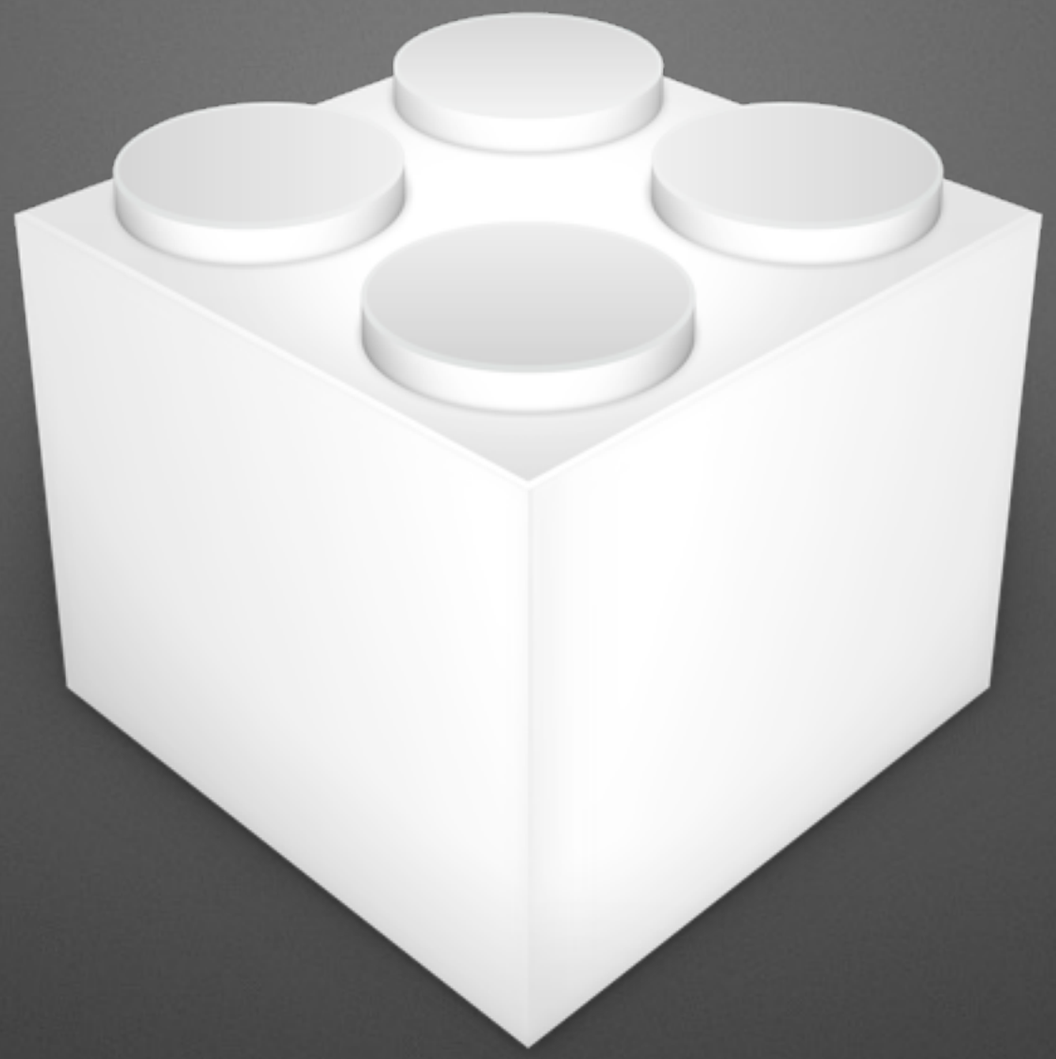
AuthorizationPlug-in:Programming

whoami:TomBurgin

Slack:tburgin
GitHub:tburgin
Twitter:@tomjburgin
Work:(NIH)SRAInternational



Authorization:Plug-in?

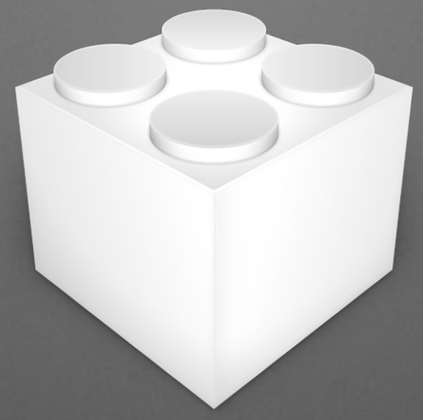
Plug-ins:GitHub

MagerValp/LoginScriptPlugin

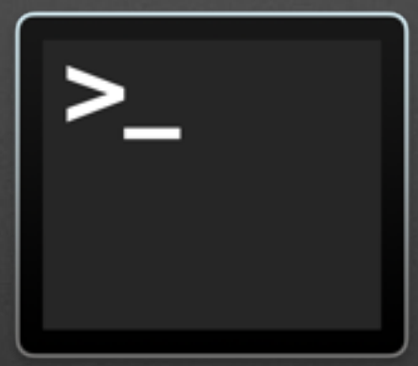
google/macops/tree/master/keychainminder

ygini/Unlock

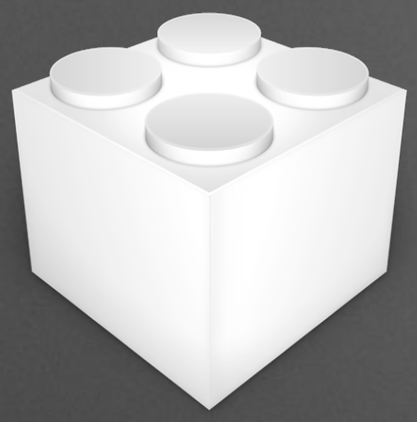
Plug-in:pieces



<string>.bundle on the file system</string>



<string>entry in the authorizationdb</string>



/Library/Security/SecurityAgentPlugins/


```
security authorizationdb read system.login.console
security authorizationdb write system.login.console
security authorizationdb read authenticate
security authorizationdb write authenticate
```

Today:functionality

!Today:Plug-inCore

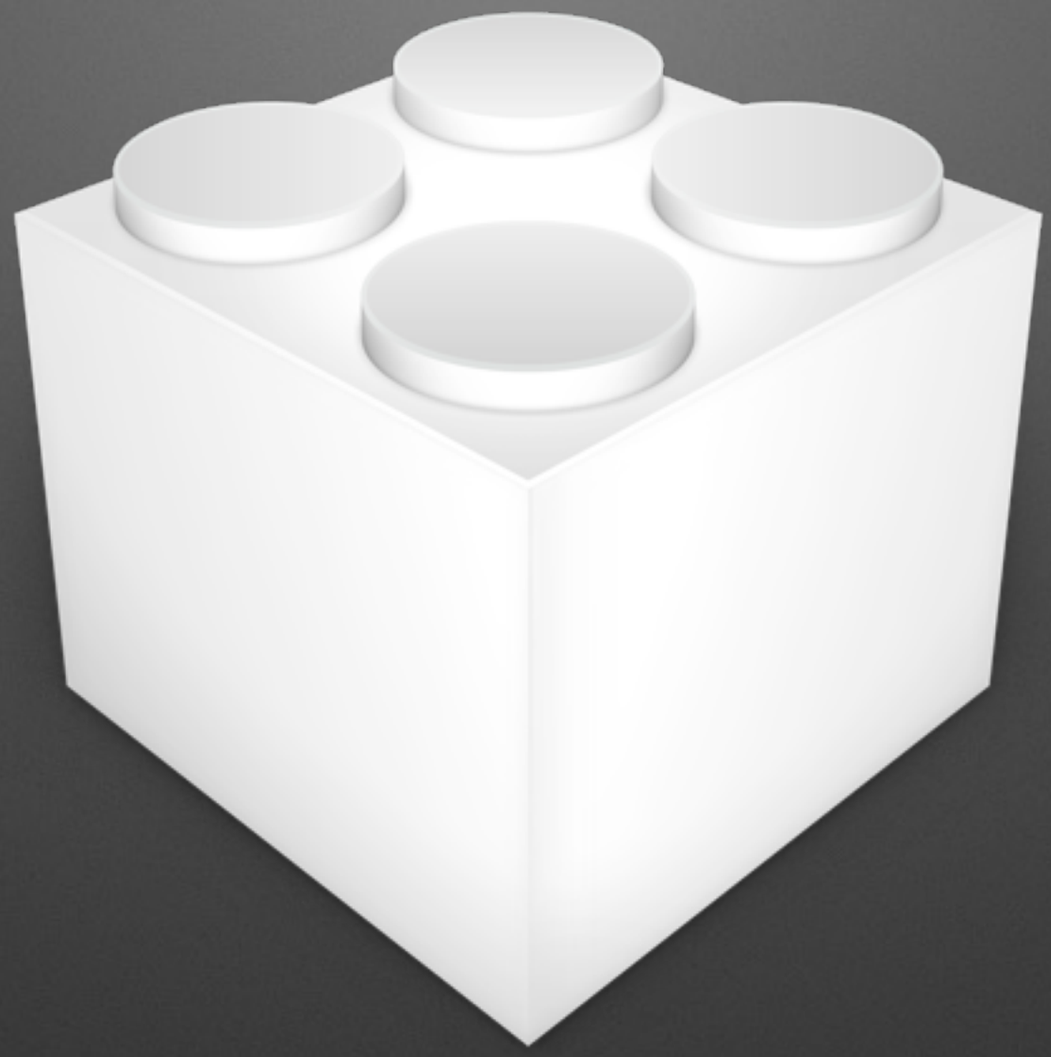
Let's make some decisions: Authorization Plugin Programming – Tom Burgin & Jeremy Baker

http://macadmins.psu.edu/conference/resources/

VerifyAuthPlugin:Goals

<goal01>Create 2 Mechanisms</goal01>
<goal02>Share Data between Mechanisms</goal02>
<goal03>Prompt user for PIN</goal03>
<goal04>Make an authorization decision</goal04>

VerifyAuthPlugin:Mechs



```
<string>VerifyAuthPlugin:MachinePIN,privileged</string>
<string>VerifyAuthPlugin:Verify</string>
```

```
security authorizationdb read system.login.console
```

```
<string>builtin:policy-banner</string>
<string>loginwindow:login</string>
<string>builtin:login-begin</string>
<string>builtin:reset-password,privileged</string>
<string>builtin:forward-login,privileged</string>
<string>builtin:auto-login,privileged</string>
<string>builtin:authenticate,privileged</string>
<string>VerifyAuthPlugin:MachinePIN,privileged</string>
<string>VerifyAuthPlugin:Verify</string>
<string>PKINITMechanism:auth,privileged</string>
<string>builtin:login-success</string>
<string>loginwindow:success</string>
<string>loginwindow:FDESupport,privileged</string>
<string>HomeDirMechanism:login,privileged</string>
<string>HomeDirMechanism:status</string>
<string>MCXMechanism:login</string>
<string>loginwindow:done</string>
```

```
<string>builtin:policy-banner</string>
<string>loginwindow:login</string>
<string>builtin:login-begin</string>
<string>builtin:reset-password,privileged</string>
<string>builtin:forward-login,privileged</string>
<string>builtin:auto-login,privileged</string>
<string>builtin:authenticate,privileged</string>

<string>VerifyAuthPlugin:MachinePIN,privileged</string>
<string>VerifyAuthPlugin:Verify</string>

<string>PKINITMechanism:auth,privileged</string>
<string>builtin:login-success</string>
<string>loginwindow:success</string>
<string>loginwindow:FDESupport,privileged</string>
<string>HomeDirMechanism:login,privileged</string>
<string>HomeDirMechanism:status</string>
<string>MCXMechanism:login</string>
<string>loginwindow:done</string>
```

tom

Tom Admin

```
<string>loginwindow:login</string>
<string>builtin:authenticate,privileged</string>
<string>VerifyAuthPlugin:MachinePIN,privileged</string>
<string>VerifyAuthPlugin:Verify</string>
<string>loginwindow:done</string>
```

Tom Admin

```
<string>loginwindow:login</string>
<string>builtin:authenticate,privileged</string>
<string>VerifyAuthPlugin:MachinePIN,privileged</string>
<string>VerifyAuthPlugin:Verify</string>
<string>loginwindow:done</string>
```

```
<string>loginwindow:login</string>
<string>builtin:authenticate,privileged</string>
```

PIN

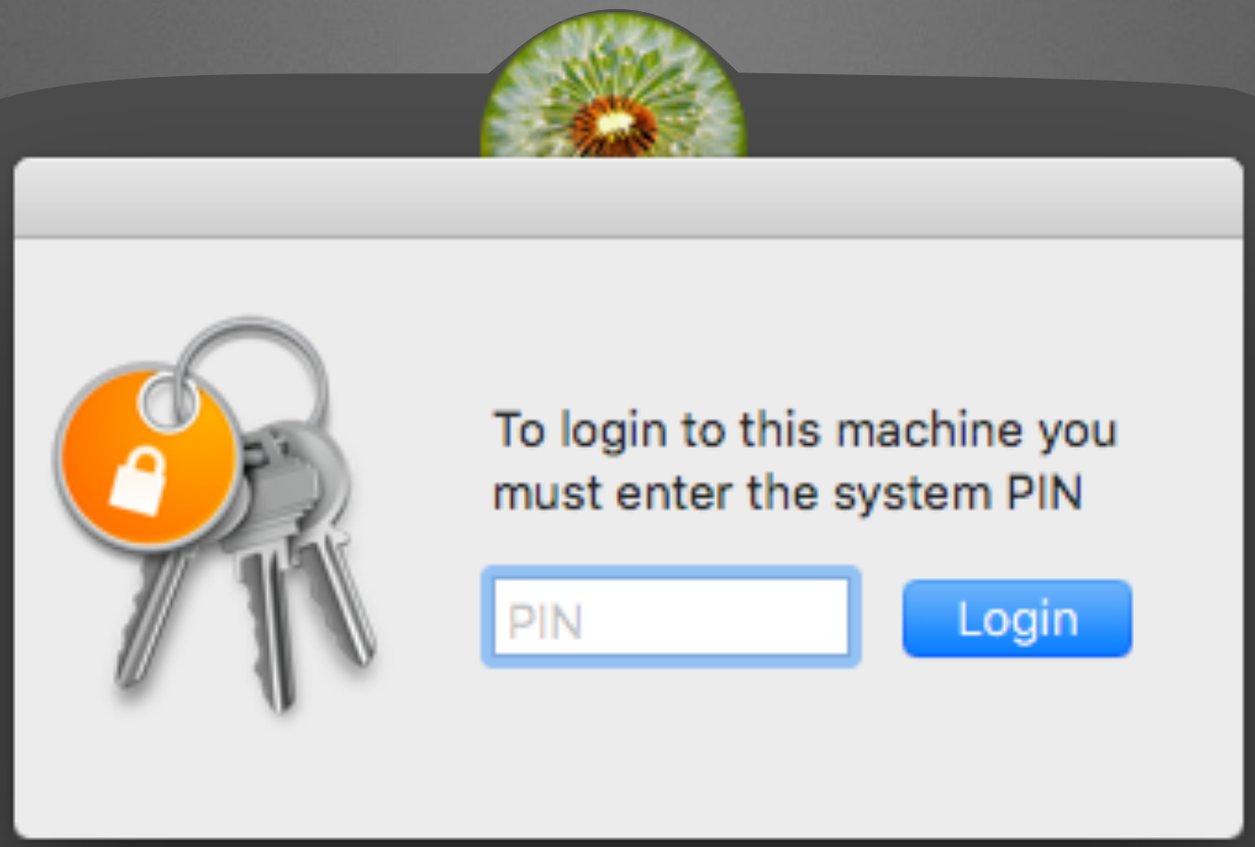
```
<string>VerifyAuthPlugin:MachinePIN,privileged</string>
<string>VerifyAuthPlugin:Verify</string>
<string>loginwindow:done</string>
```

```
<string>loginwindow:login</string>
<string>builtin:authenticate,privileged</string>
```

Verify ← PIN ←

```
<string>VerifyAuthPlugin:MachinePIN,privileged</string>
<string>VerifyAuthPlugin:Verify</string>
<string>loginwindow:done</string>
```

```
<string>loginwindow:login</string>
<string>builtin:authenticate,privileged</string>
```




```
<string>VerifyAuthPlugin:MachinePIN,privileged</string>
<string>VerifyAuthPlugin:Verify</string>
<string>loginwindow:done</string>
```

```
<string>loginwindow:login</string>
<string>builtin:authenticate,privileged</string>
<string>VerifyAuthPlugin:MachinePIN,privileged</string>
<string>VerifyAuthPlugin:Verify</string>
```

```
<string>loginwindow:done</string>
```

Download

github.com/tburgin/MacTech_2015


XCODE

Code:Snippets


SWIFT

Loading

Choose a template for your new project:

- iOS
  - Application
  - Framework & Library
  - Other
- OS X
  - Application
  - Framework & Library
  - System Plug-in
  - Other
  - Cocoa-Python

Cocoa Framework

Library

Bundle

XPC Service

Bundle

This template builds a bundle that links against the Cocoa framework.

Cancel

Previous

Next

ction

f - Define a block as

as Variable - Save
ble to allow reuse or
rgument.

ne a typedef.

Loading

Choose options for your new project:

Product Name: VerifyAuthPlugin

Organization Name: Burgin Systems

Organization Identifier: com.burginsystems

Bundle Identifier: com.burginsystems.VerifyAuthPlugin

Bundle Extension: bundle

Cancel

Previous

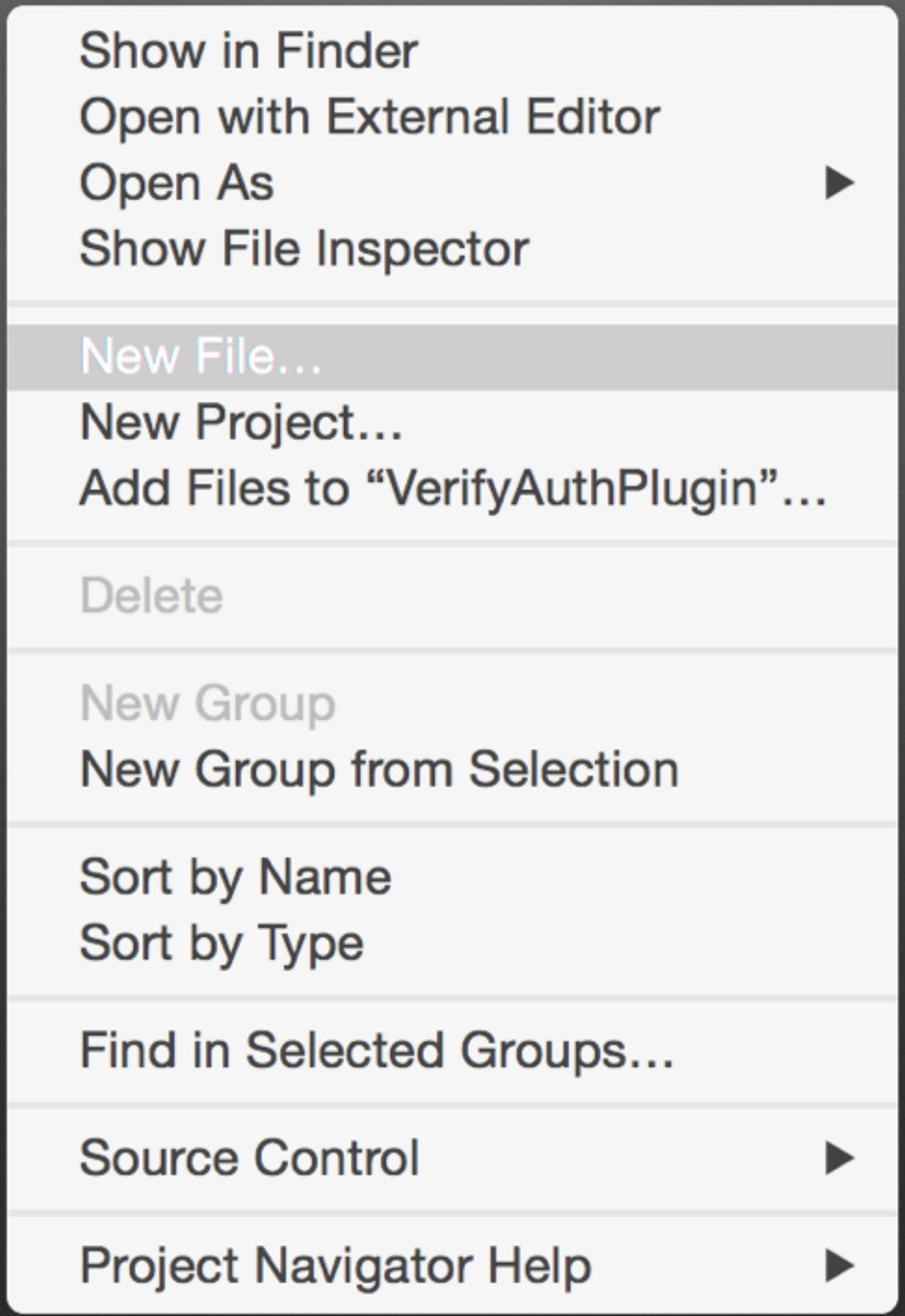
Next

ction

f - Define a block as

s Variable - Save
ble to allow reuse or
rgument.

ne a typedef.


Show in Finder
Open with External Editor
Open As
Show File Inspector
New File…
New Project…
Add Files to “VerifyAuthPlugin”…
Delete
New Group
New Group from Selection
Sort by Name
Sort by Type
Find in Selected Groups…
Source Control
Project Navigator Help

VerifyAuthPlugin: **Ready** | Today at 1:44 PM

Choose a template for your new file:

- iOS
  - Source
  - User Interface
  - Core Data
  - Resource
  - Other
  - Apple Watch
- OS X
  - Source
  - User Interface
  - Core Data
  - Resource
  - Other

Cocoa Class | Test Case Class | Playground | Swift File

Objective-C File | Header File | C File | C++ File

Cocoa Class

A Cocoa class.

Cancel | Previous | Next

VerifyAuthPlugin: **Ready** | Today at 1:44 PM

Choose options for your new file:

Class: AuthorizationPlugin

Subclass of: NSObject

Also create XIB file for user interface

Language: Objective-C

Cancel

Previous

Next

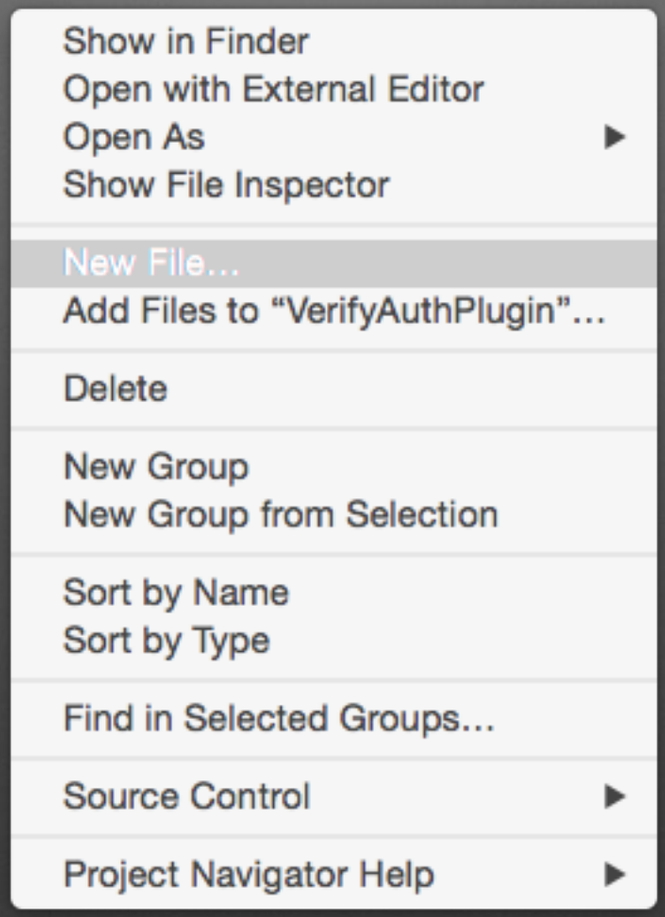
Code:AuthorizationPlugin

Core:Done

Let's make some decisions: Authorization Plugin Programming – Tom Burgin & Jeremy Baker

http://macadmins.psu.edu/conference/resources/

Mechanism:MachinePIN


Show in Finder
Open with External Editor
Open As
Show File Inspector
New File…
Add Files to “VerifyAuthPlugin”…
Delete
New Group
New Group from Selection
Sort by Name
Sort by Type
Find in Selected Groups…
Source Control
Project Navigator Help

VerifyAuthPlugin | Build VerifyAuthPlugin: **Succeeded** | Today at 11:53 AM

Choose a template for your new file:

- iOS
  - Source
  - User Interface
  - Core Data
  - Apple Watch
  - Resource
  - Other
- watchOS
  - Source
  - User Interface
  - Core Data
  - Resource
  - Other
- OS X
  - Source
  - User Interface
  - Core Data
  - Resource

Cocoa Class | UI Test Case Class | Unit Test Case Class | Playground

Swift File | Objective-C File | Header File | C File

C++ File | Metal File

Swift File

An empty Swift file.

Cancel | Previous | Next

VerifyAuthPlugin | Build VerifyAuthPlugin: **Succeeded** | Today at 11:53 AM

Save As: MachinePIN.swift

Tags:

Mechanisms

Search

Favorites

Devices

Shared

Tags

- Red
- Orange
- Yellow
- Green
- Blue
- Purple
- Gray
- All Tags...

Mechanisms
VerifyAuthPlugin
VerifyAuthPlugin.xcodeproj
Views

Group: Mechanisms

Targets: VerifyAuthPlugin

New Folder

Cancel

Create

VerifyAuthPlugin | Build VerifyAuthPlugin: **Succeeded** | Today at 11:53 AM

Would you like to configure an Objective-C bridging header?

Adding this file to VerifyAuthPlugin will create a mixed Swift and Objective-C target. Would you like Xcode to automatically configure a bridging header to enable classes to be accessed by both languages?

Cancel | Don't Create | Create Bridging Header

Choose a

- iOS
  - Sou
  - Use
  - Cor
  - Apple Watch
  - Resource
  - Other
- watchOS
  - Source
  - User Interface
  - Core Data
  - Resource
  - Other
- OS X
  - Source
  - User Interface
  - Core Data
  - Resource

Swift File | Objective-C File | Header File | C File

C++ File | Metal File

Swift File

An empty Swift file.

Cancel | Previous | Finish

VerifyAuthPlug
VerifyAuthPl
Mechanisms
Views
Products

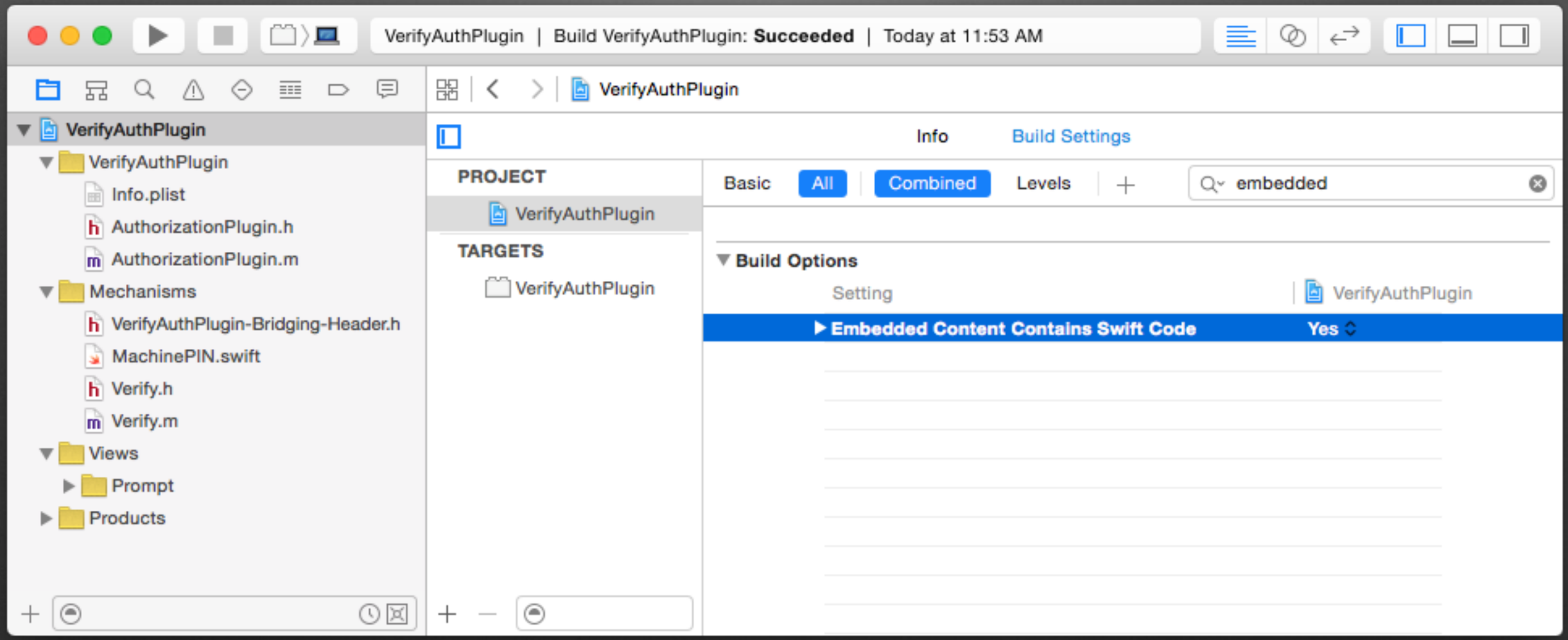
Group

itWithFrame:
e a custom view

lugin.h -
in Header

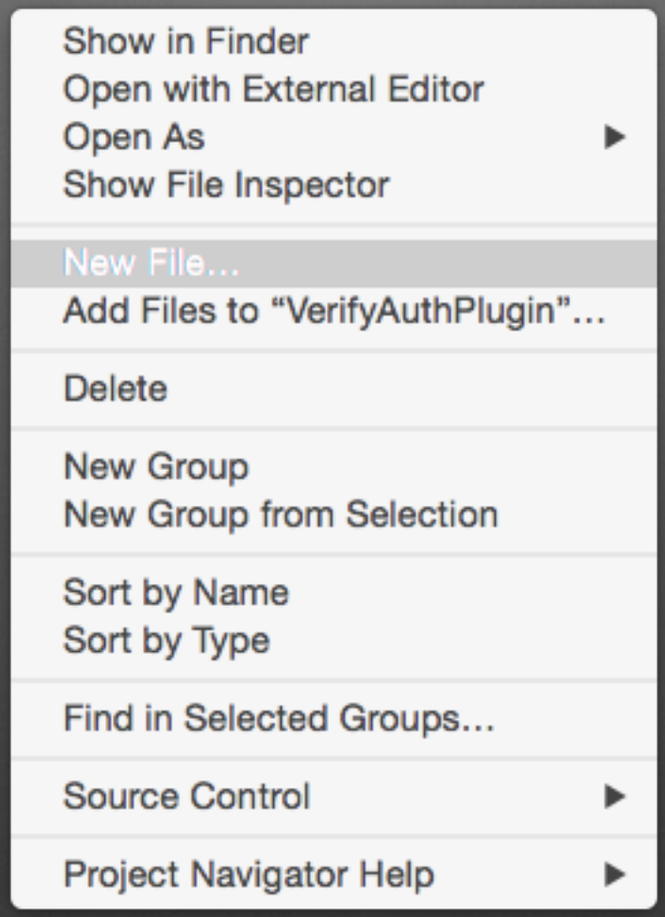
lugin.m -
in Implementation

```
//
// This is how we set the inter-mechanism context data
private func setHintValue(pin : NSString?) -> Bool {

    // Try and unwrap the optional NSString
    guard let pin = pin
        else {
            NSLog([+] Failed to unwrap inPin");
            return false
        }

    // Try and unwrap the optional NSData returned from archivedDataWithRootObject
    // This can be decoded on the other side with unarchiveObjectWithData
    guard let data : NSData = NSKeyedArchiver.archivedDataWithRootObject(pin)
        else {
            NSLog([+] Failed to unwrap archivedDataWithRootObject");
            return false
        }

    // Fill the AuthorizationValue struct with our data
    var value = AuthorizationValue(length: data.length,
        data: UnsafeMutablePointer<Void>(data.bytes))

    // Use the MechanismRecord SetHintValue callback to set the
    // inter-mechanism context data
    let err : OSStatus = self.mechanism.memory.fPlugin.memory.fCallbacks.memory.
        SetHintValue(mechanism.memory.fEngine, contextPINDomain.UTF8String, &value)

    return (err == errSecSuccess) ? true : false

}
```


VerifyAuthPlugin | Build VerifyAuthPlugin: Succeeded | Today at 11:53 AM
VerifyAuthPlugin
VerifyAuthPlugin
VerifyAuthPlugin
Info.plist
AuthorizationPlugin.h
AuthorizationPlugin.m
Mechanisms
VerifyAuthPlugin-Bridging-Header.h
MachinePIN.swift
Verify.h
Verify.m
Views
Prompt
Products
Info
Build Settings
PROJECT
VerifyAuthPlugin
TARGETS
VerifyAuthPlugin
Basic
All
Combined
Levels
embedded
Build Options
Setting
VerifyAuthPlugin
Embedded Content Contains Swift Code
Yes

Code:MachinePIN

Mechanism:Verify


Show in Finder
Open with External Editor
Open As
Show File Inspector
New File…
Add Files to “VerifyAuthPlugin”…
Delete
New Group
New Group from Selection
Sort by Name
Sort by Type
Find in Selected Groups…
Source Control
Project Navigator Help

VerifyAuthPlugin | Build VerifyAuthPlugin: **Succeeded** | Today at 11:53 AM

Choose a template for your new file:

- iOS
  - Source
  - User Interface
  - Core Data
  - Apple Watch
  - Resource
  - Other
- watchOS
  - Source
  - User Interface
  - Core Data
  - Resource
  - Other
- OS X
  - Source
  - User Interface
  - Core Data
  - Resource

Cocoa Class | UI Test Case Class | Unit Test Case Class | Playground

Swift File | Objective-C File | Header File | C File

C++ File | Metal File

Cocoa Class

A Cocoa class.

Cancel | Previous | Next

VerifyAuthPlugin | Build VerifyAuthPlugin: **Succeeded** | Today at 11:53 AM

Choose options for your new file:

Class: Verify

Subclass of: NSObject

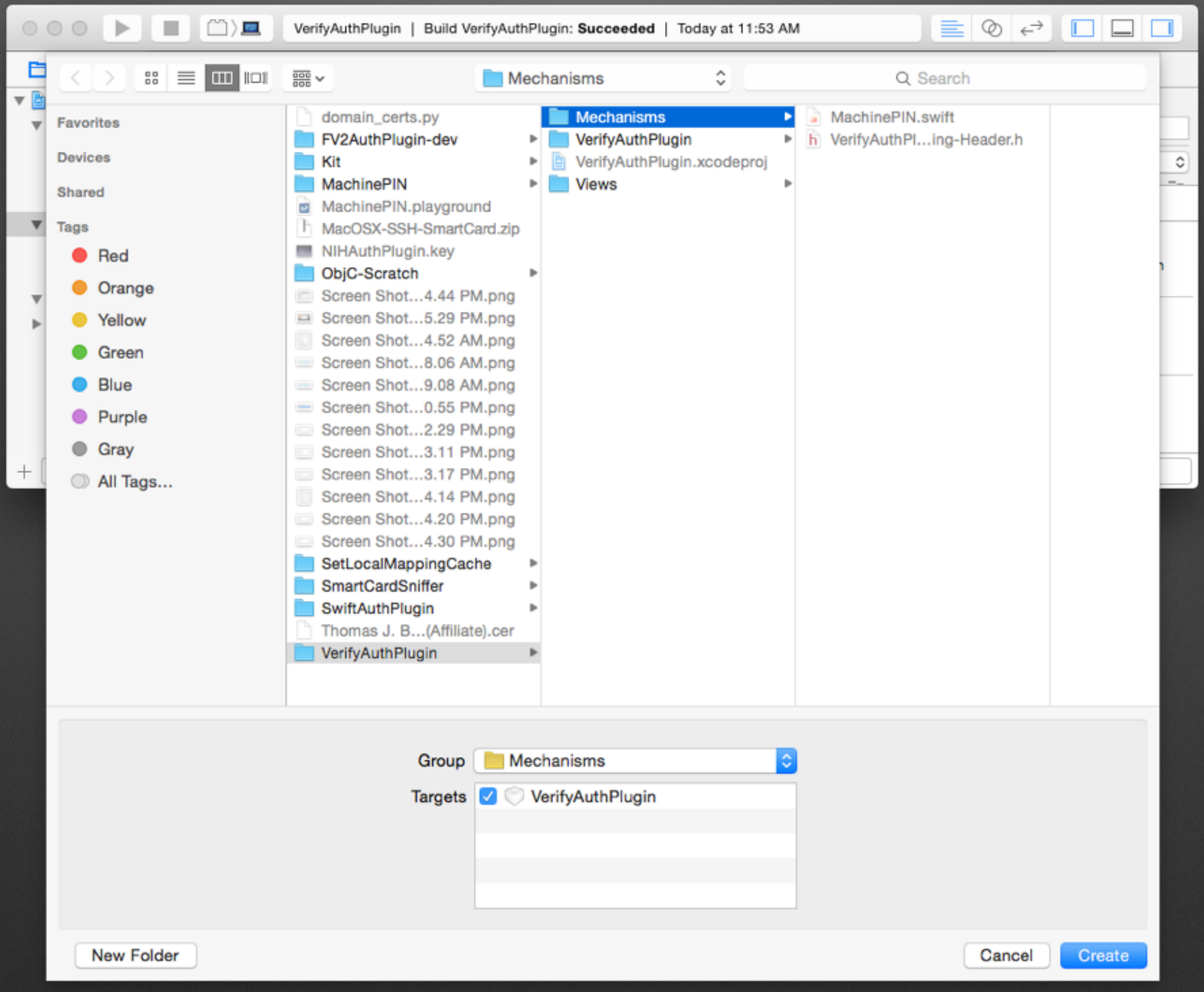
Also create XIB file for user interface

Language: Objective-C

Cancel

Previous

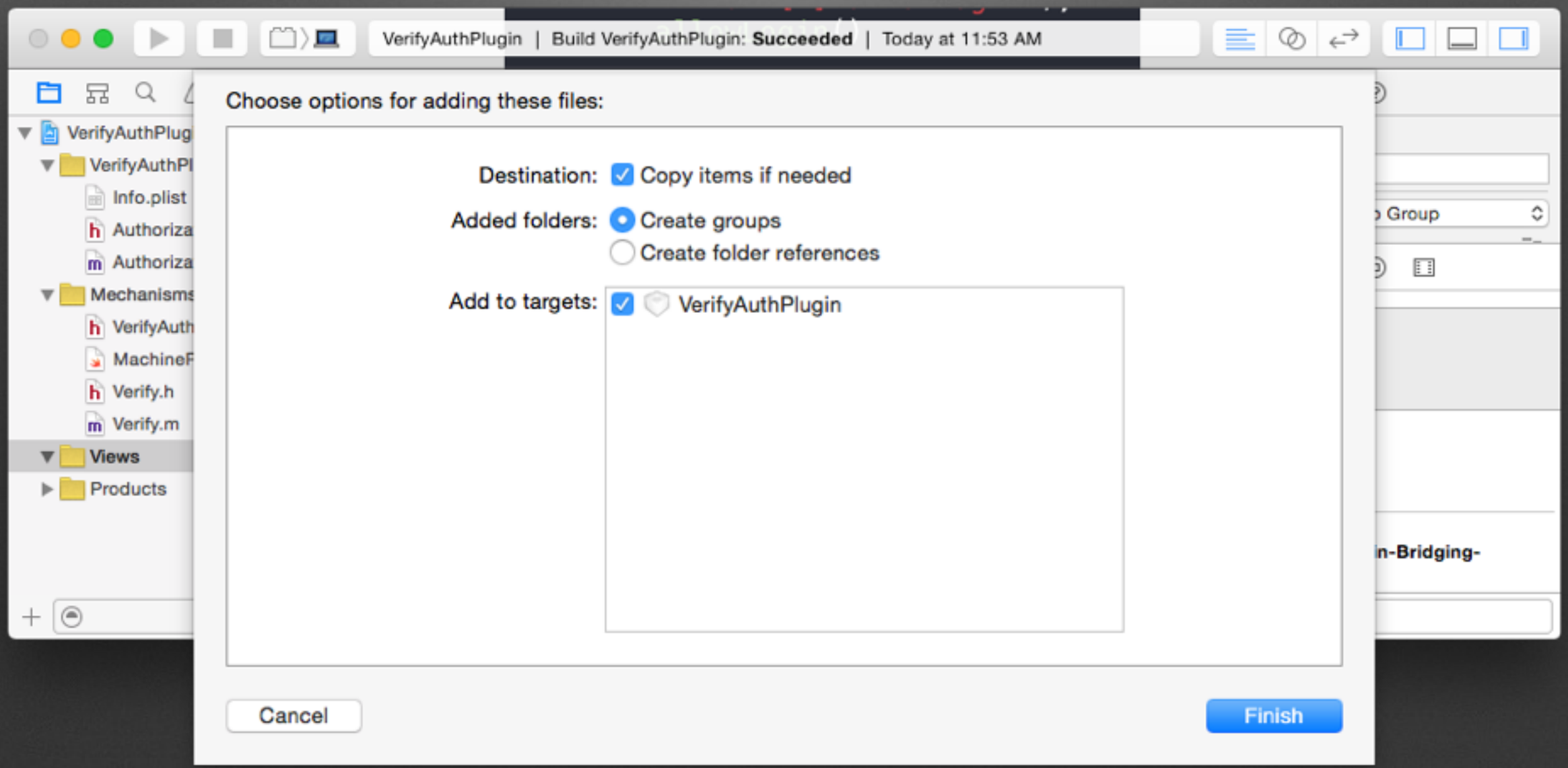
Next


VerifyAuthPlugin | Build VerifyAuthPlugin: Succeeded | Today at 11:53 AM
Mechanisms
Search
Favorites
Devices
Shared
Tags
Red
Orange
Yellow
Green
Blue
Purple
Gray
All Tags...
domain_certs.py
FV2AuthPlugin-dev
Kit
MachinePIN
MachinePIN.playground
MacOSX-SSH-SmartCard.zip
NIHAuthPlugin.key
ObjC-Scratch
Screen Shot...4.44 PM.png
Screen Shot...5.29 PM.png
Screen Shot...4.52 AM.png
Screen Shot...8.06 AM.png
Screen Shot...9.08 AM.png
Screen Shot...0.55 PM.png
Screen Shot...2.29 PM.png
Screen Shot...3.11 PM.png
Screen Shot...3.17 PM.png
Screen Shot...4.14 PM.png
Screen Shot...4.20 PM.png
Screen Shot...4.30 PM.png
SetLocalMappingCache
SmartCardSniffer
SwiftAuthPlugin
Thomas J. B...(Affiliate).cer
VerifyAuthPlugin
Mechanisms
VerifyAuthPlugin
VerifyAuthPlugin.xcodeproj
Views
MachinePIN.swift
VerifyAuthPl...ing-Header.h
Group
Mechanisms
Targets
VerifyAuthPlugin
New Folder
Cancel
Create

```objc
- (NSString *)getPIN {

    // Setup method variables
    NSString *pin;
    const AuthorizationValue *value;

    // This NSString will be used as the domain for the inter-mechanism context data
    NSString *contextPINDomain = @"com.burginsystems.pin";

    // Use the MechanismRecord GetHintValue callback to get the
    // inter-mechanism context data
    NSLog(@"[+] Attempting to read %@", contextPINDomain);
    if (_mechanism->fPlugin->fCallbacks->GetHintValue(_mechanism->fEngine,
                                                      [contextPINDomain UTF8String],
                                                      &value) == errAuthorizationSuccess) {

        NSData *pinData = [[NSData alloc] initWithBytes:value->data length:value->length];
        id ret = [NSKeyedUnarchiver unarchiveObjectWithData:pinData];
        pin = (NSString *)ret;

    } else {
        NSLog(@"[!] Failed to read %@", contextPINDomain);
    }

    return pin;
}
```

Code:Verify

```
Add:theView
```


VerifyAuthPlugin | Build VerifyAuthPlugin: Succeeded | Today at 11:53 AM
Choose options for adding these files:
Destination:
Copy items if needed
Added folders:
Create groups
Create folder references
Add to targets:
VerifyAuthPlugin
Info.plist
Mechanisms
Verify.h
Verify.m
Views
Products
Group
Cancel
Finish

VerifyAuthPlugin 〉 My Mac

VerifyAuthPlugin | Build VerifyAuthPlugin: **Succeeded** | Today at 11:53 AM

VerifyAuthPlugin 〉 Views 〉 Prompt 〉 PromptWindowController.xib 〉 Prompt Window

- VerifyAuthPlugin
  - VerifyAuthPlugin
    - Info.plist
    - AuthorizationPlugin.h
    - AuthorizationPlugin.m
  - Mechanisms
    - VerifyAuthPlugin-Bridging-Header.h
    - MachinePIN.swift
    - Verify.h
    - Verify.m
  - Views
    - Prompt
      - BorderlessWindow.h
      - BorderlessWindow.m
      - Keychain_Locked@2x.png
      - PromptWindowController.h
      - PromptWindowController.m
      - PromptWindowController.xib
      - shallnotpass.mp3
      - you_shall_not_pass.png
  - Products

- Placeholders
  - File's Owner
  - First Responder
  - Application
- Prompt Window
- Grey Background
- Prompt View
- Stop Login View

Window

Title: Title

Autosave: Autosave Name

Appearance: Title Bar, Unified Title And Toolbar, Full Size Content View, Shadow, Textured

Controls: Close, Minimize, Resize, Toolbar Button

Behavior: Restorable, Visible At Launch, Hide On Deactivate, Release When Closed, Always Display Tooltips, Recalculates View Loop

Spaces: Inferred Behavior

Exposé: Inferred Behavior

Cycling: Inferred Behavior

Full Screen: Inferred Behavior

Tiling: Inferred Behavior

C Block typedef - Define a block as a type.

C Inline Block as Variable - Save a block to a variable to allow reuse or passing it as an argument.

C typedef - Define a typedef.

VerifyAuthPlugin:Goals

<done>Create 2 Mechanisms</done>
<done>Share Data between Mechanisms</done>
<done>Prompt user for PIN</done>
<done>Make an authorization decision</done>

Plug::in

Verify.m

```
#import “PromptWindowController.h”

&&

// Create an instance of the PromptWindowController
```

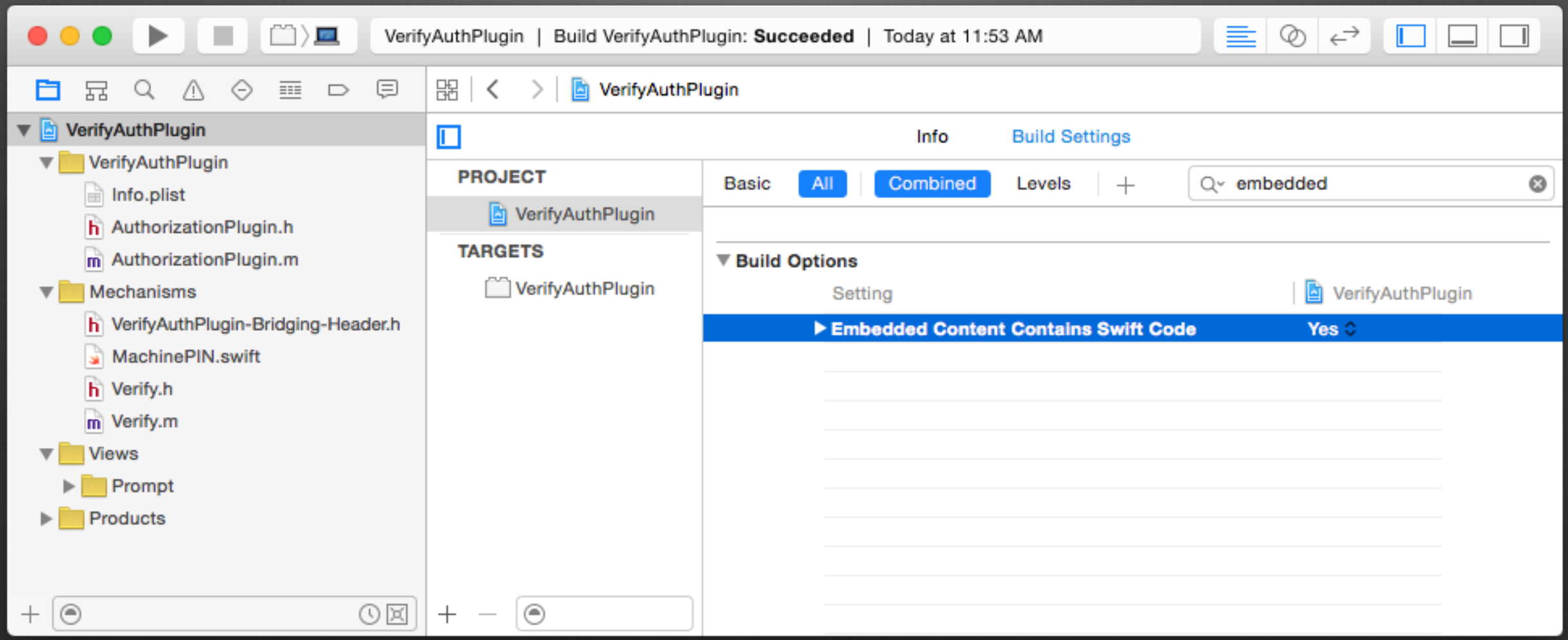
Code

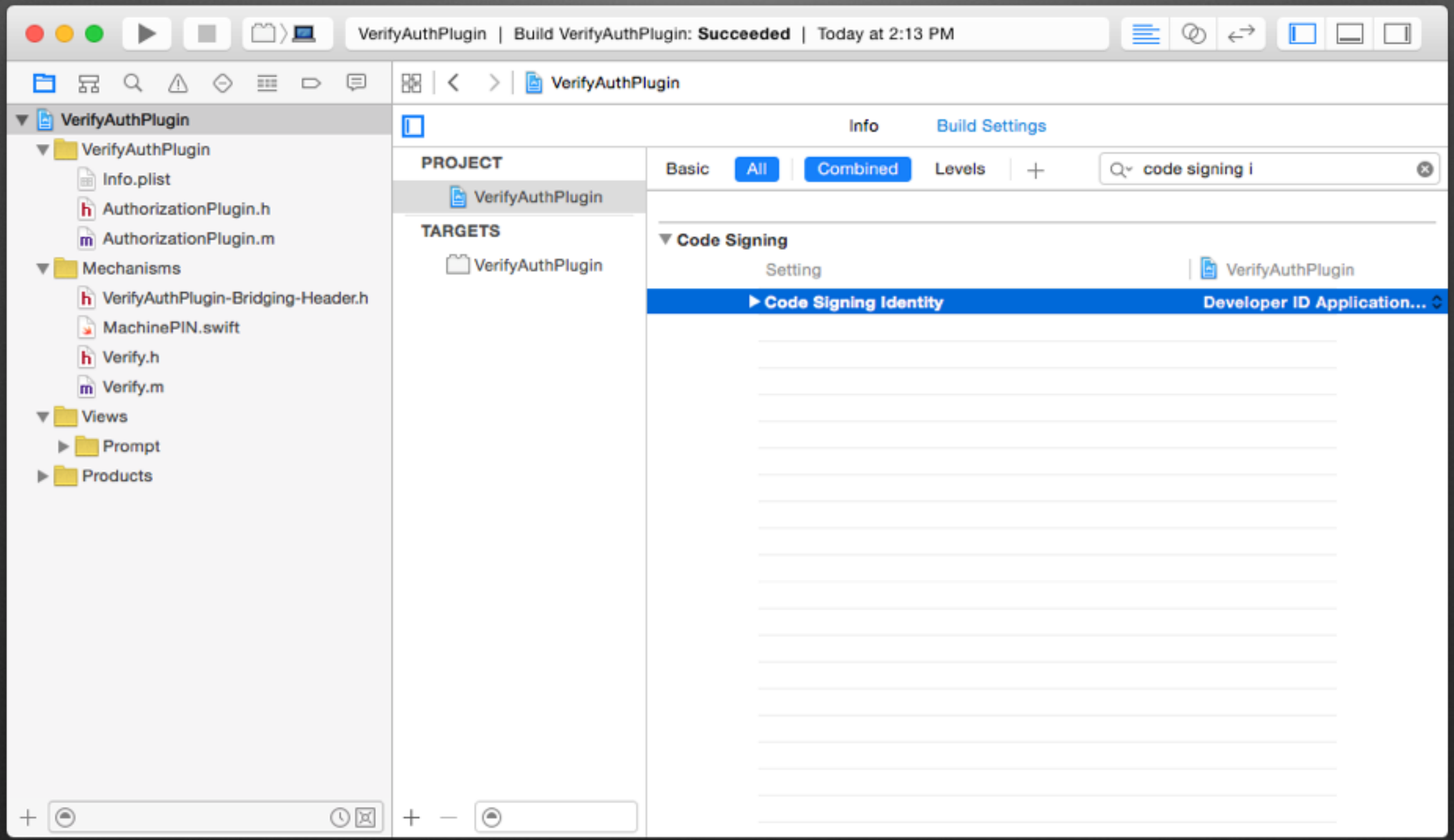
AuthorizationPlugin.m

```
// Special auto-generated header. It makes the Swift
                classes available to ObjC
            #import “VerifyAuthPlugin-Swift.h"

// Gains access to the ObjC Verify Mech
            #import “Verify.h"

// Call the MachinePIN mechanism
// Call the Verify mechanism
```

Code

Don't Forget!


VerifyAuthPlugin | Build VerifyAuthPlugin: Succeeded | Today at 11:53 AM
VerifyAuthPlugin
VerifyAuthPlugin
VerifyAuthPlugin
Info.plist
AuthorizationPlugin.h
AuthorizationPlugin.m
Mechanisms
VerifyAuthPlugin-Bridging-Header.h
MachinePIN.swift
Verify.h
Verify.m
Views
Prompt
Products
Info
Build Settings
PROJECT
VerifyAuthPlugin
TARGETS
VerifyAuthPlugin
Basic
All
Combined
Levels
embedded
Build Options
Setting
VerifyAuthPlugin
Embedded Content Contains Swift Code
Yes


VerifyAuthPlugin | Build VerifyAuthPlugin: Succeeded | Today at 2:13 PM
VerifyAuthPlugin
VerifyAuthPlugin
VerifyAuthPlugin
Info.plist
AuthorizationPlugin.h
AuthorizationPlugin.m
Mechanisms
VerifyAuthPlugin-Bridging-Header.h
MachinePIN.swift
Verify.h
Verify.m
Views
Prompt
Products
Info
Build Settings
PROJECT
VerifyAuthPlugin
TARGETS
VerifyAuthPlugin
Basic
All
Combined
Levels
code signing i
Code Signing
Setting
VerifyAuthPlugin
Code Signing Identity
Developer ID Application...


Build Succeeded

Install:Test

```
<step01>Copy:bundle</step01>
<step02>Add:authorizationdb</step02>
<step03>Keychain:com.burginsystems.pin</step03>
```

Lessons:Learned

- Privileged and non-privileged mechanisms run on different processes.
  - Use hints to share data between address spaces. Use objects to share data between like mechanisms.
- Check for null terminated strings within the context and hint values.
  - Different session callers fill the data inconsistently.