How the Mimikatz Hacker Tool Stole the World's Passwords

By Andy Greenberg

Published: 2017-11-09 · Archived: 2026-04-05 20:26:01 UTC

Five years ago, Benjamin Delpy walked into his room at the President Hotel in Moscow, and found a man dressed in a dark suit with his hands on Delpy's laptop.

Just a few minutes earlier, the then 25-year-old French programmer had made a quick trip to the front desk to complain about the room's internet connection. He had arrived two days ahead of a talk he was scheduled to give at a nearby security conference and found that there was no Wi-Fi, and the ethernet jack wasn't working. Downstairs, one of the hotel's staff insisted he wait while a technician was sent up to fix it. Delpy refused, and went back to wait in the room instead.

When he returned, as Delpy tells it, he was shocked to find the stranger standing at the room's desk, a small black rollerboard suitcase by his side, his fingers hurriedly retracting from Delpy's keyboard. The laptop still showed a locked Windows login screen.

The man mumbled an apology in English about his keycard working on the wrong room, brushed past Delpy, and was out the door before Delpy could even react. "It was all very strange for me," Delpy says today. "Like being in a spy film."

It didn't take Delpy long to guess why his laptop had been the target of a literal black bag job. It contained the subject of his presentation at the Moscow conference, an early version of a program he'd written called Mimikatz. That subtly powerful hacking tool was designed to siphon a Windows user's password out of the ephemeral murk of a computer's memory, so that it could be used to gain repeated access to that computer, or to any others that victim's account could access on the same network. The Russians, like hackers around the world, wanted Delpy's source code.

In the years since, Delpy has released that code to the public, and Mimikatz has become a ubiquitous tool in all manner of hacker penetrations, allowing intruders to quickly leapfrog from one connected machine on a network to the next as soon as they gain an initial foothold.

Benjamin Delpy

Most recently, it came into the spotlight as a component of two ransomware worms that have torn through Ukraine and spread across Europe, Russia, and the US: Both NotPetya and last month's BadRabbit ransomware strains paired Mimikatz with leaked NSA hacking tools to create automated attacks whose infections rapidly saturated networks, with disastrous results. NotPetya alone led to the paralysis of thousands of computers at companies like Maersk, Merck, and FedEx, and is believed to have caused well over a billion dollars in damages.

Those internet-shaking ripples were enabled, at least in part, by a program that Delpy coded on a lark. An IT manager for a French government institution that he declines to name, Delpy says he originally built Mimikatz as a side project, to learn more about Windows security and the C programming language—and to prove to Microsoft that Windows included a serious security flaw in its handling of passwords.

His proof-of-concept achieved its intended effect: In more recent versions of Windows, the company changed its authentication system to make Mimikatz-like attacks significantly more difficult. But not before Delpy's tool had entered the arsenal of every resourceful hacker on the planet.

"Mimikatz wasn’t at all designed for attackers. But it's helped them," Delpy says in his understated and French-tinged English. "When you create something like this for good, you know it can be used by the bad side too."

Even today, despite Microsoft's attempted fixes, Mimikatz remains an all-too-useful hacker tool, says Jake Williams, a penetration tester and founder of security firm Rendition Infosec. "When I read a threat intelligence report that says someone used Mimikatz, I say, 'tell me about one that doesn’t,'" Williams says. "Everyone uses it, because it works."

Secrets for the Taking

Mimikatz first became a key hacker asset thanks to its ability to exploit an obscure Windows function called WDigest. That feature is designed to make it more convenient for corporate and government Windows users to prove their identity to different applications on their network or on the web; it holds their authentication credentials in memory and automatically reuses them, so they only have to enter their username and password once.

While Windows keeps that copy of the user's password encrypted, it also keeps a copy of the secret key to decrypt it handy in memory, too. "It’s like storing a password-protected secret in an email with the password in the same email," Delpy says.
