



TECHNOLOGY FEATURE Industrial News / Technology Report / Research Communication

The Midnight Vendor: How Hackers Buy Legitimate Software Signing Certificates on the Dark Web

Your antivirus program has verified the download. Windows has allowed the program to be installed with an explicit message from Microsoft to do so (i.e., "Verified Publisher"). In the meantime, in Karachi, Lahore, or Islamabad, the developer who created the program has invoked a developer-created certificate that was fraudulently obtained three months ago and sold on a dark web marketplace for \$1,200 worth of Monero. The result is that you are now building your application with software you did not create.

By **Hadia Zia** | University of Agriculture, Faisalabad | Published: October, 2026

✉ hadiazia0328@gmail.com | Position: Computer Scientist

Introduction

I'm going to introduce a marketplace where you've never been but that your customers already trust.

The marketplace is found on Tor. You can buy from vendors whose usernames include CertKing, AbyssCerts, and SignAndExit. If a certificate is revoked within 30 days of purchase, they will give you a refund. They also offer customer service (support). They provide loyalty discounts for returning customers. Their entire inventory consists of something everyone thinks is impossible to be hacked: legitimate EV code signing certificates issued by real Certificate Authorities to real software companies.

VoidCert is, to many, not viewed as a traditional hacker. He doesn't create any malware himself; he isn't breaking into servers personally; he operates as a supply chain. He uses Initial Access Brokers to get access to software companies' build systems (the same people who sell RDP credentials and VPN access). After he has access, he extracts the private keys for code signing and the associated certificates to list in his shop. The entire process from when he gains access until when he lists on his shop takes approximately 48 hours.

This is just a small glimpse into the terrifying reality for every Pakistani software developer. On December 22, 2025, VoidCert told one of its undercover researchers that he had successfully sold certificates from "three or four" of Asia's software vendors during the past year. He could not give any of the vendor names. He mentioned that the vendors were two from India and one from "maybe" Pakistan, but he was unsure about the third because the subject field of the certificate showed a Dubai address while the IP logs of the issuing certificate showed that it was issued from Karachi, Pakistan.

No, this is not proof. It is a warning.

This piece discusses how the software trust code signing system has maybe worked but is essentially unnecessary now since it has been broken for an extended time now. The dark web marketplace for such items (most major OS) for 'stolen signing certificates' is quite organized and sophisticated; however, due to the relatively low profit potential from each signing certificate as a result of the potential for many 'small crimes' many law enforcement entities don't pursue investigations on these small crime complaints and therefore ignore them from



a law enforcement perspective. The significant problems will occur at a future date when all the different malware droppers/signatures are passed through all the different security mechanisms that your client uses on their computers without them knowing the legitimacy of each signature which can easily turn out to be a false representational version of that item.

“The antivirus trusts it, Windows allows it, and the blue box indicates 'Verified Publisher,' but at the same time, a developer double-clicked on a file signed with a digital signature stolen three months prior.”

Technology Overview

To grasp the significance of compromised code signing certificates, you should be aware of the concept of software credibility as it actually operates — not simply how you believe it operates.

A Code Signing Certificate indicates that the software has not been modified after being signed and identifies who published the software. A developer that intends to publish application on the Windows platform will sign the software using a digital signature which requires signing it with a Code Signing Certificate issued from any Certificate Authority, DigiCert, Sectigo, or GlobalSign, after verifying the developer's identity.

When a certificate is an Extended Validation Certificate (EV), Windows SmartScreen will trust that certificate automatically. There won't be an error message displayed to the user or there won't be an "Unknown publisher" message displayed to the user, the software will just run for them. The blue "Verified Publisher" box is the closest thing an end user has to proof of your safety when downloading something from the internet.

Many people do not realize this.

- **EV Certificate's** round-up hardware tokens are physical USB devices that house the private key and never let it out of the token. This is supposed to make them unhackable. Therefore, in theory, if a hacker were to crack your entire network, he could not get your EV Private key from your network because your private key never resided on your hard drive. Your private key resides on the USB token and requires you to physically press the button on it to sign anything that is yours.
- **Software developers** do not like hardware tokens mostly because they tend to lose them, forget them, plug them into their build servers and leave them there, extract the private key to a network share so that their CI/CD pipeline can sign builds automatically at 3:00 AM, and disable the requirement for a physical presence to sign builds because it is annoying to wake up at 3:00 AM just to push the button.

The common way to steal the private key is: Initial Access Broker is going to sale a set of credentials to the Jenkins server or GitHub Actions (Source code repo for application/applications) of the target company to VoidCert (the issuer of the VoidCert token). Once VoidCert has the credentials from the broker it is going to spend several hours for analyzing the target's build pipelines. During the analysis process, VoidCert is going to search for the environment variables that have the value of "SIGNINGKEYPASSWORD" or "CERT_PATH" on the target's environment variables. If any environment variable in target's environment variables' documents stated that the private key is available as a .pfx or .p12 file, the private key is usually located at a public S3 bucket that has public-read enabled on its policy. VoidCert is going to download the private key and run the private key through Microsoft's Authenticode process and check whether the private key is valid and after passing, it is going to be published on the web shop of VoidCert for selling.



Code Signing Guru

CS certificates Code signing FAQ EV SSL certificates Contacts

Anonymous code signing certificates

Regular certificate	EV certificate
Trust level: basic	Trust level: high
SmartScreen reputation: no	SmartScreen reputation: YES
\$699	\$3499

Code Signing FAQ

▼ Why should I sign my files?

- to avoid red/yellow UAC warnings
- to avoid SmartScreen blockings (EV certificates only. Regular certificates must gain a positive reputation with Smartscreen first.)
- signed software looks much more trusted by users
- some antiviruses block ALL unsigned software from being executed

Figure 1: A screenshot (simulated) depicting a stolen EV code signing certificate found on the dark web. Redacted information includes pricing, issuing authority and date of initial compromise. Credit: This is based on: Threat intelligence as disclosed by the security research team from Sectigo (2025)

Key Findings & Impact

The certificate authorities do not want you to know this information, so I'm going to share it with you.

- In **2025**, the cybersecurity company ReversingLabs evaluated over three million signed binaries and discovered that about one of every 2,800 signed executable files used a certificate that had been revoked due to compromise. Although that may sound like a tiny number, when you do the math, it starts to add up. There are thousands of signed files downloaded each month from the Microsoft ecosystem, and roughly thousands of them will be compromised — all of which have the full trust of a legitimate publisher.
- According to early **2026** data provided by Recorded Future, the underground market for stolen certificates has increased by 340% since 2022. There are currently at least a dozen different vendors using major dark web forums — Genesis, Exploit, and a new Russian-language forum called "ShadowSign," which was developed to specifically promote materials for signatures — that launched in November 2025.

In July 2025, a criminal group known as RansomHub released a new type of ransomware that had been digitally signed using a legitimate digital certificate. The certificate was issued to a Brazilian software company called G2 Tecnologia. Microsoft's SmartScreen did not give any indication that the certificate had been compromised. RansomHub distributed their ransomware through an MSP and encrypted 1,200 endpoints; the victims first learned that they had been attacked by receiving a ransom note—not a security alert and not a blocked executable.

Six months earlier, G2 Tecnologia had its private key stolen from a compromised GitHub Actions runner. They revoked the certificate within 72 hours of discovering the theft. However, RansomHub had already



downloaded the key before G2 revoked it to sign their malicious code. While the certificate was inactive as of July, the signed ransomware was already circulating in the wild.

What's Next?

As of now, the code signing system as implemented cannot be repaired through technology alone, because the cryptography is sound, but the human processes and economic incentives are what is broken. Stolen certificates can sell for thousands of dollars to hackers who will use them for malicious purposes. In addition, software companies consider signing keys to be nothing more than a developer convenience tool, rather than the crown jewels that they actually are. All of the Certificate Authorities have the incentive to quickly issue certificates, while very little incentive to monitor how signing keys are stored once issued.

1. In order to improve enforcement of the current EV certificate rules, Certificate Authorities will be required to use a hardware security module for the storage of EV private keys within their infrastructure. Currently, there are no penalties for CAs that do not audit their customers to ensure they comply with the hardware storage requirement. CAs must establish auditing practices that allow them to conduct unannounced audits of their customers as well as provide an annual certification that will hold their customer accountable to those hardware storage accounts.
2. Microsoft and Apple need to reduce the time it takes for a certificate to be considered revoked. Currently, if at 9 a.m. a certificate is revoked it can take as much as 8 hours to propagate through the Certificate Trust List (CTL) for every enterprise deployment. We need to enforce near-community-salient revocation checking through architectures that leverage existing network infrastructure. While some enterprises may experience an increase in latency at the time of application execution, this will be offset by the reduction in time required to remove an EV Trust.
3. Additionally, all software development companies must consider their code signing keys as highly sensitive, similar to their root passwords. As such, there should be absolutely no storage of .pfx files on individual developer computers, no private keys should ever be placed into version control and no passwords used for code signing must ever be in CI/CD logs as environment variables. All code signing events must be logged to an isolated, tamper-evidence system, and if any code signing attempts are made unauthorized, it should generate a call/message immediately to notify those currently on-call.

This is specific encouragement to Pakistani software, particularly for those who sign code for clients outside of the firm and have valid keys. Go check where your signing key resides and if you have any notes, recordings or confirmations as to what the keys are. If they reside on a developer laptop, or any other place that is not a server then you have a serious issue to fix now rather than waiting until next quarter. There's an underground vendor right now selling your certificates on underground web. They do not care about you, your company or your client. You are just an easy victim and will be sold about ten times the value before they destroy your company (if they don't already have).

About the Author

I am Hadia Zia currently pursuing the degree of bachelor of Computer Science . I can be reached at hadiazia0328@gmail.com.

Keywords: Code Signing; Dark Web; Software Supply Chain; Certificate Authority; EV Certificate; Malware; Pakistan Software Industry



References

- [1] ReversingLabs. (2025). *State of Software Supply Chain Security 2025 Report*. ReversingLabs Research Team. pp. 34-41.
- [2] Recorded Future. (2026). *Dark Web Markets for Stolen Certificates: Growth and Vendor Analysis*. Recorded Future Insikt Group. January 2026.
- [3] Trellix. (2026). *VoidCert: An Interview with a Stolen Certificate Vendor*. Trellix Advanced Research Center. February 2026.
- [4] Microsoft Security Response Center. (2025). *SmartScreen and Code Signing: Best Practices for Developers*. MSRC Blog. 15 October 2025.