DBsign and the DoD Java JRE STIG

Version 4.0

Copyright Notice:

Table of Contents

1.0 Introduction

On March 27th, 2013, DoD issued the "Java Runtime Environment 6 STIG Version 1" and the "Java Runtime Environment 7 STIG Version 1"[1] to restrict DoD users from running signed applets which were signed by code signing certificates that are not already trusted by the JRE.

This document explains signed applet security and the security vulnerabilities that are addressed by this STIG. It also explains how the STIG works and its impact on the use of signed applets such as the DBsign Universal Web Signer (UWS) in browser based web applications.

1.1 About Signed Applets

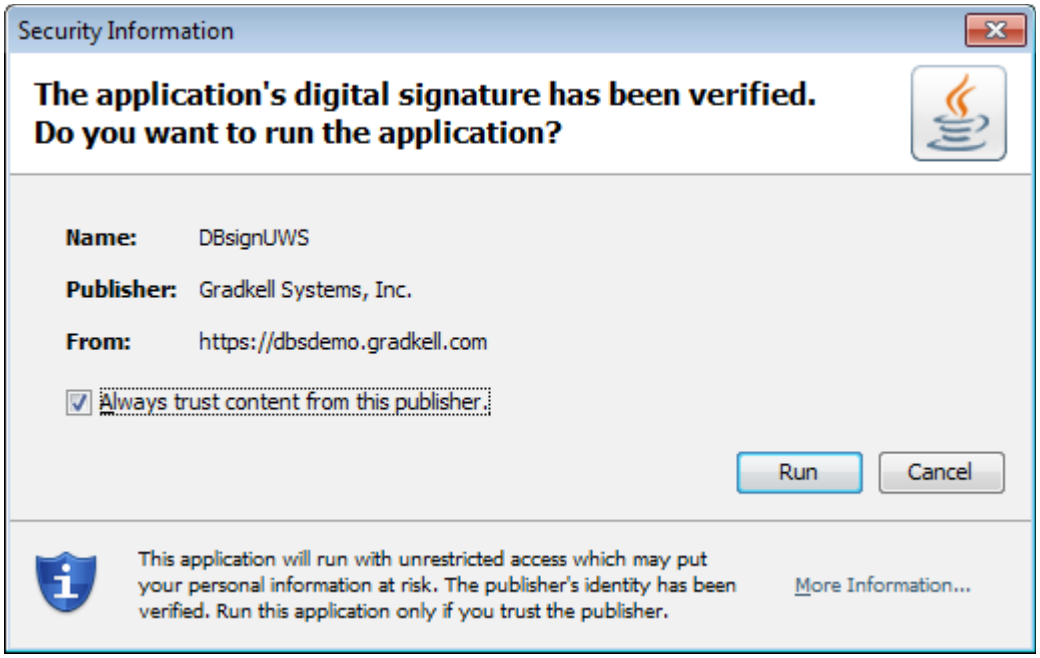
Although browsers can use cryptography and smart cards for SSL/HTTPS connections, they do not provide the ability to digitally sign documents or application data. Plug-ins, Active-X controls, add-ons, extensions, and applets are some of the mechanisms to extend the functionality of web browsers. DBsign uses Java applets to extend the browser because applets work in many browsers and operating systems and can handle both 32 bit and 64 bit environments.

By default, Java applets are not "trusted", so they execute in a "sandbox" where they cannot access data on the host computer. If an applet is digitally signed, then their origin is known and users can make a decision as to whether or not they trust the publisher of that applet. If signed applets are trusted, then they do not run in the Java "sandbox" and can get access to the host computer. Since DBsign needs to interact with operating system certificate stores and smart cards, the DBsign Universal Web Signer applet is signed with Gradkell's code signing certificate. This allows users to know that the applet was published by Gradkell and has not been altered. With this knowledge a user can safely trust the DBsign UWS applet to execute on their system.

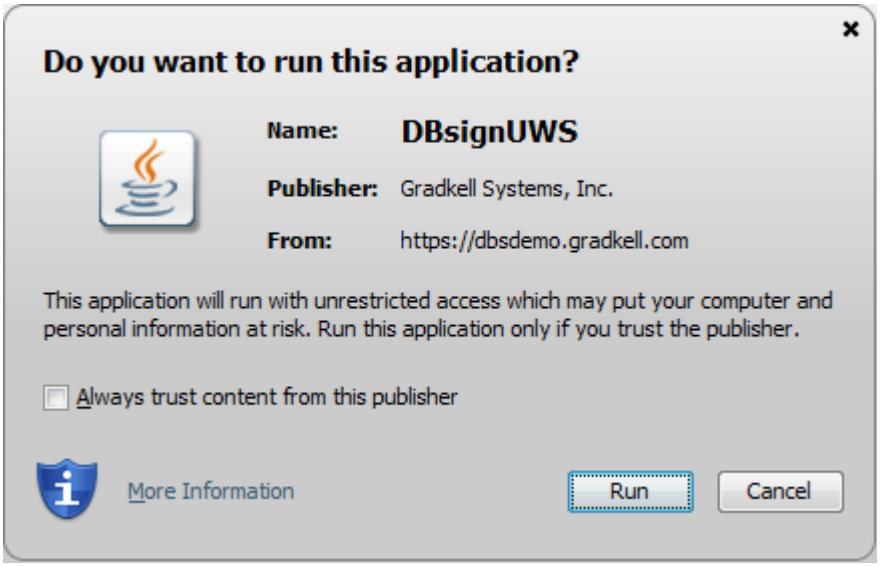
The Java Runtime Environment (JRE) maintains a list of trusted certificate authorities that issue code signing certificates to organizations like Gradkell. If an applet was published using a trusted

1 STIG is available at http://iase.disa.mil/stigs/app_security/app_sec/app_sec.html#JRE

code signing certificate, the JRE will indicate to the user that the publisher's identity has been successfully validated. This, however, does not necessarily mean that the user should trust the publisher. Because of this, the JRE will prompt the user for permission to execute the signed applet with a dialog such as one of the following.[2]



2 Note that on the Java 6 the “Always trust content from this publisher” option is selected by default and not Java 7 it is not.



If the publisher's identity could not be validated, the default behavior or the JRE is to notify the user of this fact and to prompt the user as to whether or not to execute the signed applet anyway.

1.2 Security Vulnerabilities Addressed by this STIG

Many times users do not read security warning prompts like those shown above and will just click “Run” in an attempt to make the system work. This can lead to users making bad trust decisions and trusting a signed applet from a malicious publisher whose identity cannot be validated.

To protect DoD systems from this vulnerability, the Java JRE STIG disables the ability of the user to execute signed applets from a publisher whose identity cannot be validated.

2.0 How the STIG Works

The STIG effects two files in the JRE's lib directory: deployment.config and deployment.properties. These files control control some of the security settings of the JRE and prevent end users from changing those settings in the Java Control Panel. These settings and apply to all users of the JRE on the computer.

The deployment.config file simply tells the JRE where to find the deployment.properties file and what to do if it doesn't exist. The STIG specifies that the deployment.properties should be in the JRE's lib directory and that the JRE should still function normally if the deployment.properties file is not there. On a 64-bit Windows 7 system, the deployment.config would look something like the following.

```
deployment.system.config=file:C\:\\Program
Files\\Java\\jre7\\lib\\deployment.properties
deployment.system.config.mandatory=false
```

The deployment.properties file actually contains the settings. It looks something like the following.

```
deployment.security.askgrantdialog.notinca=false
deployment.security.askgrantdialog.notinca.locked
deployment.security.validation.crl=true
deployment.security.validation.crl.locked
deployment.security.validation.ocsp=true
deployment.security.validation.ocsp.locked
```

There are three settings specified and each of them are “locked”, which means that the end user of the workstation cannot change them in the Java Control Panel. Each of the settings will be explained.

2.1 Protection Against Publishers Whose Identity Cannot Be Validated

The `deployment.security.askgrantdialog.notinca` setting prevents the JRE from prompting the user about whether or not to trust a publisher whose code signing certificate could not be validated. That is, it was not issued by a certificate authority (CA) that is trusted by the

JRE. If the publishers signing certificate cannot be validated, then the applet with not be executed. The end user will only be prompted to trust applets signed by publishers whose identity is verifiable.

The `deployment.security.askgrantdialog.notinca.locked` setting indicates that the user should not be allowed to change this setting the in the Java Control Panel.

2.2 Protection Against Publishers With Revoked Certificates

Since a Certificate Authority can revoke a publishers code signing certificate, the DoD has mandated that the revocation status of the publishers certificate should be checked. This can be performed in two ways: Certificate Revocation Lists (CRLs) and the Online Certificate Status Protocol (OCSP). Both of these methods require access to the Internet. Since Gradkell's code signing certificate was issued by Verisign, these revocation status checking methods require access to `verisign.com`.

The `deployment.security.validation.crl=true` setting tells the JRE to attempt to perform revocation status checking using CRLs. When this is enabled the JRE will attempt to download a CRL from a URL that it specified inside the publishers code signing certificate. In the DBsign UWS 4.0.7.0 the URL is `http://csc3-2010-crl.verisign.com/CSC3-2010.crl`.

The `deployment.security.validation.crl.locked` setting indicates that the user should not be allowed to change this setting the in the Java Control Panel.

The `deployment.security.validation.ocsp=true` setting tells the JRE to attempt to perform revocation status checking using OCSP. When this is enabled the JRE will attempt to access an OCSP responder at a URL that it specified inside the publishers code signing certificate. In the DBsign UWS 4.0.7.0 the URL is `http://ocsp.verisign.com/`.

The `deployment.security.validation.ocsp.locked` setting indicates that the user should not be allowed to change this setting the in the Java Control Panel.

3.0 Possible Side Effects Of The STIG

Since the STIG requires that the publishers code signing certificate be both

- issued by a CA certificate that is trusted by the JRE, and
- proven to be not revoked via a successful revocation status check (using either CRLs or OCSP),

the JRE will not allow the signed applet to be executed outside the sandbox unless both these conditions are satisfied. When this STIG is applied, if either of these conditions are not satisfied, the JRE will present a dialog that looks something like the following, but the text at the bottom may be different depending on the cause of the validation failure.

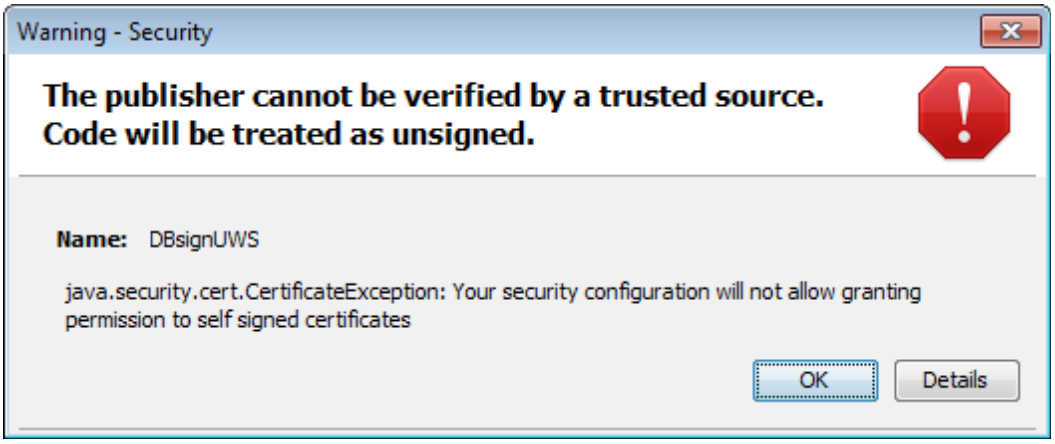


Figure 3: JRE Security Warning Dialog After STIG

The dialog indicates the the applet will not execute as a signed applet, but will execute in the Java sandbox. Some applets are signed just to prove the origin of the applet. However, if the signed applet does indeed require operating system services not available within the Java sandbox, the applet will not run properly.

3.1 Applets from Untrusted CAs

Some signed applets may not be published using certificates that are trusted by default in the JRE. This would include signed applets published under DoD certificates because the DoD root certificate is

not in the JRE's default trust list. These applets will not be able to run on JREs with default trust lists. New CAs will have to be added to the JRE's trust list.

3.2 Clients Without Network Access to CRLs or OCSP

In some DoD environments, access to DoD servers is permitted, but access to the Internet is blocked. In these environments, users will not be able to execute many signed applets (including the DBsign UWS) because the required CRL or OCSP revocation status checks cannot be performed.

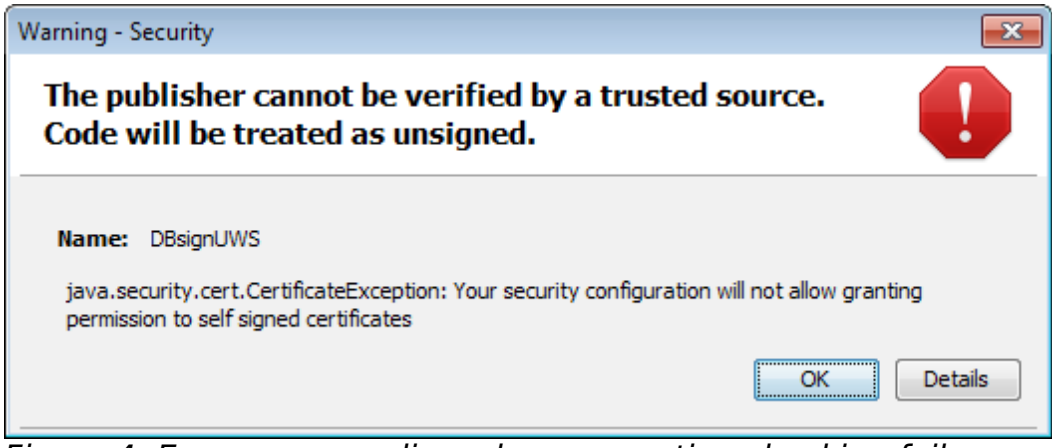


Figure 4: Erroneous wording when revocation checking fails.

The dialog in Figure 4 was produced by applying the STIG to a Windows 7 machine running Java 1.6 and disallowing access to the Verisign OCSP responser and the Verisign CRL URL. The error message indicates that there was a problem related to self-signed certificates, however the DBsign UWS is not signed by a self-signed certificate. The problem was really that the JRE could not access Verisign's CRLs or OSCP responders. Although the signed applet should not have been allowed to run outside the sandbox, and this dialog should have been presented, we consider the text regarding self-signed certificates to be erroneous.

4.0 Conclusions

The DoD JRE STIG is a good security patch that will mitigate a legitimate vulnerability (i.e., users potentially trusting rouge publishers of signed applets). However, there are side effects of the revocation status checking settings that prevent users from executing signed applets in environments where Internet access is blocked or unreliable.