



Equinox Security

Matt Flaherty, IBM Lotus

August 6, 2008



Agenda

- Introductions and background
- Vision for Equinox security
- Equinox 3.4 security features
- Next steps for Equinox security
- Question & Answers



Introduction

- Matt Flaherty – IBM Lotus
 - Iris/Lotus/IBM employee ~10 years
 - Notes, Domino
 - Security



Background

- Lotus Notes 8.....

.....now based on Eclipse!

Lotus Notes 8



The screenshot displays the IBM Lotus Notes 8 mail client interface. The window title is "Mail - IBM Lotus Notes". The menu bar includes "File", "Edit", "View", "Create", "Actions", "Tools", "Window", and "Help". The toolbar shows various icons for mail actions. The left sidebar shows the "Dwight Masman" profile and a list of folders including "Inbox (188)", "Drafts", "Sent", "Follow Up", "All Documents", "Junk", "Trash", "Chat History", "Views", "Folders", "Archive", "Tools", and "Other Mail". The main pane shows a list of emails with columns for "Sender", "Subject", "Date", and "Size". The selected email is from Michael Melchar, dated 04/23/2007 03:48 PM, with the subject "Re: Sales Meeting at XYZ Corp". The email body contains the text: "I can make that meeting... see you there !!!!", "Michael Sloan", "ofc. 555-555-5555", "tie/line 555-555-5555", and a quote: "Success isn't a result of spontaneous combustion. You must set yourself on fire." Arnold H. Glasow. Below the email body, a conversation thread is visible, showing a previous message from Tara Troxell dated 04/23/2007 03:46:36 PM with the subject "It's scheduled for Friday, May 4th.". The right sidebar shows a list of contacts under "BluePages" and "Work (58/)", including names like "Dwight", "Elena", "Erica", "Jennife", "Kenneth", "Penny", "Bus. Dr", "Dev (2)", "Legal", "Market", "Arda", "Ang", "Bret", "Davi", "Dian", "Dwik", "Edm", "Geor", "Heid", "Jam", "Jasor", "Joar", "Kenn", "kristi", "Lindi", "Marc", "mar", "Marb", "mkt", and "Penr".

About Lotus Notes



- Collaborative application platform
- Over 20 years young
- Robust security model, key business differentiator
 - Collaborative means that users are empowered to create
 - One of the largest deployed PKIs in existence
 - Support for encryption, signing and execution control
- Integration with Java security model via standards



Vision for Equinox security

- Protect the user and platform from a variety of attacks in environments where not all code and users are friendly.
 - Examples of scenarios of concern:
 - File system breaches
 - Multi-user system scenarios
 - Malicious code packaged as bundles
- Extend the Java & OSGi security models
 - Provide additional services and user interfaces as appropriate
- Focus on usability and manageability



File system based attacks

- Sensitive data saved on the file system
 - Credit card numbers
 - Social security information
 - Medical records
 - Passwords
 -
- Compromised by viruses, physical OS access

Example of recent attacks



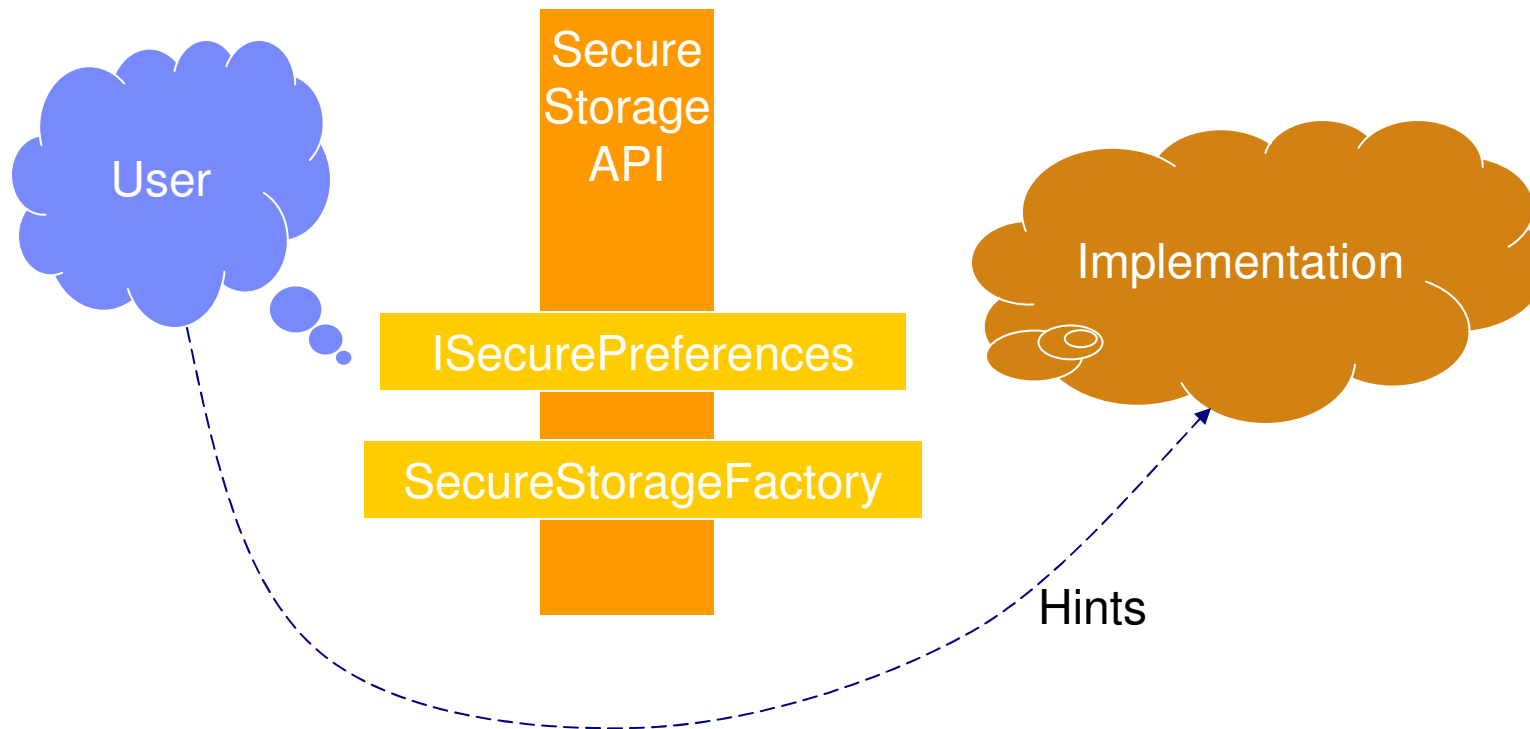
- 2006 – Veteran's Association
 - Laptop and external hard drive
 - 26.5 Million records
 - Burglary from employee home

Preventing file system based attacks



- Encryption can limit exposure
 - Scrambles data based on key
 - Without the key, data cannot be recovered
- Services in the Java platform, operating systems
- Equinox feature: **Secure Storage**

Secure Storage: Architecture



Secure Storage: Public API



- Available in `org.eclipse.equinox.security` bundle
- Code in `org.eclipse.equinox.security.storage` package
- Factory interface to obtain an implementation
 - `ISecureStorageFactory`
 - Pass in extension ID of a handler that collects the password
 - Default: `org.eclipse.equinox.security.ui.defaultpasswordprovider`
- Service interface similar to Preferences API
 - `ISecurePreferences`
 - Methods to get/put by name – boolean parameter for encryption
- Extensible via the `PasswordProvider` extension point

Secure Storage: Sample usage



```
import java.io.File;
import java.util.HashMap;
import java.util.Map;
import org.eclipse.equinox.security.storage.ISecurePreferences;
import org.eclipse.equinox.security.storage.SecurePreferencesFactory;
import org.eclipse.equinox.security.storage.provider.IProviderHints;

public class StorageSample {

    private static final String SECURE_PREFS_DEFAULT_ID = "org.eclipse.equinox.security.ui.defaultpasswordprovider"; //$NON-NLS-1$
    private static final String NODE_ID = "/eclipse.org/cvs/accountid"; //$NON-NLS-1$

    void usePreferencesSample() {
        try {
            File securePreferencesFile = File.createTempFile("storage", ".properties"); //$NON-NLS-1$ //$NON-NLS-2$
            Map options = new HashMap();
            options.put(IProviderHints.REQUIRED_MODULE_ID, SECURE_PREFS_DEFAULT_ID);
            ISecurePreferences securePreferences = SecurePreferencesFactory.open(securePreferencesFile.toURL(), options).node(NODE_ID);
            securePreferences.put("password", "mypassword", true); //$NON-NLS-1$ //$NON-NLS-2$
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

Secure Storage: Demo

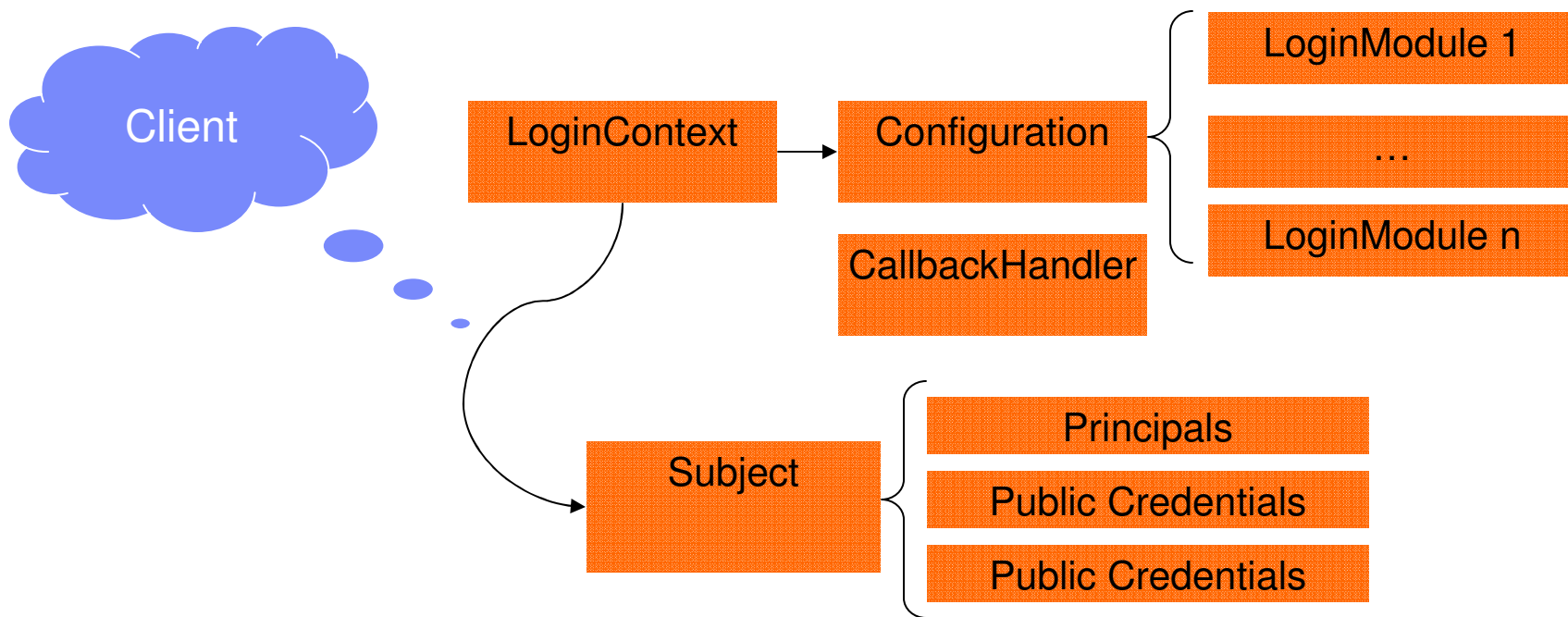




Shared system scenarios

- Kiosks, shared terminals, etc
- Require authentication to access the system
- Different behaviour based on user “roles”
- Equinox feature: **Login via JAAS**

Login via JAAS: Architecture



Login via JAAS: Public API



- APIs related to JAAS are in the `javax.security.auth` package
- Extension points to contribute the following JAAS artifacts
 - Configuration, LoginModule, CallbackHandler
 - Configuration -> CallbackHandler mapping
- Additional API in `org.eclipse.equinox.security` bundle
- Code in `org.eclipse.equinox.security.auth` package
- Interface that wraps JAAS `LoginContext` - `ISecureContext`
 - Adds ability to listen for Login/Logout events
- Factory class for creating `ISecureContext` - `SecurePlatform`

Login via JAAS: Sample usage



```
LDAP {
    org.eclipse.equinox.security.auth.module.ExtensionLoginModule required
        extensionId="org.eclipse.equinox.security.sample.ldapLoginModule"
        debug=true
        user.provider.url="ldap://localhost:10389/ou=Users,dc=example,dc=com"
        group.provider.url="ldap://localhost:10389/ou=Groups,dc=example,dc=com";
};

<extension
    id="ldapLoginModule"
    name="LDAP LoginModule"
    point="org.eclipse.equinox.security.loginModule">
    <loginModule
        class="com.sun.security.auth.module.JndiLoginModule"
        description="LoginModule for LDAP">
    </loginModule>
</extension>

<extension
    point="org.eclipse.equinox.security.callbackHandlerMapping">
    <callbackHandlerMapping
        callbackHandlerId="org.eclipse.equinox.security.sample.basicAuthDialog"
        configName="LDAP">
    </callbackHandlerMapping>
</extension>
```

Login via JAAS: Sample usage



```
import java.net.URL;
import java.security.PrivilegedAction;
import javax.security.auth.Subject;
import org.eclipse.equinox.app.IApplication;
import org.eclipse.equinox.app.IApplicationContext;
import org.eclipse.equinox.security.auth.ISecureContext;
import org.eclipse.equinox.security.auth.SecurePlatform;
import org.eclipse.swt.widgets.Display;
import org.eclipse.ui.IWorkbench;
import org.eclipse.ui.PlatformUI;

String configName = AuthAppPlugin.getConfiguratonName();
URL configUrl = AuthAppPlugin.getBundleContext().getBundle().getEntry(JAAS_CONFIG_FILE);
ISecureContext secureContext = SecurePlatform.createContext(configName, configUrl);

secureContext.registerListener(new ProgressMonitorListener());

Integer result = null;
final Display display = PlatformUI.createDisplay();
try {
    result = (Integer) Subject.doAs(secureContext.getSubject(), getRunAction(display));
} finally {
    display.dispose();
    secureContext.logout();
}
```

Login via JAAS: Demo





Malicious code attacks

- Code that may be malicious installed into the platform
 - Obtains access to sensitive data – KeyLoggers, etc
 - Replicates itself – Worms, Viruses
 - Sets up as an unauthorized service – Botnets
 - ...



Example of recent attacks

- 1999 - Melissa worm
 - Not intended to do harm – overloaded mail servers
 - Macro virus exploited mail clients

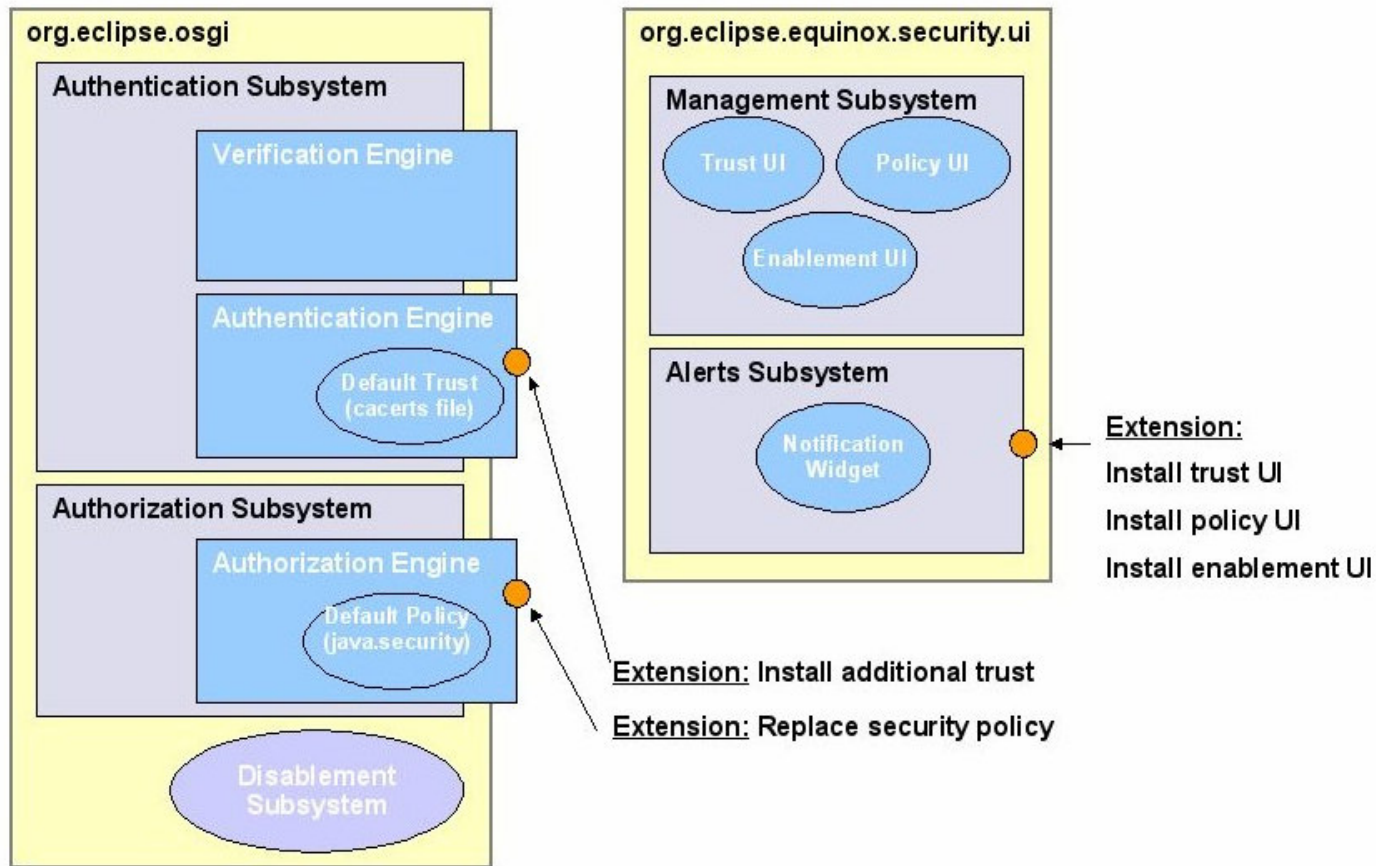
Preventing malicious code attacks



- A range of solutions are available for protection
 - Give deployers levers to trade security vs. performance/deployability
- Highest level of security: Java2 permissions
 - Built in support in the Java platform
 - Fine grained permissions, flexible configuration
 - Downsides: difficult to code, difficult to manage
- Highest level of deployability: Provisioning
 - Eclipse has built in support for signed code
 - Checks signatures as content is downloaded
- Equinox feature: **Trusted Bundles**



Trusted Bundles: Architecture





Trusted Bundles: Public API

- Available in `org.eclipse.osgi` framework bundle
- Code is in `org.eclipse.osgi.signedcontent` package
 - Factory class: `SignedContentFactory`
 - Signed bundle class: `SignedContent`
- Extensibility via services in `org.eclipse.osgi` framework bundle
 - `TrustEngine` as discussed earlier
 - `AuthorizationEngine` for implementing load policy
- Extensibility via services in `org.eclipse.equinox.service.security.ui`
 - `SecurityContributionItemFactory` for installing alert widget
 - `AuthorizationManager` for implementing user UI
- User interface for managing default policy

Trusted Bundles: Demo



Next steps



- Proselytize the Secure Storage subsystem
 - CVS account passwords
 - HTTP password manager – ECF
 - Repository for credentials - Higgins

- More certificate management functions
 - Feature parity with popular browsers
 - Public widgets and interfaces for certificates
 - Integrate trust management – P2, ECF

- Deeper User integration
 - Declaratively wire true RunAs into application lifecycle
 - Integration with Jobs framework, etc

- Enable Java2 permission infrastructure
 - End-user and Admin usability



Question and Answers

???



sample @ equinox-incubator/security/org.eclipse.equinox.security.sample

Disclaimer



- Copyright © IBM Corp., 2008. All rights reserved. Source code in this presentation is made available
- under the EPL, v1.0, remainder of the presentation is licensed under Creative Commons Att. Nc Nd 2.5
- license.
- IBM and the IBM logo are trademarks or registered trademarks of IBM Corporation, in the United
- States, other countries or both.
- Java and all Java-based marks, among others, are trademarks or registered trademarks of Sun
- Microsystems in the United States, other countries or both.
- Eclipse and the Eclipse logo are trademarks of Eclipse Foundation, Inc.
- Other company, product and service names may be trademarks or service marks of others.
- THE INFORMATION DISCUSSED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL
- PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND
- ACCURACY OF THE INFORMATION, IT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY
- KIND, EXPRESS OR IMPLIED, AND IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES
- ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, SUCH INFORMATION. ANY
- INFORMATION CONCERNING IBM'S PRODUCT PLANS OR STRATEGY IS SUBJECT TO CHANGE
- BY IBM WITHOUT NOTICE.