



Oren Novotny

- Chief Architect, DevOps & Modern Software at BlueMetal
- Microsoft Regional Director
- Microsoft MVP
- VS ALM Ranger

- Blog: [//oren.codes](http://oren.codes)
- Twitter: [@onovotny](https://twitter.com/onovotny)
- GitHub: github.com/onovotny



Agenda

- Introduce Code Signing & Authenticode
- Key Vault's HSM and certificate support
- Automated signing with CI/CD

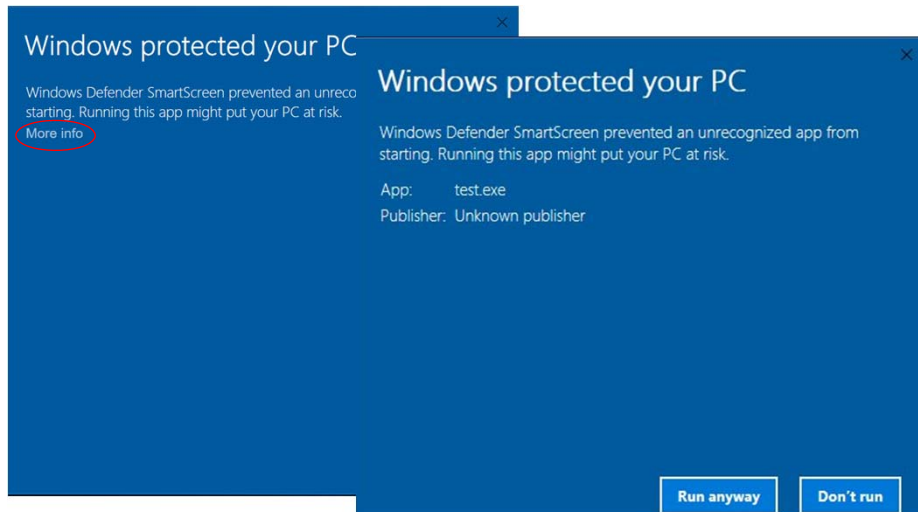


Survey says ...

- Create applications
 - For the public?
 - For internal use?
- Create libraries
 - For the public?
 - For internal use?



Why sign your code?



Why sign your code?

- Prove integrity
- Develop reputation
- Reputation is earned, malware can be signed
- Non-repudiation



Code signing reasons

- Better install experience
- Enterprise AV may flag unsigned files
- “It’s the rules” – “The higher-ups said so”
- Workstation lockdown – “only run code signed by these publishers”
- Prove origin – “That’s the real one”
- Drivers – OS requires it
- NuGet
- Reduce Incident Response time

“Companies that don’t are unprofessional and don’t care about the integrity of their delivered product.” - @SwiftOnSecurity



Part of defense in depth

- Reputation based
 - SmartScreen
 - NuGet – verified publisher information
- Restricted environments
 - Device Guard
 - AppLocker
 - Example: Privileged Access Workstations (bit.ly/ms-paw)



DEMO: SIGNING CODE

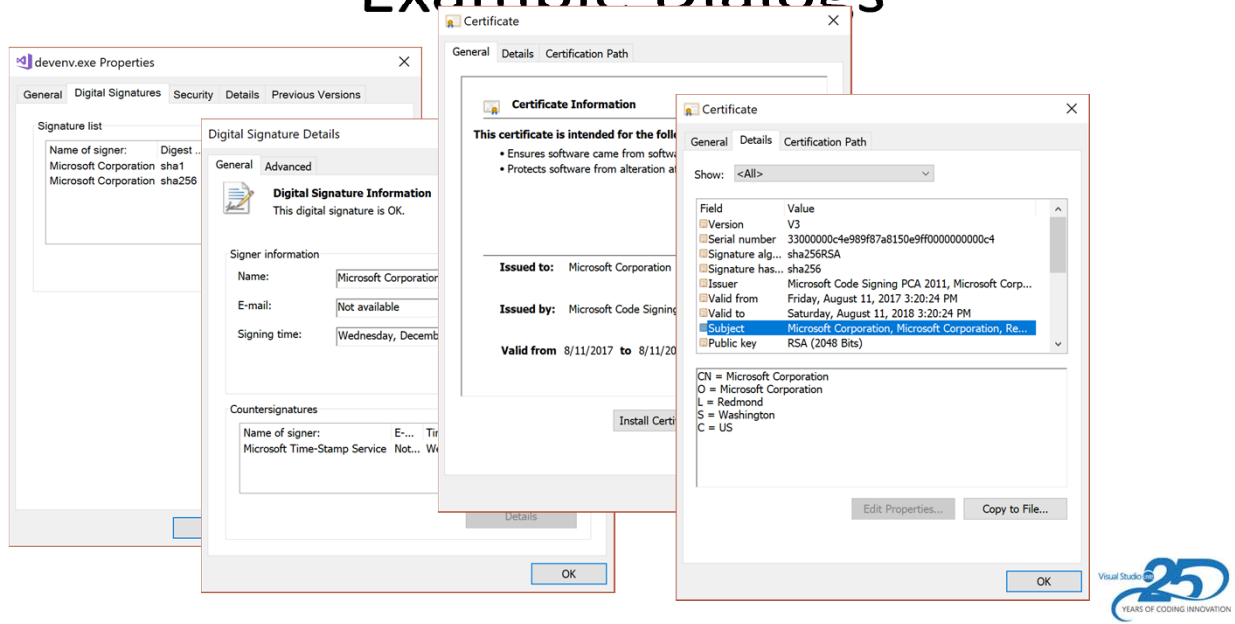


Authenticode

- >20 years old
- Initially for ActiveX controls downloaded web (remember when that was a thing?)
- Ensures authenticity & integrity
- Subsequent versions added Timestamp support
- Based on existing standards (PKCS#7 & ASN.1)
- Relies on PKI – CA's to verify identity
- Nothing to do with Strong Naming



Example Dialogs



Digital Certificates

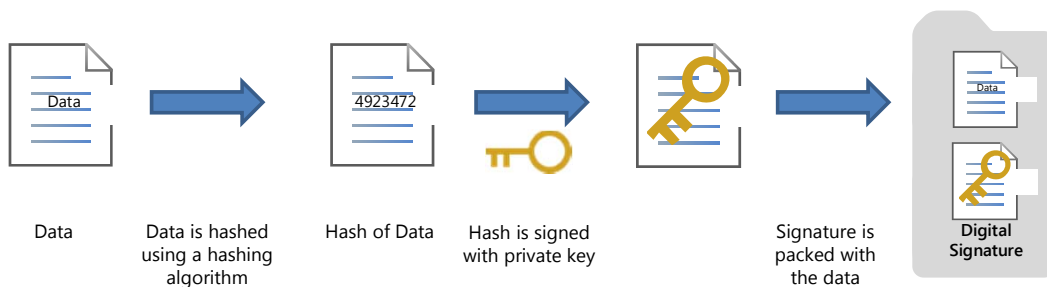
- X509 certificate and private key
- Includes identity information in public portion, signed by issuer
 - Chain of trust
- Obtained from public CA
 - DigiCert
 - GlobalSign
 - GoDaddy
- Corporations often have internal CA's
- Self-signed (for testing only)

Major certificate types

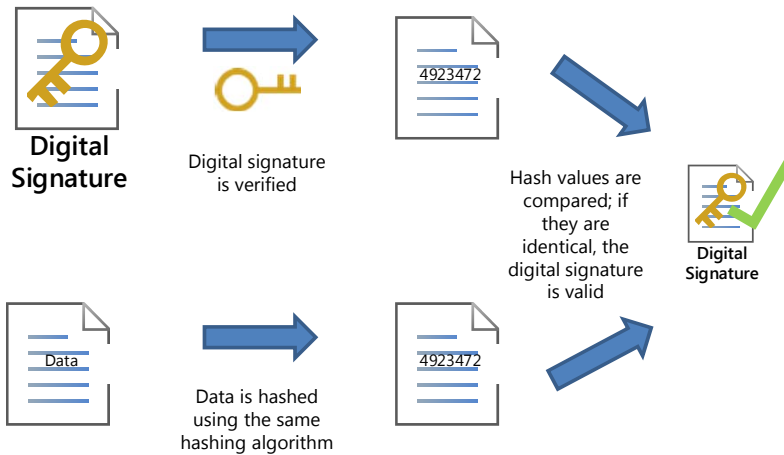
- Server (Authentication, TLS) / EV
- Code Signing / EV
- Document Signing
- Client (Authentication, S/MIME)
- RSA vs. Elliptic Curve
- Requirements for each set by
 - CA/B Forum
 - Microsoft Trusted Root Certificate Program
 - Adobe (for PDFs)
 - Google Root Certificate Program
- EKUs – what kind of certificate it is



Digital signatures - creating



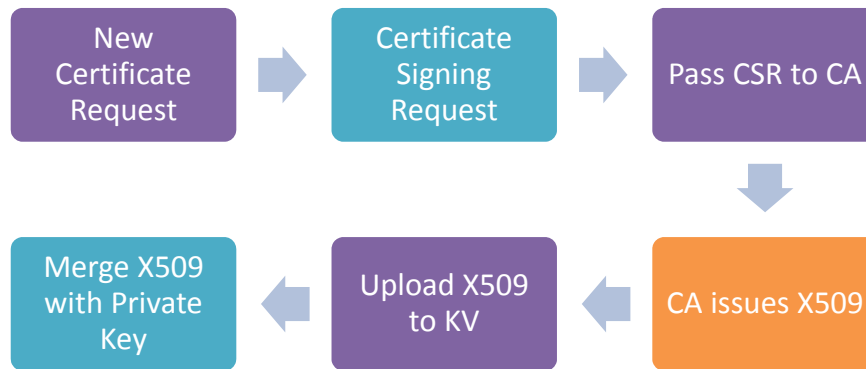
Digital signatures - verifying



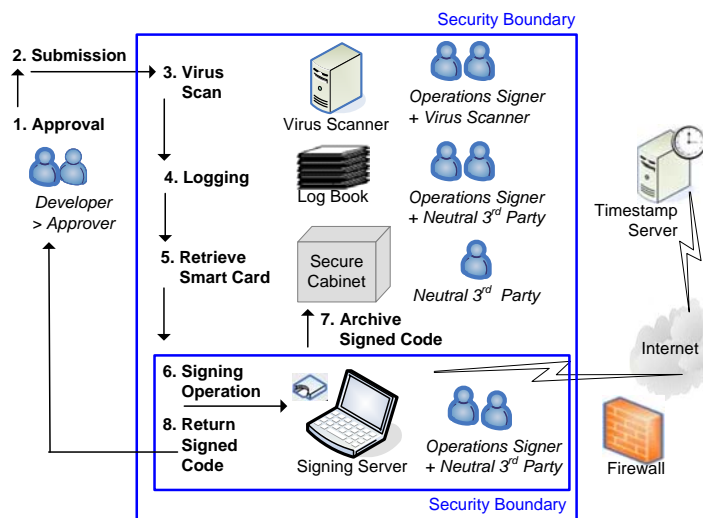
HOW IT WORKS



Getting a certificate



Offline code signing



Signing Tools

- Different file types have different tools
 - Authenticode: SignTool.exe (exe, dll, ps1, appx, msix, sys, msi)
 - VSIX packages: VsixSignTool.exe (vsix)
 - ClickOnce / VSTO: Mage.exe (vsto, application)
 - NuPkg: NuGet.exe (nupkg)
 - Java: Jarsign, Apksigner (jar, apk)
- None of them have direct support for external signing - Key Vault
- Rely on CSP (operating system) support



CODE SIGNING CHALLENGES



Protecting the certificate

- Protecting the key is hard
- Most important – three strikes and you’re out
- Developers should not have direct access to it – auditing
- Manifests need to match final certificate for UWP
 - Painful to “do it right”
- EV requires HSM
- The built-in tooling doesn’t extend – needs HSM drivers
 - USB tokens often can only be used interactively, not remote or as a service



Arcane commands

- Hard to remember commands
- Not as simple as “sign this thing”

```
signtool sign /tr http://timestamp.digicert.com /td sha256 /fd sha256  
/a Setup.exe
```

```
signtool sign /tr http://timestamp.digicert.com /td sha256 /fd sha256  
/sha1 [thumbprint] /itos file.exe
```

```
signtool sign /tr http://timestamp.digicert.com /td sha256 /fd sha256  
/sha1 [thumbprint] /tseal file.exe
```

```
signtool sign /tr http://timestamp.digicert.com /td sha256 /fd sha256  
/f "c:\path\to\mycert.pfx" /p pfxpassword "c:\path\to\file.exe"
```

- Once you figure it out for SignTool, now you need VSIX, Jar, NuGet, etc

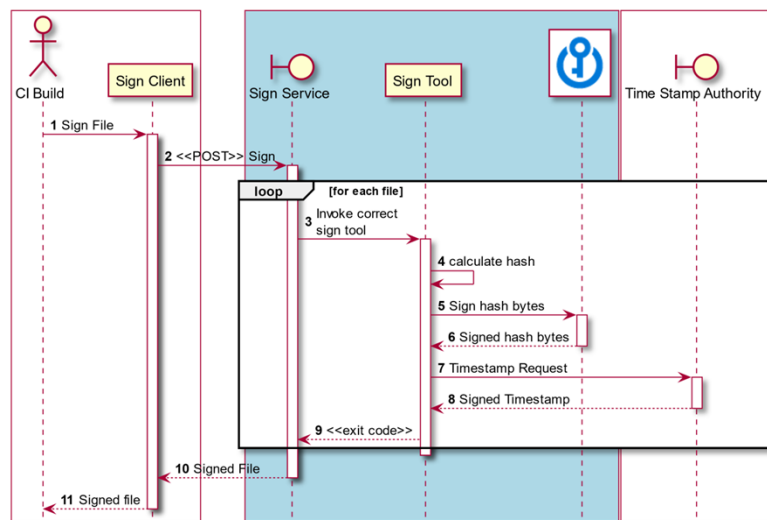


Orchestrating multiple tools

- VSIX contains DLL's and a NuPkg, what do you do?
- VSTO contains DLL's?
- AppX manifest publisher name must match cert.
- Need to sign the “inner” files first, update manifests, sign outer files
- Gets tedious and error-prone



Solution: CI code signing



SIGNING SERVICE

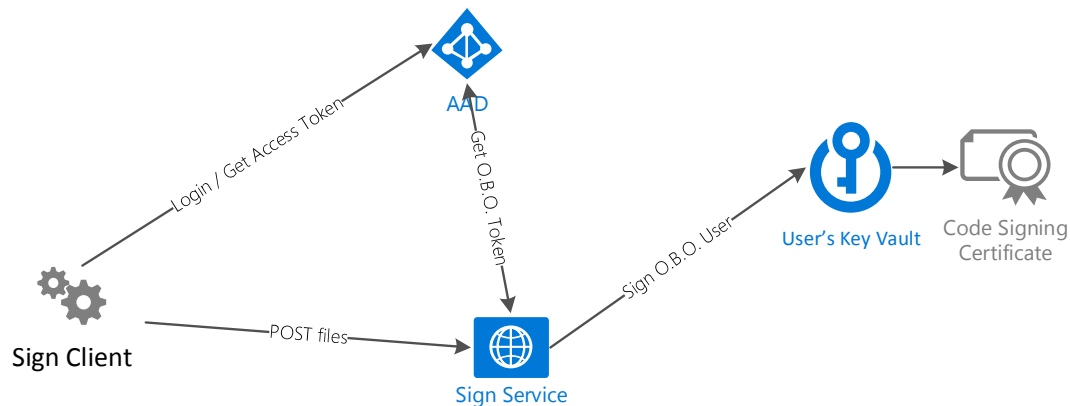


Goals

- Securely sign files during CI builds
- Multi-certificate capable – some organizations use cert per group
- Built for Azure. Uses Azure AD, Key Vault, and Websites
- Handles nested files, no need to manually orchestrate
- Open Source. Audit the code, contribute back if you want
- Inexpensive to operate. Compare to the cost of on-prem HSM's



Signing Service Architecture



Security Model

- Signing operations in Key Vault restricted to User + Application
 - Each user gets their own vault – ACL's at Vault level
- Service itself only has management rights, not crypto rights
- User configuration as AAD custom attributes on user object
 - No extra database
- Sign Client gets access token to Service
- Service uses OBO flow to get token to user's configured vault



Supporting Key Vault

- Kevin Jones created OpenVsixSignTool and AzureSignTool
 - Open source implementations of closed source tools
 - AzureSignTool relies on SignerSignEx3, on Win10/Server 2016
- Mage – adapted signed xml code
- NuGet – custom signing implementation using API
- Jar Sign – coming soon



Success Stories - .NET Foundation

"Code signing can be complicated, and it's not something you want to guess at. Oren has distilled a deep understanding of how to sign code correctly into a service that makes it easy to do it right."



Previously, the .NET Foundation had a cumbersome process relying on virtual machines with signing keys installed, batch files, and luck. Oren's signing service allows us to cleanly integrate code signing directly into our automated builds with keys stored in Key Vault. Very highly recommended!"

- Jon Galloway, Executive Director



Success Stories – Cake Build



“Oren has with his Sign Service & Client transitioned the task of signing code from frustrating and complex, to almost trivial. This while still remaining secure and compliant - which is an impressive achievement!”

The amount of friction and yak shaving needed has been significantly reduced, to the level where we no longer got a valid excuse, to not sign all the things!” – Mattias Karlsson, Maintainer



CERTIFICATES WITH AZURE KEY VAULT



Azure Key Vault

- Two editions, standard and premium
- Premium uses FIPS 140-2 Level 2 validated HSMs
- Protected by Azure AD
- Logging and usage auditing
- Holds keys, secrets, and certificates
- RSA-HSM keys can never leave hardware boundary
- Multiple layers of redundancy
 - Replicas within region
 - Secondary region >150mi away



Azure Key Vault - Usage

- Management of certificates, keys, and secrets
 - Create Certificate, Get public cert, list certificates, etc
- Cryptographic
 - Sign, Decrypt
- REST APIs, use from anywhere
- C# clients
- Many libraries and infra support – Managed Service Identities



Azure

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

SETTINGS

Keys

Secrets

Certificates

+ Add

Refresh

Certificate Contacts

NAME	THUMBPRINT	STA
Oren-Novotny	6FA13D52D40823B7...	✓
TestCert	1C162977CF0E81994...	✓

Create a certificate

Method of Certificate Creation
Generate

* Certificate Name

Type of Certificate Authority (CA)
Certificate issued by a non-integrated CA

* Subject
For example: "CN=mydomain.com".

DNS Names
0 DNS names

Validity Period (in months)
12

Content Type
PKCS #12 PEM


Lifetime Action Type
E-mail all contacts at a given percentage lifetime

Percentage Lifetime
80

Advanced Policy Configuration
Not configured

☐ Pin to dashboard

Create



Configuration Requirements

- Azure Subscription and Global Admin of the backing AAD
 - Can use an a dedicated directory/subscription
- Recommended: VSTS CI/CD
- Resources created:
 - App Service Plan
 - App Service (Website), with site extensions
 - Application Insights
 - Key Vault for runtime config

Demo: Getting Started

- Clone the repo <https://github.com/onovotny/SignService>
- Run the InstallUtility to create Azure AD configuration & resources
- Run ARM template to create resources
- Build / Deploy App Service
- Admin UI
 - Create service accounts, creates dedicated Key Vault
 - Create CSR or import certificate
 - Specify parameters in user profile
- Client build script with protected/encrypted variables
- Add a script to the build to download/invoke client



//oren.codes
@onovotny

THANK YOU!

