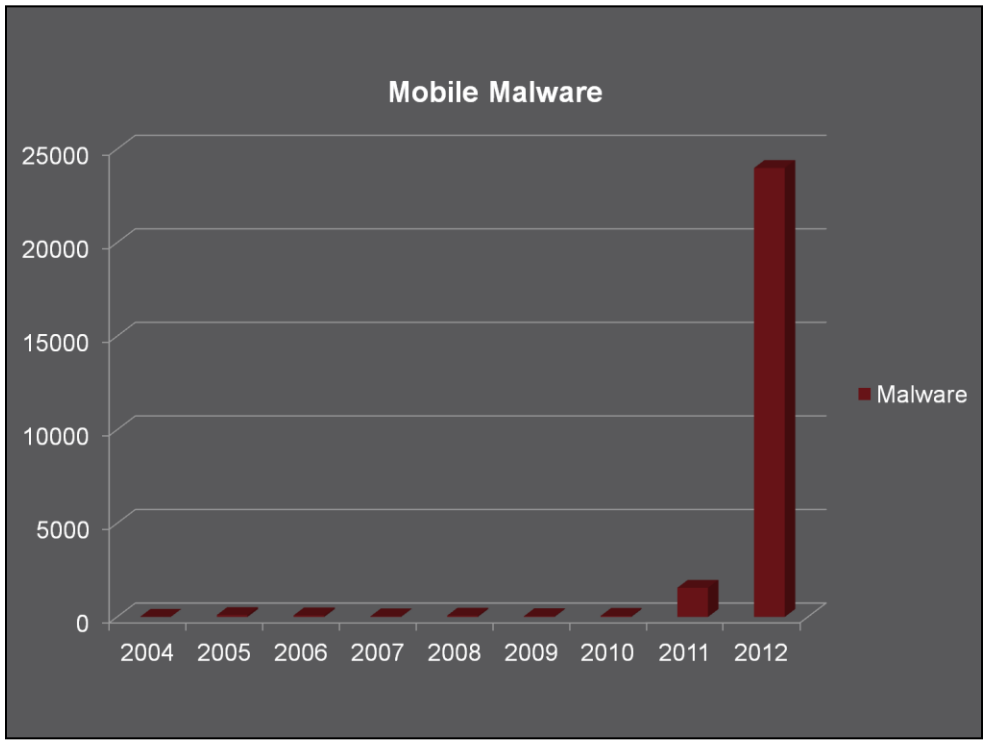
-IOS become yet another operating system that could no longer claim to be virus free.

-Ashley also released the source code. It has since been modified in a Dutch themed attack which also added Backdoor functionality and would redirect any internet connections to the Dutch ING Bank to a phishing page.

So all of that brings us right up to the modern era

In 2011 we saw more of the same attacks we have talked about so far, in particular SMS trojans

There was one major difference in 2011 however..



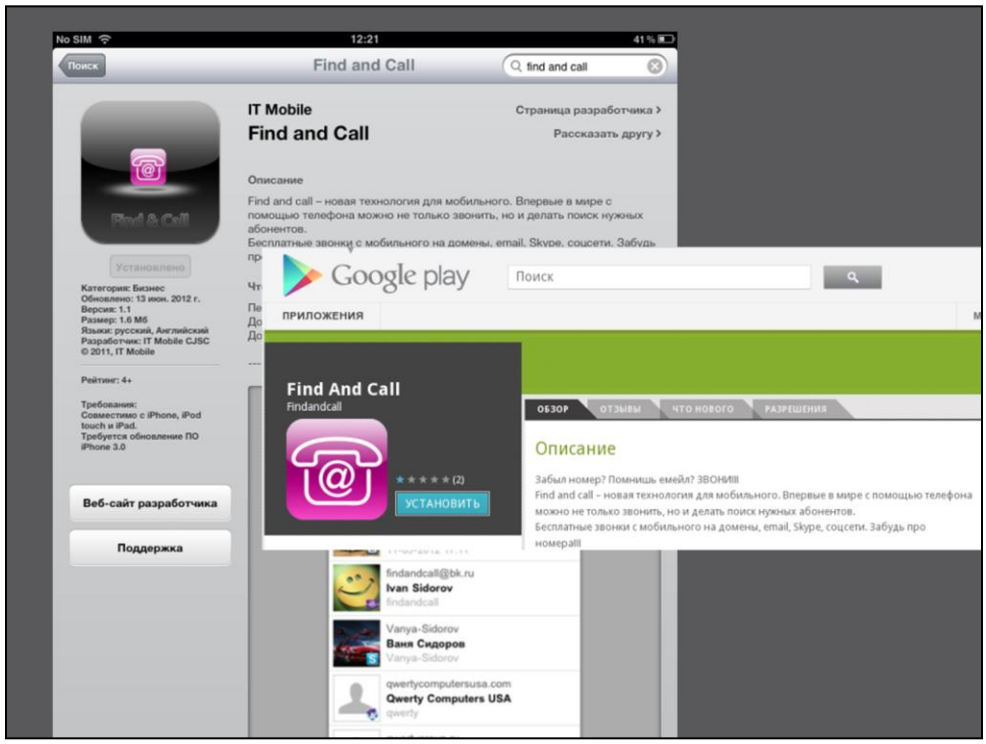
The amount of Android malware has absolutely exploded. At the same time Symbian and Java malware have basically faded out of existence. Once Android hit the magic mark of having over 50% of all smartphones – our 3 rules for malware to be successful kicked back into gear. We now once more had a platform that had great development support, some serious security concerns – and most importantly was very popular. Mobile Cybercrime has been searching for a standard platform – and now they had one.

What are these malware doing – well just about everything

-SPAM Botnets, SMS, Backdoors, Spyware, Worms, Intercepting calls etc.

Two common trends is that Android malware tends to give the attacker root access to the device, and also backdoors are very popular – which is normally a warning sign that botnets are about to become a big problem.

Incidentally – in 2012 things are even worse – we saw more malware in June alone than in the entire 7 years before that



Malware has even shown up in both the official iPhone app store, and the official Google Play store. In fact Google play has had at least 17 malicious apps uploaded which have all had over 700,000 downloads in total.

New ways of spreading mobile malware are also been seen – for example the use of QR Codes. When you scan a QR code with your phone it will decode the tag to ascii text. If that text is a contact card, the user will be prompted to add the contact. If it is a URL the site will normally open automatically. If it a URL pointing to APK file however (Android installer), most phones will straight away prompt the user to install the application.

The problem is that the user never sees the full the URL. So an attacker could for example print out QR Code stickers and place them all around a McDonalds. People would scan them expecting to get a McDonalds app – but instead would receive malware.

In Late 2011 we also say the first mobile malware motivated by Hacktivism. This malware was purely politically motivated, and this trend is highly likely to continue in 2012.

In this case, the malware know as Arspam hid in a Trojan application. Like many other malware this was simply a SMS trojan, which would send out links to other mobile users.

What was unique about ArSpam however was the political angle. All of the links lead to forum articles talking about the sacrifice of Mohamed Bouazizi, a Tunisian street vendor who set himself on fire on 17 December 2010 in protest over the confiscation of his wares and harassment from local authorities. His act became the catalyst for the Tunisian revolution, and ultimately the wider Arab Spring movement. ArSpam was released on the 1 year anniversary of this incident.

The Trojan also checked the country code of the phone, and if it was from Bahrain it would open a PDF on a report from the Bahrain Independent Commission of Inquiry on the violation of civil rights.

Although this is the first hactivism focused mobile malware I doubt it will be the last.

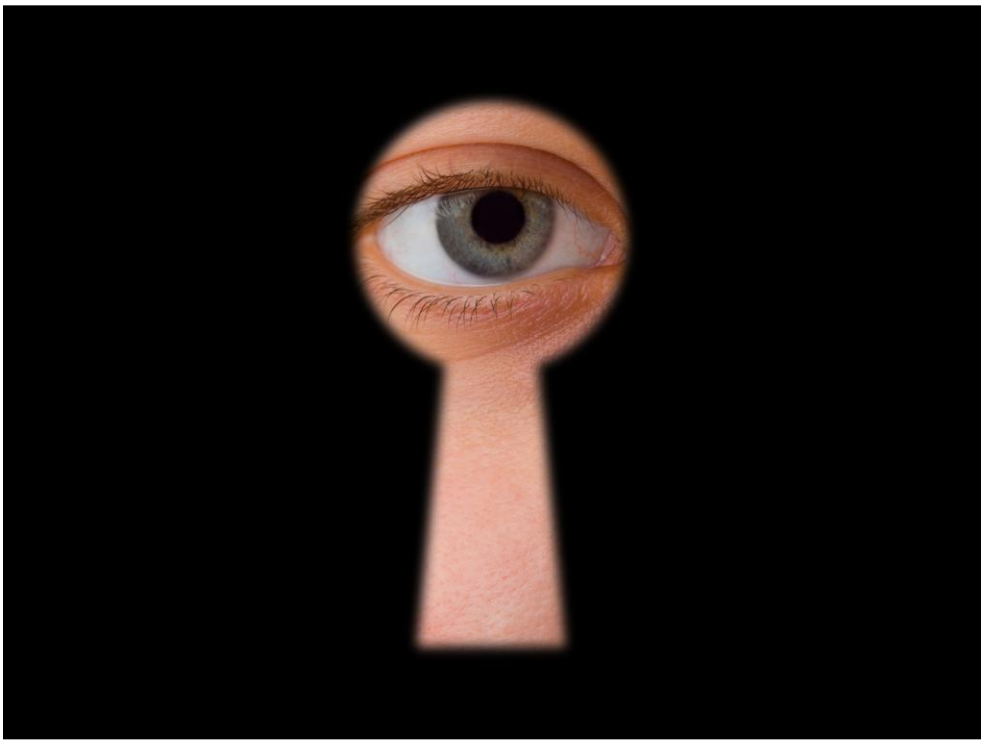
So Mobile malware has come along way since Caribe in 2004. Lets wind down this part of the workshop by talking about what we see as the trends in mobile malware in 2012.

First off the OS of choice is clearly Android – the openness and popularity of the platform make it perfect for mobile malware.

Secondly the vast majority of mobile malware is now financially motivated, although the methods for making money differ

You have a variety of

- SMS Trojans to high cost numbers
- Trojans that also dial high cost numbers
- Trojans generating money from SPAM
- And of course banking trojans such as Zitmo

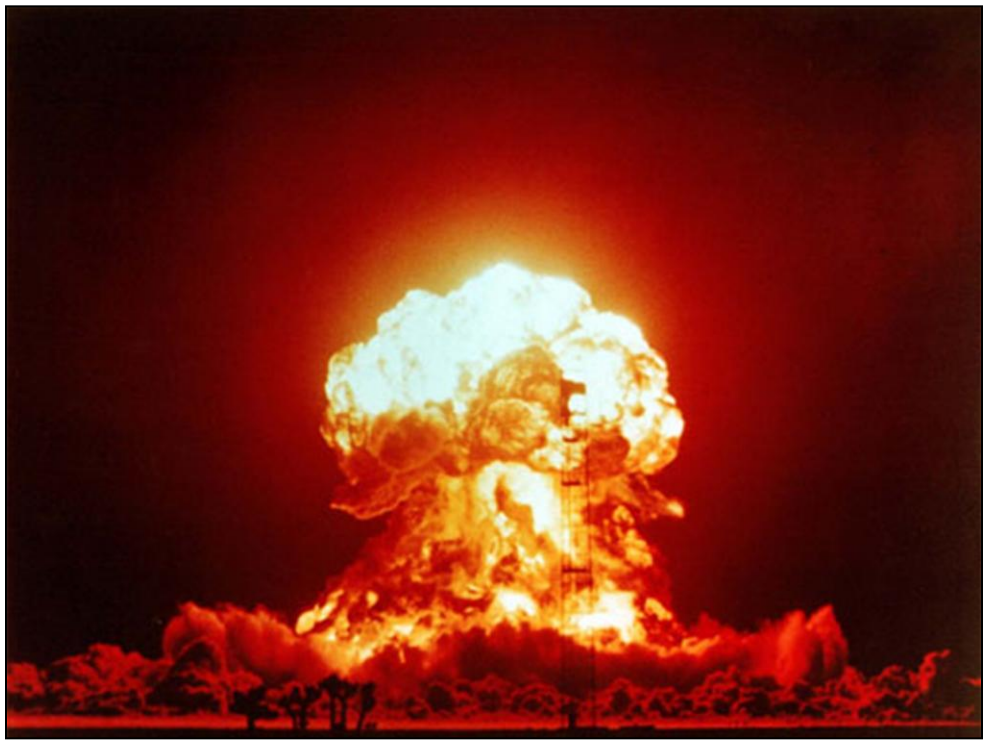


The second category of mobile malware is Spyware

Modern mobile Spyware can

- Steal everything from the phone
- Track everything the person does
- Intercept all of the persons communications

Whether its your address book, or email, your physically location, or finding out what you look like from turning on your camera – all of that is a reality for mobile spyware.



Destructive malware has almost entirely died out. While malware does exist that can wipe, lock and corrupt the phone – just as was the case in Windows malware, with the emergence of professional cybercrime on mobiles – destructive malware has pretty much stopped. Financial malware is simply too profitable to justify simply destroying the phone.

And lastly Hacktivism has seen its first appearance on a mobile phone, although with groups such as Anonymous – it is highly unlikely this will be the last time mobiles are used for this purpose.

So lets wrap up this part of the workshop with a demo of mobile malware in action.

[DEMO TIME]