

The logo for Nixu, featuring the lowercase letters 'nixu' in a white, sans-serif font. The letters are positioned to the left of a vertical white line that extends upwards from the bottom of the 'u'. The background is a solid dark blue.

nixu

Active content,  
Mobile, E-commerce  
and Convergence  
Security

# Agenda

- Executable content
- Mobile security
- E-commerce and web security
- Convergence of telecommunications and data communications

## Executable Content, Definition

- Executable content is received data which is run on the client host
  - automatic execution
  - usually received from a WWW page
- Security is a problematic issue
  - how to prevent the received program from doing nasty things?
    - using all CPU capacity
    - deleting files
    - reading and sending files to external users
    - editing files and security parameters
  - signed code helps some but not all

# ActiveX

- Microsoft active technology
- Basically very little security
- Idea of small controls, i.e. functional components
  - buttons, labels, charts etc
- Control security
  - loaded from disk, if not there fetched from the net
  - control is signed by a CA and the signature checked by the client
- What about signed but malicious controls?
  - examples can be found

# Java

- Sun Java technology
- Java is many things
  - an object oriented programming language
  - run time environment
- Security is handled by a sandbox ideology
  - the program runs in the sandbox
  - user defines the sandbox limits
  - Java Virtual Machine guarantees the sandbox
- JVM must be compiled/plugged in the browser
- Client executable code is called an applet

# Javascript

- Not related to Java
- A scripting language created by Netscape used in Web pages
- Microsoft has a non-compatible Jscript
- WAP has WMLscript
- More limited action possibilities than Java/ActiveX
- However, no sandbox-like security features!

# Problems with Executable Content

- Computation is moved to the client
- Clients need to be protected from rogue service providers
- Mobile code moves from host to host, executing a task given to it
  - Clients must be protected from malicious mobile applications
  - The mobile code must be protected from a malicious host
- Users are forced to become administrators and policy makers
- Executable content keeps on appearing
  - Proxlets
  - Active networks
  - Agents

# CGI and Other Server Side Code

- Code is executed in the server
  - Bugs can compromise the server (intrusions)
  - Execution requires computational resources from the server (denial of service)
- Many scripts are written by people who know little or nothing about security
- If you are using CGI scripts or servlets:
  - Keep track of what scripts you have, remove the ones you do not need
  - Control the access rights the scripts have (don't run them as root or administrator!)
  - Do source code security review if possible

## Signed code

- The program is digitally signed
- Signature keys are certified
- The browsers come with certification root keys
  - It is easy to delete and add more root keys
- With signed code, you probably know who wrote the program
- With signed code, you **DO NOT KNOW** if the code is malicious or not

## ActiveX Authenticode

- Microsoft's solution for securing executable content
- Code is signed
- Browser asks user whether to allow the downloaded code to run or not
- If the user accepts the certificate, the software is allowed to run without any restrictions
  - It could delete all your files
- Problem: users often want to try a program even if they do not trust its source

# Java

- Applets may come from any source
- Users may want to securely run code they do not trust (they might not even know where it came from)
- Code is run in a restrictive sandbox, where it cannot do harm
  - No access to files, obtaining information about the user or network connections
- The Java programming language was designed with security in mind
  - Byte code verifier, class loader & security manager
- Sandbox implementations in browsers have had serious bugs

# Security Model of a Java Applet

- Java is a general purpose language, here we are looking at applet use
- Classloader in the run time environment differentiates between local (trusted) and network (applet) code
  - Local class is (should be) always preferred to network class
- Verifier checks the byte code
  - Byte code is the binary code compiled from the Java source code and native to the Java Virtual Machine
  - The Verifier attempts to find stack over and under flows, checks correct use of variable types and generally the syntax of the byte code
- SecurityManager implements the Java sandbox
  - Sandbox limits the applet's actions severely

# The Java Sandbox

- The applet in the sandbox may not:
  - Read or write files
  - Open network connections to hosts other than the originating host
  - Initiate execution of new processes or programs
  - Use any native methods
- Only trusted code (local classes) can use the OS services
  - Local library classes check if they are called from the sandbox or from a local applet running outside the sandbox
- Signed applets can exceed the sandbox limitations

# Mobility in the network

- A device may travel in an IP network
- Portable computers
  - Different IP addresses at different networks
    - IPSec has difficulties (depends on implementation)
    - Can not act as an server
  - Is outside the home network protection domain
- Mobile IP -technology
  - The device visiting a different network invites a home station to tunnel all packets to the device from the home network to the new IP address the device is using in the visiting network
  - To avoid connection capture and man-in-the-middle attacks these requests should be signed

## Mobile networks

- Currently various radio networks
  - GPRS, UMTS, WLAN, GSM data, Bluetooth etc.
- Some of these crypt the traffic over the radio link, others do no
  - WLAN crypto has been broken
  - GPRS backbone network is IP packets tunneled over IP packets, unencrypted
- The network itself is a target for attackers

# The Handset Programming Environment

- Handsets are opening up as programming environments
- The SIM toolkit kept the applications in the operator's control
- New phones have open applications environments
  - Symbian EPOC
  - PalmOS
  - Microsoft
- Mobile devices are typically limited by memory, CPU power and communications bandwidth
  - Implementing anything on these devices is going to be interesting
  - Implementing full scale PKI solutions is going to be demanding

## SIM Card

- The SIM card is a tiny computer (CPU and memory) usually owned by the telephone operator
- It usually contains the user's identity and cryptographic functions for authentication on the cellular network
- The SIM Application Toolkit
  - The SIM Toolkit is a standard for applications that can be included in the production stage or downloaded from the cellular network to the SIM
- This means that the SIM can contain security functions, like encryption or authentication
- On this platform it is possible to build commerce and other systems

# Handset side Security

- The main problem is interference between applications
  - In most environments it would be easy to write a trojan horse which would target a popular banking application on the handset
  - The proposed programming environments do not offer tight security
- The possibility of internal firewalls in handsets is not too remote
- Viruses are sure to appear in the future
  - A Bluetooth virus would interestingly spread like a medical virus
- Handsets will also be used as interfaces to ERMs and other information systems

## Who controls the handset?

- The manufacturer decides the environment
- The operator owns the tamper resistant SIM
- The owner (company) is in theory in control
- The user has the physical control

# Securing E-commerce

- E-commerce is an application over some infrastructure, like the Internet
- As an application it has several security needs
  - Security of the serving infrastructure technology
  - Security of the information in the server
  - Security of the transaction
  - Non-repudiation needs

# Types of E-commerce

- Business to Business
  - Typically medium size to large transactions and long term relationships
- Business to Consumer
  - Typically small to medium size transactions and loose relationships
- Consumer to Consumer
  - Typically small to medium size transactions, lack of trust between the parties and no prior relationships
- Different types of commerce prefer different solutions

## E-commerce servers

- WWW and e-mail are the most common applications
- Standard firewall and host security solutions can be used to secure the server
- The server often contains credit card information, customer addresses, business confidential data, pending orders etc.
  - Some credit card companies already require that the credit card information is located in a separate server
    - Front and back-end server architecture
  - Threats both to confidentiality and integrity

# E-commerce transactions

- Identifying the participants is often required
  - SSL authenticates the server
  - PKI systems could be used to authenticate both participants (once they are in global use)
  - PGP and S/MIME could be used, but are rarely used
  - Extranets can use PKI or usernames and passwords
  - Sometimes it is easiest to accept a certain amount of losses
- There are formal standards for B2B commerce
  - EDI/OVT
  - XML-based standards are emerging
  - PKI-based signatures are beginning to be used

## Non-repudiation in E-commerce

- PKI systems could provide electronic signatures
  - Many countries have laws about these
- What happens if the signer repudiates the signature?
  - The whole system may be evaluated in public court
- Transaction logs can be useful
- Instead of non-repudiation, how about pre-payment
  - In Finland the banks have rather flexible online systems
  - The credit card companies have different solutions, too
  - Remember that WWW forms and cookies are freely editable by the user

# Security in Telecom Networks

- Separate user and control planes
  - Technical implementation failed too often
  - E.g. hacking of in-band signalling
- Signaling security
  - Customer terminals assumed not be able to send inter-switch signals
- Signaling is truly international
  - Barriers between operators and countries
- Security by obscurity
  - A handful of key persons have vast amounts of information

# Security in Data Networks

- Conventional reusable passwords
  - Subject to eavesdropping
- One-time passwords
  - Subject to connection hijacking
- Filtering based on originator addresses
  - Subject to IP spoofing and similar attacks
- Real firewalls (application proxies)
- Cryptographic protocols
  - The only real method of protecting end-to-end traffic in any open network

# Integration and New Problems

## Telecom network perspective

- No more user/control plane separation
  - Signaling and user data intermixed
- Borders between operators blurred
  - There are separate inter-operator and intra-operator routing etc. protocols
  - However, nothing blocks signalling data
- Terminal equipment much more intelligent
- Networks extended to customer premises
  - Physical protection not any more the same

## Data network perspective

- Accounting data means real money
- Signaling spoofing may mean real money
- Telecom network more lucrative target for
  - Cover up operations
  - Terrorist attacks
- More eavesdropping possibilities
- Possibilities to create more covering profiles by eavesdropping and data analysis
- More covert channels
  - Data can be easily hidden in digital speech or video

## Summary of the new risks

- Physical and logical protection weaker than in Telecom nets
- Signaling integrity importance grows
- Potential gains from attacks increase
- System is more complex, and therefore harder to manage
- New services create new possibilities for fraud
  - E.g. ATM Virtual Circuits create new virtual connections between networks

## Future View: Basic security after convergence

- Data and Telecom Networks integrated
  - Signaling integrated, accounting combined
  - Signaling protected cryptographically
- Accounting integrated, possibly through a millicent system like ecash in order to reduce delay
- Integrated, fast firewalls everywhere (hardware FW)
- User data protected cryptographically
- A couple of problems left
  - How to manage cryptographic keys?
  - How to manage firewall access control?

# Conclusions for future communications

- Traditional telecom security based on
  - Distinct user and control planes
  - Security by obscurity
- Traditional data security based on
  - Passwords and other weak technologies
  - Firewalls and cryptography
- Integration brings in new problems
- Only real solutions are based on cryptography
- Managing crypto poses a problem
- Trust certificates seem like a promising solution

## Future of Security

- Cryptography and PKI are seen currently as the silver bullet to solve all problems
  - PKI is more complex than originally thought
- It has been said that this is the “golden age” of hacking and cracking
  - Current and future systems will have security included from the start of the design process, not as an afterthought
- In the future security services are going to be more clearly defined and easily available
  - Security is an infrastructure service
- However implementing security will continue to require know-how in the foreseeable future