Exploiting Windows Access Control Model for Local Elevation of Privilege

This TOPIC will teach you about Windows Access Control and how to find instances of misconfigured access control exploitable for local privilege escalation.

Why Access Control Is Interesting to a Hacker

Access control is about the science of protecting things. Finding vulnerabilities in poorly implemented access control is fun because it feels like what security is all about. It isn't blindly sending huge, long strings into small buffers or performing millions of iterations of brute-force fuzzing to stumble across a crazy edge case not handled properly; neither is it tricking Internet Explorer into loading an object not built to be loaded in a browser. Exploiting access control vulnerabilities is more about elegantly probing, investigating, and then exploiting the single bit in the entire system that was coded incorrectly and then compromising the whole system because of that one tiny mistake.

How Windows Access Control Works

It's important to first understand how Windows Access Control works. This introductory section is large because access control is such a rich topic. But if you stick with it and fully understand each part of this, it will pay off with a deep understanding of this greatly misunderstood topic, allowing you to find more and more elaborate vulnerabilities.

This section will be a walkthrough of the four key foundational components you'll need to understand to attack Windows Access Control: the security identifier (SID), the access token, the security descriptor (SD), and the access check

Tools for Analyzing Access Control Configurations

With the concept introduction out of the way, we're getting closer to the fun stuff. Before we can get to the attacks, however, we must build up an arsenal of tools capable of dumping access tokens and security descriptors. As usual, there's more than one way to do each task. All the enumeration we've shown in the figures so far was done with free tools downloadable from the Internet. We'll demonstrate each tool we used earlier, show you where to get them, and show you how to use them.

Dumping the Process Token

The two easiest ways to dump the access token of a process or thread are Process Explorer and the! token debugger command. Process Explorer ...

Special SIDs, Special Access, and “Access Denied”

You also are armed with tools to enumerate the access control objects that factor into AccessCheck. It’s time now to start talking about the “gotchas” of access control and then start into the attack patterns.

Special SIDs

You’ve seen the user SID several times. You’ve seen the SID of the Administrators and Users groups and how the presence of those SIDs in the token changes the privileges present and the access granted. You’ve seen the Local System SID. Let’s discuss several other SIDs that might trip ...

Analyzing Access Control for Elevation of Privilege

All the file read access discussion earlier was to help you understand concepts. The attack methodology and attack process are basically the same no matter the resource type.

- Step 1: Enumerate the object’s DACL and look for access granted to non-admin SIDs.
  We look for non-admin SIDs because attacks that require privileged access to pull off are not worth enumerating. Group those non-admin SIDs in the DACL into untrusted and semi-trusted users. Untrusted users are Users, Guest, Everyone, Anonymous, INTERACTIVE, and so on. Semi-trusted users are interesting in the case of a multistage attack. Semi-trusted ...

Attack Patterns for Each Interesting Object Type

Let’s apply the analysis methodology to real objects and start finding real security vulnerabilities. The following sections will list DACL enumeration techniques, then the power permissions, and then will demonstrate an attack.

Attacking Services

Services are the simplest object type to demonstrate privilege escalation, so we’ll start here. Let’s step through our attack process.

Enumerating DACL[**Discretionary Access Control List**] of a Windows Service

We’ll start with the first running service on a typical Windows XP SP2 system.

```
C:\tools>net start

These Windows services are started:
```

```
Alerter

Application Layer Gateway Service

Ati HotKey Poller

Automatic Updates

...
```

We used AccessChk.exe earlier to enumerate file system DACLs.

What Other Object Types Are out There?

Services, registry keys, files, and directories are the big four object types that will expose code execution vulnerabilities. However, several more object types might be poorly ACL'd. Nothing is going to be as easy and shellcode-free as the objects listed. The remaining object types will expose code execution vulnerabilities but you'll probably need to write "real" exploits to leverage those vulnerabilities. Having said that, let's briefly talk through how to enumerate each one.