AUTODESK **UNIVERSITY** 2014

Security Best Practices for CAD Managers and Developers

Davis Augustine – Autodesk, Inc.

Code SD6746

We will present effective security practices for AutoCAD software developers, CAD managers, and advanced AutoCAD users. Straight from members of the AutoCAD Security Team. Concepts include types of vulnerabilities and threats, existing AutoCAD software security features, and recommended system settings and administration. For native C++ development, we'll also briefly cover secure build switches, APIs and testing strategies.

Learning Objectives

At the end of this class, you will be able to:

- Understand various kinds of malware threats against AutoCAD
- Configure AutoCAD and Windows to defend against threats
- Use safe practices in deploying apps
- Develop plugins and extensions in a secure way

About the Speaker

Davis has been in various AutoCAD software development teams for several years, focusing recently on security.

Note: Please also see the Presentation pptx file and the slides' notes for more info.

Who are the hackers and what can they do?

Recreational hackers, students, vandals, copycats.
These groups may put up annoying messages, damage data files or the system. They may have crudely copied known exploits, either reliably or not. Most known AutoCAD attacks fall into this category.

Extortionists, Organized Crime, Nation States
These are more serious operators, who are out for financial gain and high value targets. They may steal data, ransom it, destroy it, damage devices connected to the system or launch attacks against other systems.

File "Planting" Exploits

Automatically executed files
AutoCAD automatically loads various executable files at startup time and at drawing open time. Acad.lsp, express tools, pgp files, acad.rx and cuix files are loaded at startup. Acaddoc.lsp is loaded at drawing open time.

Executable files may be planted among data files
Data sets provided from external sources may be unzipped into a local download folder, and then those drawing files opened from explorer. If there are executable files hidden within the data files, those executables may get loaded instead of the intended ones from the support path. This is the classic pattern for an AutoCAD lisp virus, the first of which was "Bursted" back in the early 2000's.

TrustedPaths Sysvar protects against this
This recently added sysvar causes a warning to be put up if there is an attempt to load an executable from a folder outside of the trusted paths. You should not put your data folders (where downloaded dwg files reside) on the trusted paths, if you can avoid it.

Future Acad versions may stop searching the drawing and start-in folders for executables
This provides another level of safety, as data and executable files will be kept separate and it will be less easy for attackers to sneak executable files in among their data sets.

Mistakenly Trusted Apps and Digital Signing

Users run apps and plug-ins which they shouldn't

Naïve users don't know any better and will run malware if their systems are not configured to prevent it. Even sophisticated users can be tricked into running malware, if the attackers are patient and determined enough. But that would only happen in the case of a high value target.

CAD Managers can require users to be standard user, and to enable UAC.

This protects the system and other users from changes made by malware mistakenly run by the user. But it doesn't protect the user's own data. UAC is very effective even if the user has admin rights, as long as the user doesn't accept things that they shouldn't when the UAC dialog pops up.

Group Policy controls can be used to prevent standard users from installing software

So only admins can install software. Except that some software installs without using the Windows installer, and we can't prevent that. This other software would not be able to install to Program Files, however, and would not be able to damage non-user data.

Digital Signatures identify a file's publisher and certificate authority

If the signature is valid, then you know that the file is intact from the publisher, but you still have to decide whether you trust the publisher and the certificate authority. AutoCAD apps may currently be signed, in which case they will load even if not on the trusted paths. In the future, acad may warn about signed apps found outside trusted paths, and allow the user to say that they want to always trust apps provided by that publisher (so don't show the warning again).

Native Code Injection is a powerful exploit

Carefully crafted data files carry malware payloads

Ingenious programming is used by attackers to take advantage of bugs in native code, causing control to jump to unexpected addresses and execute hostile code, or to execute existing code in unexpected order. Once the malware has control of the process, it can attack the system in various ways. The bugs taken advantage of typically involve unexpectedly long input that overflows buffers. This doesn't happen in managed code because buffer overruns are detected.

DEP and ASLR are common ways to defend against buffer overruns

Data Execution Prevention is a hardware supported mode which disallows executing of code inside data segments. This makes it more difficult for buffer overruns to inject code. DEP can be disabled in Windows and on some machines in the BIOS, so users and admins should make sure it is enabled.

Address Space Layout Randomization causes modules to load at random addresses, making it harder for attackers to construct injected code. This is enabled at build (link) time by the app developers.

Other possible future Acad Security changes

Drawing Passwords

Password encrypted dwg files have been supported since acad 2004, but the encryption is not very powerful. It can be cracked using modern technology in days if not hours. We may want to drop this functionality altogether.

Lockable Sysvars

Currently, acad keeps sysvar values under HKCU (current user) in the registry. This allows standard users to update their sysvars. In the future, acad may allow sysvar values to also be placed in HKLM (local admin) in the registry, with these values overriding the ones in HKCU. That would allow administrators to lock values for some sysvars.

Opening dwg files directly from zip files

ETransmit and other zip files often come from the internet and thus may be considered unsafe. Rather than unzipping the contents into a local folder, it may be safer to leave the dwg files and other support files in the zip file, and open them directly from there. One problem with that though is that acad may not want to support writing back to the zip file. So it would be a readonly/publishing mode of operation.

Protected Sandbox mode

Future acad versions may want to open drawings coming from the internet in a protected “low integrity” (or sandbox) mode, to isolate any malware within the dwg files. This would be similar to IE, Chrome and Office’s sandboxing modes of operation.

Security Practices for Native C++ Developers

Enable DEP, ASLR, Buffer Overflow Detection and other safe modes

The first three are enabled by default, so you just need to make sure not to disable them. The MSVC /SDL switch goes a little farther than these in its runtime checking and also does things like pointer clearing after delete. It has some performance impact, but it's minimal.

Digitally sign all app modules

This will help guarantee reliability and accountability.

Use Secure APIs

All APIs which take a destination buffer arg, such as strcpy, wcscpy, memmove, etc, should also take a destination buffer size arg. Don't use or create APIs which don't fit this pattern. Use banned.h to find and fix calls to the unsafe APIs.

External tools like BinScope, Static Analyzers, Fuzz Testers

Binscope is a free tool from Microsoft which finds unsafe build issues in apps. Static analyzers like Fortify and ClokWork search source code for unsafe or erroneous usages. Fuzz testers tweak input files and send them to your app, in order to find failures to properly validate input.

Conclusion

Security is an ongoing effort.

The exploits and defenses are constantly evolving and becoming more sophisticated. The danger is not going away, and both CAD Managers and Developers need to keep up with the best practices and apply them.