

Symbian phone Security

Job de Haas ITSX



Overview

- Symbian OS.
- Security Risks and Features.
- Taking it apart.
- Conclusions.



Symbian History

- Psion owner of EPOC OS, originally from 1989, released EPOC32 in 1996
- EPOC32 was designed with OO in C++
- 1998: Symbian Ltd. formed by Ericsson, Nokia, Motorola and Psion.
- EPOC renamed to Symbian OS
- Currently ~30 phones with Symbian and 15 licensees.



Symbian Organization

- Symbian licenses the main OS
- Two GUI's on top of Symbian:
 - Series 60, led by Nokia
 - UIQ, subsidiary of Symbian
- Ownership:

Nokia	47.5%	Panasonic	10.5%
Ericsson	15.6%	Siemens	8.4%
SonvEricssor	13.1%	Samsung	4.5%



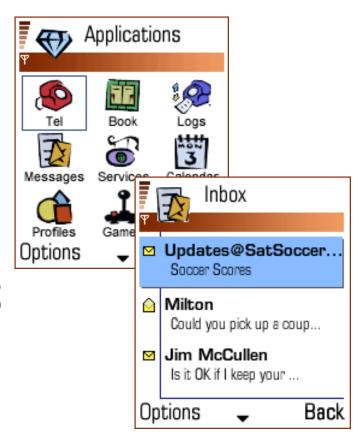
Symbian Versions

- EPOC32
- EPOC R5
- Symbian v6.0
- Symbian v7.0
- Symbian v8.0
- Symbian v9.0 announced for Q3 '05



Series60 versions

- 1st edition
- 2nd edition
- 3rd edition, announced feb. 2005





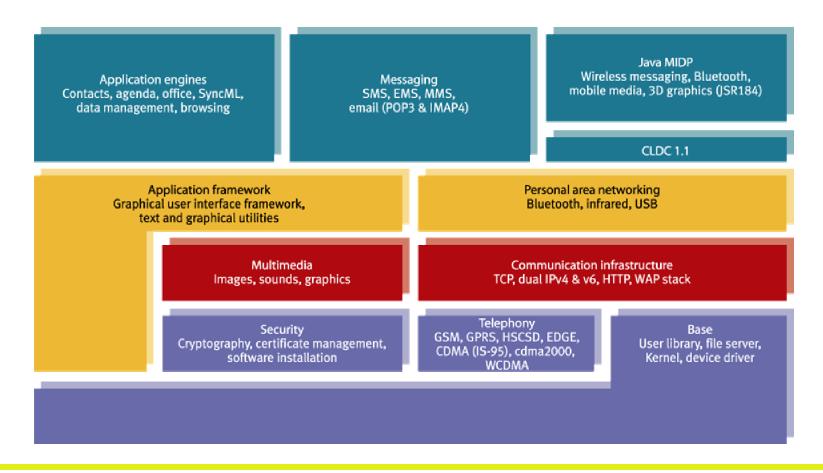
UIQ versions

- UIQ 1.0
- UIQ 2.1
- UIQ 3.0 released feb 2005





Symbian OS





Symbian OS

- Multitasking, preemptive kernel.
- MMU protection of kernel and process spaces.
- Strong Client Server architecture
- Plug-in patterns
- Filesystem in ROM, Flash, RAM and on SD-card



Symbian development

- Symbian v6 and v7 are compiled with a modified GCC.
- The base environment consists of a set of Perl scripts and command line tools.
- IDE's interface to that:
 - Metrowerks CodeWarrior
 - MS Visual Studio
 - Borland C++ Builder



Symbian development

- Emulator on x86 runs a majority of the native code base.
- Compiled to x86 (so not running ARM)
- Emulator is one windows process.





Symbian development

- Limited support for on-target debugging
 - It does not work on all devices.
 - Uses a gdb stub.
 - Metrowerks provides MetroTRK.
- Future: v9 will move to ARM Real View (RVCT) and the EABI standard.



Mobile phone risks

- Toll fraud:
 - Auto dialers.
 - High cost SMS/MMS.
 - Phone Proxy.
- Loss or theft:
 - Data loss.
 - Data compromise.
 - Loss of Identity (caller ID)



Mobile phone risks

- Availability:
 - SPAM.
 - Destruction of the device (flash)
 - Destruction of data.
- Risks induced by usage:
 - Mobile banking.
 - Confidential e-mail, documents.
 - Device present at confidential meetings: snooping



Attack vectors

- Too many entry points to list:
 - Executables
 - Bluetooth
 - GPRS / GSM
 - OTA
 - IrDa
 - Browser
 - SMS / MMS
 - SD card
 - WAP
 - E-mail



Security features

- Crypto:
 - Algorithms
 - Certificate framework
 - Protocols: HTTPS, WTLS, ...
- Symbian signed:
 - Public key signatures on applications
 - Root CA's in ROM



Symbian security features

- Separation
 - Kernel vs. user space;
 - process space
 - Secured 'wallet' storage
- Access controls
 - SIM PIN, device security code
 - Bluetooth pairing



Symbian security features

- Artificial Limitations / patches
 - Preventing loading device drivers in the kernel (Nokia).
 - Disallowing overriding of ROM based plug-ins.



Risks covered / mitigated

- Fraud: hardly, user should not accept unsigned apps.
- Loss/theft: In practice, none.
- Availability: hardly, any application can render phone unusable (skulls trojan).



What goes wrong?

- So-far all of known attacks have needed user confirmation. Often more than once.
- People loose a hell of a lot devices.
- Not that much so-far actually.



Viruses and Trojans

Skulls Trojan:

 theme that replaces all icons and can not be de-installed.

• Caribe:

 Installs itself as a 'Recognizer' to get activated at boot time and starts broadcasting itself over Bluetooth.



Problems in Symbian

- No concept of roles or users.
- No access controls in the file system.
- No user confirmation needed for access by applications.
- User view on device is limited: partial filesystem, selected processes.
- Majority of interesting applications is unsigned.



Symbian security features

- Future: v9:
 - Data 'caging': per application file system access
 - Capabilities tied to signing:
 - Limited access to sound.
 - Only signed apps get access to keys.
 - OMA DRM v2.0
 - Suspect: tying to ARM TrustZone: hardware assisted code protection.



Risk improvements

- Limits the damage an application can do.
- Improvements for data access and integrity.
- Will also limit what you (as a developer) can access on the phone without going through certification.



Still things go wrong

 February 23, 2005 a notice appeared on the Nokia site:

The Java Verified testing and certification program uses a UTI root certificate (also known as "GeoTrust CA for UTI"). This root certificate is missing from the Nokia 6600, Nokia 6230, and Nokia 3220 devices. When this root certificate is missing, the Java Verified tested applications cannot be installed.



Their 'work around'

Solution / Workaround

If a MIDlet has been signed with a certificate for Java Verified applications, it can be installed if the following attributes are removed from the JAD file:

- MIDlet-Certificate-1-1
- MIDlet-Jar-RSA-SHA1

Without these lines the MIDlet is treated as untrusted (MIDP 1.0 MIDlets are treated as untrusted), and it thereby gives a warning message during the installation. This makes the application appear as if it were unsigned, but it is possible to install the application similarly as any unsigned application. When



The following figure shows the normal user prompt which is shown when an untrusted MIDlet is installed.





Unraveling a ROM

- Obtaining ROMs:
 - Vendor upgrades
 - Reading them from the device
- Understanding the ROM structure:
 - Base porting guide (7.0 help file)
 - Header files



Extracting the ROM

- Creating a Device driver on a P900
- Read the MMU Page tables.
- Copy physical memory out to memory card



Unpacking a ROM

- Small toolkit with Python scripts
- Extracts files from a ROM
- Creates a browse-able interface with details and relation between files.



Demo



Examining binaries (OS)

- Use the emulator versions with debug info.
- Translate names to ordinals with Libs.
- Import them in IDA
- Load the full ROM at once or use a DB to name direct references to other files.



Conclusions

- Currently Symbian is not prepared for serious attacks.
- Embeddedness and OO implementation raised the bar for attacks.
- Relatively few tools available for dissecting Symbian ROMs and applications.
- V9 promises much needed improvements