Threat Actors abuse signed ConnectWise application as malware builder

By G DATA Security Center

Published: 2025-07-23 · Archived: 2026-04-05 15:02:27 UTC

06/23/2025

Reading time: 8 min (2040 words)

Since March 2025 there has been a noticeable increase in infections and fake applications using validly signed ConnectWise samples. We reveal how bad signing practices allow threat actors to abuse this legitimate software to build and distribute their own signed malware and what security vendors can do to detect them.

Analysis by Lance Go and Karsten Hahn

ConnectWise abuse 2024-2025

This isn't the first time that ConnectWise has been used by threat actors. Back in February 2024, we saw a spike in ransomware activity tied to two ConnectWise vulnerabilities: CVE-2024-1708 and CVE-2024-1709.

Around March 2025, a new wave of ConnectWise abuse started showing up, now being tracked under the name "EvilConwi".

When people suspect an infection, they often turn to the Internet for help. "UNITE against malware" forums (such as BleepingComputer.com) provide disinfection assistance in such cases. Several threads on BleepingComputer's forums (link1, link2) show unwanted ConnectWise clients as the culprit of the infection, usually with phishing emails as the starting point. The existence of several posts like these indicates a failure of security programs to prevent the threat. Even in May 2025 most antivirus products did not detect maliciously used ConnectWise samples as malware.

In one BleepingComputer case the origin of infection is a phishing email with a OneDrive link that promises to show a large document. The link redirects to a Canva page with a "View PDF" button which downloads and runs a ConnectWise installer. The user describes "fake Windows Update screens" and their mouse "moving on its own randomly". Aside from those indicators, there were no other visible signs for the active remote connection (sample[1]).

Reddit users have also reported similar incidents, for example in one case, a maliciously crafted ConnectWise sample[2] originated from a website offering an AI-based image converter. According to the original poster, the site had been advertised on Facebook.

Sample comparison

To figure out detection opportunities and settings location, we analyze the difference between two ConnectWise samples.

The images below show PortexAnalyzer reports for two ConnectWise samples[6][7] which we compare with Meld.

Aside from the certificate table in the overlay, the section contents have the same hashes. We confirmed with a binary diffing tool that the only substantial differences reside in the certificate table.

Thus, any customization that we could use to distinguish ConnectWise installers from each other must reside in the certificate table. At this point we suspect Authenticode stuffing.

Authenticode stuffing

Authenticode stuffing is deliberate misuse of the certificate structure that allows modifications to an executable without invalidating its signature. Developers use this technique to avoid re-signing their applications for minor changes. It's a relatively common practice, applications like Dropbox use it. Some installers[3], for example, track installation statistics by saving user agents, referrers, campaign IDs or similar data from the browser's cookies in the certificate shortly before the file is downloaded. In such cases, the Authenticode stuffing is harmless because it does not influence the sample's behavior.

There are various ways to abuse authenticode signing. To figure out which method ConnectWise uses, we run an authenticode linter on both samples.

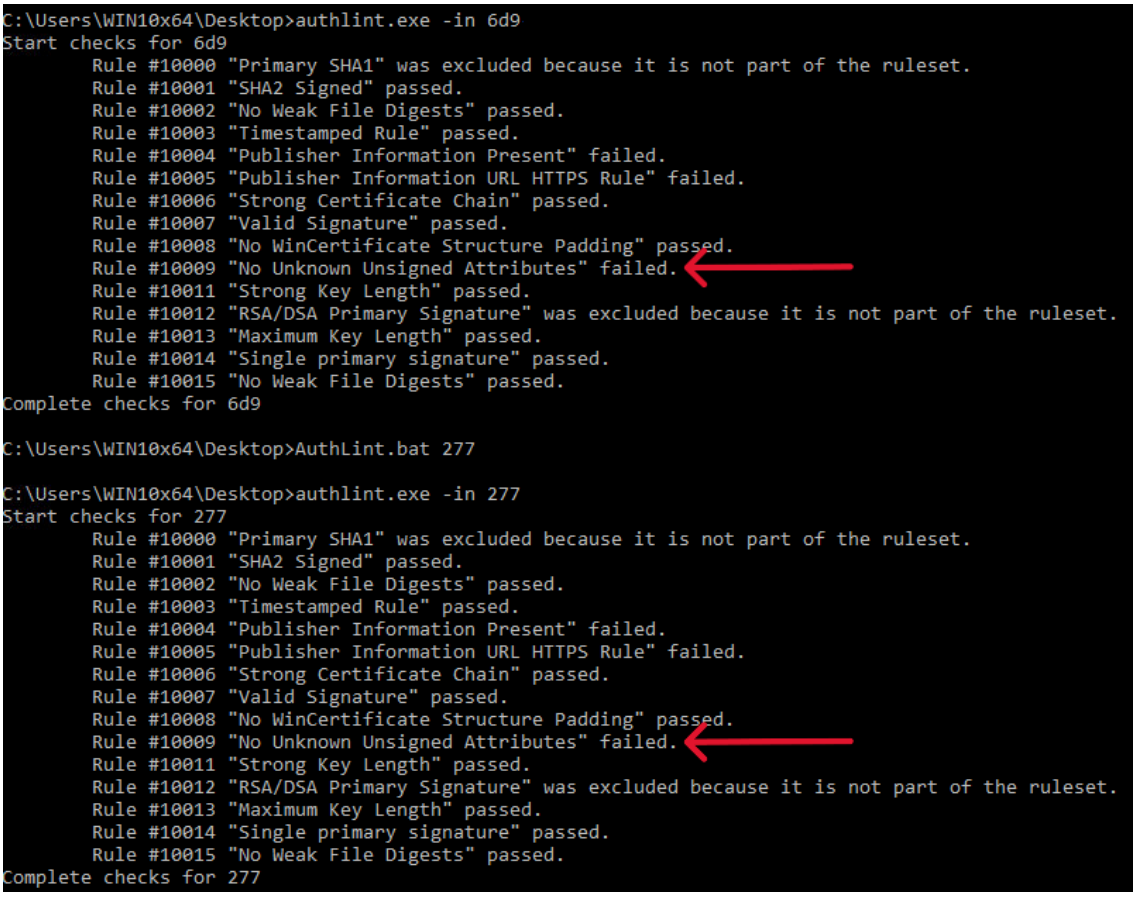

Figure3: output of AuthenticodeLint tool

The linter output shows that the “No Unknown Unsigned Attributes” check fails for both samples. That means ConnectWise has unauthenticated attributes which should not be there.

The following image shows the structure of a signed Portable Executable file, and where the unauthenticated attributes reside in the certificate table. The original image is from Microsoft’s official Windows Authenticode PE Signature Format document.

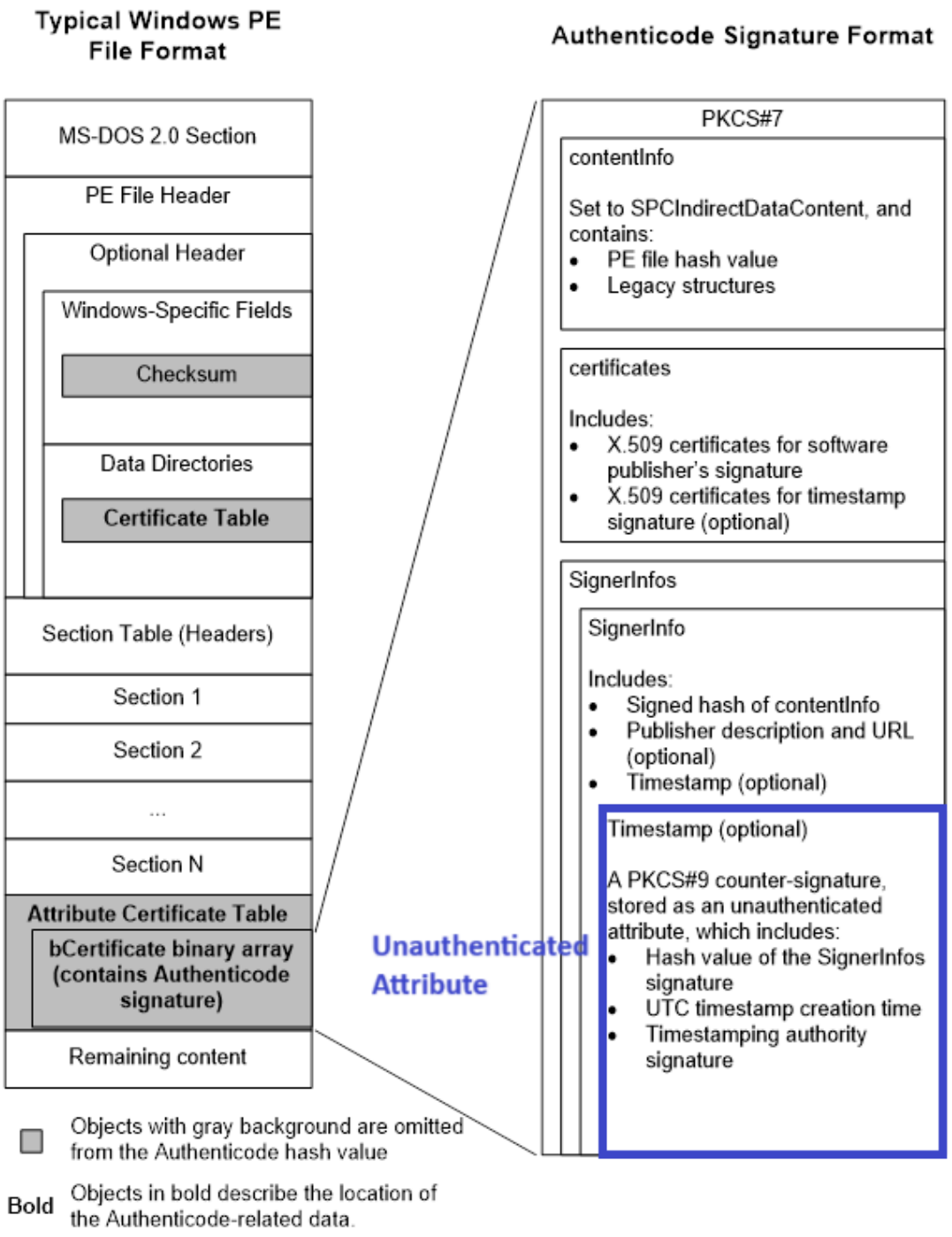

Figure 4: Windows Authenticode PE signature format

To verify the certificate of a Portable Executable file, Windows compares the authentihash in the certificate with the actual hash of the file. If these hashes are different, the verification fails, and the file is not validly signed anymore.

Windows calculates the authentihash on the file's contents except for the grayed-out areas on the left side of the image. That means the checksum in the Optional Header, the certificate table entry in the Optional Header, and the certificate table itself are omitted for the authentihash calculation. This includes unauthenticated attributes. They won't impact the validity of the certificate.

The right side of the image shows the certificate table structure. Unauthenticated attributes usually save timestamps but can also save arbitrary data.

Because we assume that ConnectWise uses unauthenticated attributes for Authenticode stuffing, we create a Python script to extract unauthenticated attributes from PE files.

ConnectWise configuration abuse

We built a configuration dumper that extracts settings and embedded files from the certificate. While we do not share the script for legal reasons, most of the meaningful data that is useful for threat detection is saved in XML format and directly visible in dumped attributes (using the Python script above) or a strings listing of the sample.

Here is an example output of the configuration dumper for a malicious sample[4]:

The first interesting indicator is the connection URL and the port which are part of the launch parameters as well as the silent installation flag which is set to false here. But there is more: embedded additional resources and configuration files.

One of the extracted additional files for this sample is a .NET resource named Client.Override.en-US.resources. In this sample[4] it modifies the ApplicationTitle so that ConnectWise fakes a Windows update and instructs the user not to turn off the system, probably to ensure that the remote connection remains active for some time.

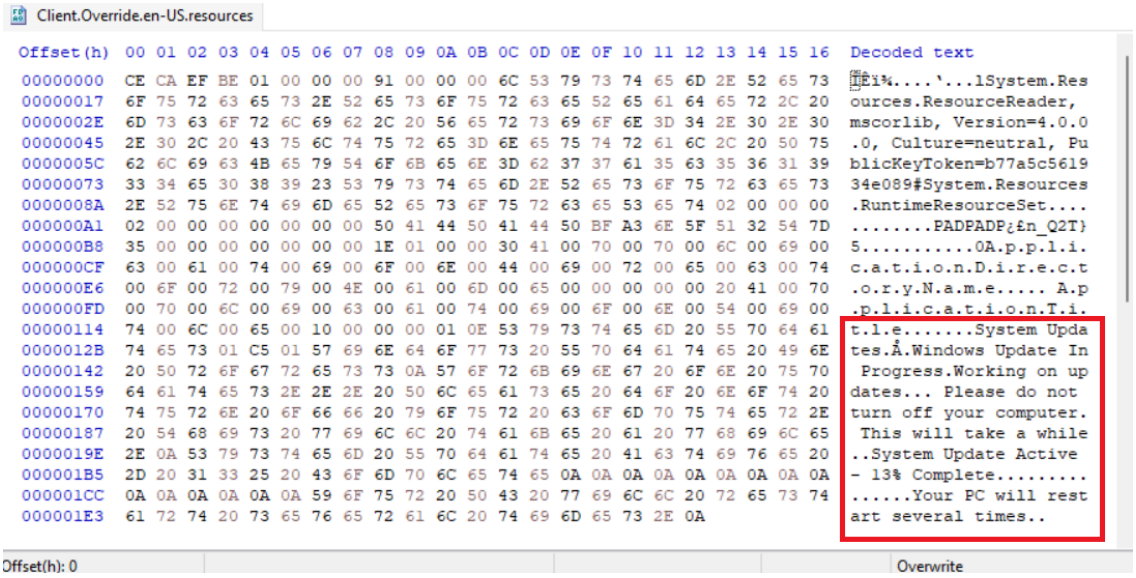

Figure 5: fake Windows update messages in a config file

Another resource named Client.Override.resources contains Google Chrome icon PNG files which override the ApplicationIcon property. Similar samples like [5] use the same file to override the property BlankMonitorBackgroundImage with a fake Windows update screen JPEG. Both images are shown below.

There are also two configuration files named system.config and app.config. These are XML files with more settings. The system.config usually includes another ClientLaunchParametersConstraint value—on top of the one already extracted using the config extractor—which holds the connection URL, port, and other parameters.

The app.config XML is also interesting for threat evaluation. The following image shows the contents of an app.config file that is typical for malicious ConnectWise samples:

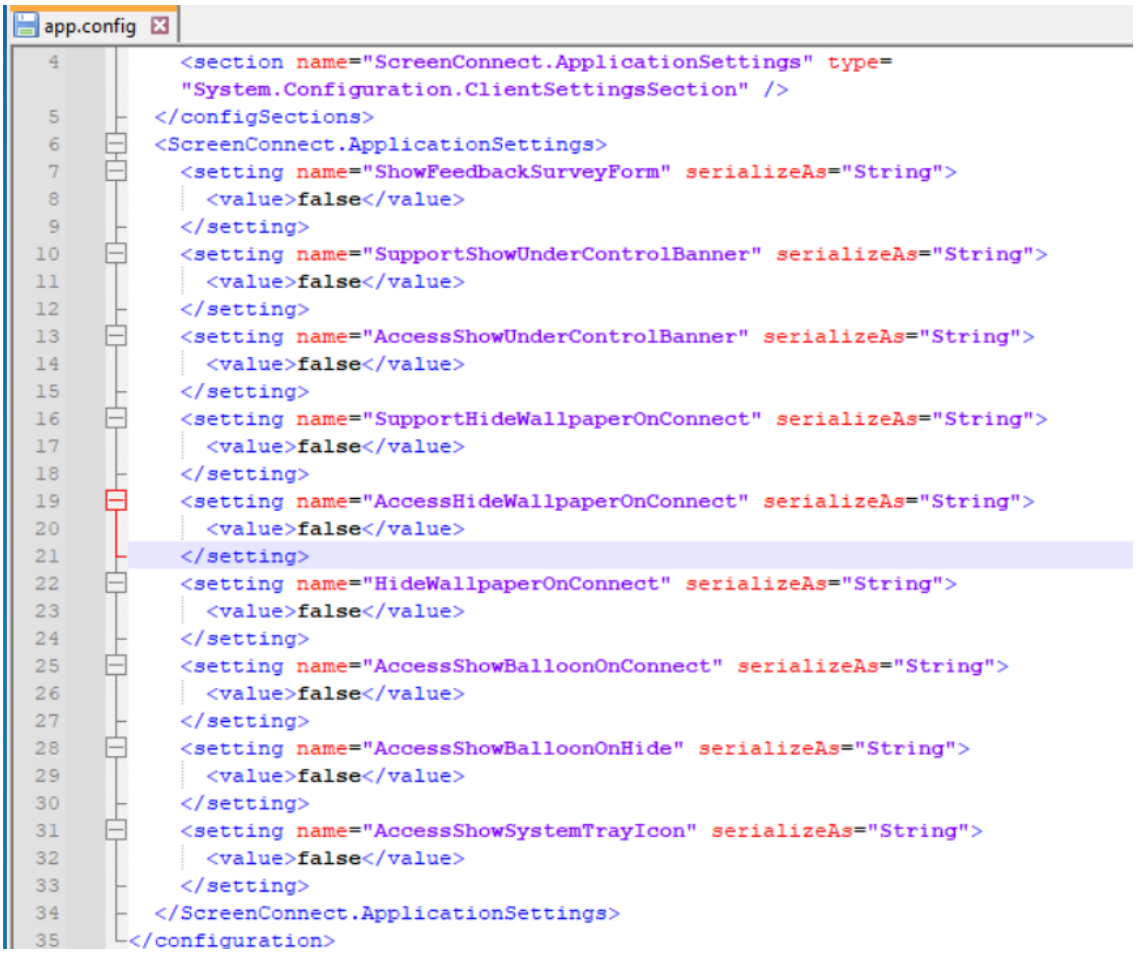

Figure 6: app.config with remote connection indicators set to false

This app.config disables several indicators which would alert a user that ConnectWise is present like a tray icon or a black wallpaper during an active connection.

To summarize, the settings in the certificate table of ConnectWise substantially influence the behavior of ConnectWise installers and clients. Among others the certificate saves:

- Silent installation option
- Launch parameters which include connection URL and Port
- Application icons
- Messages and window titles shown to the user
- Images used by the software, such as background images
- Indicators that show the presence of an active connection

By modifying these settings, threat actors create their own remote access malware that pretends to be a different software like an AI-to-image converter by Google Chrome. They commonly add fake Windows update images and

messages too, so that the user does not turn off the system while threat actors remotely connect to them.

Threat prevention recommendations

We recommend fellow defenders disallowing any ConnectWise samples that have several of the following app.config settings set to false (using regex syntax):

- (Support|Access)?HideWallpaperOnConnect
- (Support|Access)?ShowBalloonOnHide
- (Support|Access)?ShowBalloonOnConnect
- (Support|Access)?ShowSystemTrayIcon
- (Support|Access)?ShowCloseDialogOnExit

A Yara rule may look as follows:

We also recommend detecting fake application titles, fake icons and fake background images that are embedded in .NET resources within the certificate.

GDATA products detect maliciously abused ConnectWise samples as Win32.Backdoor.EvilConwi.* and samples with questionable settings as Win32.Riskware.SilentConwi.*

Vendor practices and end-user risk

Although authenticode stuffing is common practice, ConnectWise's decision to influence critical behavior and its user interface with unauthenticated attributes is clearly dangerous. It entices threat actors to build their own remote access malware with custom icons, background images and text, that is signed by a trusted company.

Given how widely (ab-)used ConnectWise's ScreenConnect is, it is a good idea to keep an eye out for these samples. Until ConnectWise changes their authenticode stuffing practices, the possibility of signed malware being created and distributed remains a threat.

On June 12, we contacted ConnectWise prior to the release of this article to make them aware of the issues described above and give them the opportunity to issue a statement. We noticed on Tuesday, June 17, 2025 that the signature used to sign the samples was revoked. We have not received a statement by the time this article was released.

Samples referenced in this article

[1] ConnectWise from BleepingComputer
7287a53167db901c5b1221137b5a1727390579dffd7098b59e6636596b37bc27

[2] ConnectWise from Reddit
7180238578817d3d62fd01fe4e52d532c8b3d2c25509b5d23cdabeb3a37318fc

[3] Setup file with tracking data in certificate
a6fb2a4be91f6178d8ba0ca345727d1cb7995c3e4a659a68bef306c9eff4b18e

[4] ConnectWise sample with fake Windows update messages and Chrome icons in config
cb8a1a1e90c29461b0503e2c5deac7b673617477128ee3baea4d8134676c8af4

[5] ConnectWise sample with fake Windows update screen in config
28f46446d711208aa7686cdaea60d3a31e2b37b08db7cfb0ce350fcd357a0236

[6] ConnectWise sample used for comparison
6d9442ae6ba5a9f34a47e234b6047f61d8ac129e269199793ebb0bed1ad7e3ba

[7] ConnectWise sample used for comparison
277ef6c0dcaf0e76291fbde0199dda1ca521c03e77dc56c54f5b9af8508e6029

Sample hashes sorted by infection vectors

Based on collected samples and their naming schemes, we observed certain patterns as possible infection vectors.

Fake installers

540c9ae519ed2e7738f6d5b88b29fb7a86ebfce67914691ce17be62a9b228e0a, ZoomInstallerFull.exe

55a228f22f68b8a22967cc5b8b2fcbea66fcaf77bebedfb1f89cd134a0268653, zoom_meetingconnect.exe

C0c48de11bc4b70fb546b9a76b6126a355c0a0f4b45ed6b6564d8f3146c9f0af, ZoomInstaller-x64.exe

67b909bbcce486baba59d66e3b4ec4c74dd64782051a41198085a5b3450d00c9, OneDriveSetup.exe

b1c36552556a69ec4264d54be929e458c985b83bbc42fe09714c6dce825ac9a7, MicrosoftExcel.ClientSetup.exe

D37e804938cf0a11c111832b509fbecf8a0f3e9373133be108d471d45db75de8, Adobe-Update-ClientSetup.wSZQ5iHP.exe.part

b61aed288b4527b15907955c7521ff63cc0171087ac0f7fea6c7019a09c96c04, Adobe.ClientSetup_v7.-2.7.exe

6bce39b7d7552dbacbb4bdf06b76b4fed3fbb9fe4042b81be12fbdff92b8d95c, SSA Viewer.exe

Fake video or movie clients

6aa1b9f976624f7965219f1a243de2bebb5a540c7abd4d7a6d9278461d9edc11, Creation_Made_By_CanvaAI.mp4 Canva.com

8fc8727b6ddb28f76e46a0113400c541fb15581d2210814018b061bb250cc0e6, FULL_MOVIE_DOWNLOAD.exe

5da9a0d0830c641ffda6be3be7733de469418abedc6fac0cfcd76ba49f8ade2e, P0RN-vidz.Client.exe

72fe38ad67a26cfd89d1bfc744d33f80277e8eb564b5b92fdac46a9a24d845f3, P0RN-vid.Client.exe

5ccc9ef3e8f7113469f4a46c3aca3939fd53b3561a9fd8ffacd531aa520c5921, FULL_MOVIE_WATCH_NOW.exe

23ff4f91db852b07c7366a3c3b8be0bade2befccbfea7e183daadb5e31d325c0, Schau mir jetzt nackt vor der Webcam zu.exe

Fake documents

41037935246da6f43615d93912bc62811c795ea4082a2bfdbf3eda53a012666e, Social_Security_Statement_873164.exe

98e3f74b733d4d44bec7b1bf29f7b0e83299350143ff1e05f0459571cb49c238, Statement.pdf.Client.exe

d6844a6050d5f6c20a3fe12df28e53a2e46559e6c5017576022372e35ab44ff5, SSA-statement-osu5ma6.PDF`.exe

573f1eefac3079790a9ab40bdd3530ce34b1d2d1c6fa6703a5a8d81cb190a458, BarryStatementPDF.exe

F55c6160ed57a97c4f0e1c6aa6e3f8f01a966e96a99a29e609ec60e63be11889, FATURA-255441144227D55224QO02GX6QL.com

4e5cfd915f44dc263f29e1eaef82b3e2e903ba92b10f88c0eaf89fe5eab82ff5, ANFRAGE FÜR VORSCHLAG.exe

E7f9b9c9205162ddee72a7b7ff86b6524e19c7e8b51f64fdbffc8015c7e8934c, Important Document.exe

---

Share Article

Content
