Code-signing certificate abuse in the Black Basta chat leaks (and how to fight back)

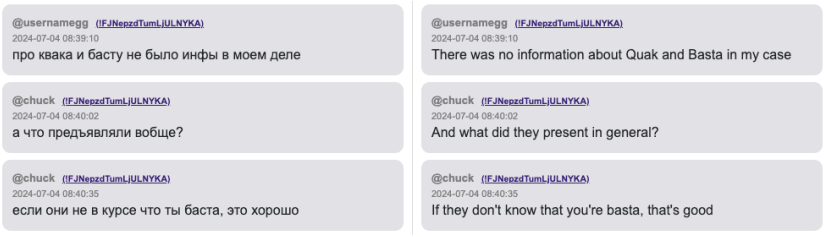
By Aaron Walton

Published: 2025-03-18 · Archived: 2026-04-05 20:29:20 UTC

TL;DR

- Code-signing helps computers have confidence in the legitimacy of files, but cybercriminals abuse them too.
- BlackBasta's leaked chat logs give us insight into how a ransomware gang leveraged code-signing certificates for their malware campaigns.
- By investigating code-signing certificates and reporting false signings, you can protect yourself and others from malicious activity.

Recently, we (the information security community) received the opportunity to look behind the curtain and see the inner workings of the Black Basta ransomware gang. In this post, we'll use the opportunity to examine how the ransomware gang used their skill and finances to abuse a core security concept: code-signing certificates.



@usernamegg discussing his escape from arrest in Armenia with @chuck. Quak refers to what defenders track as Quakbot, or Qakbot, malware.

@usernamegg: There was no information about Quak and Basta in my case

@chuck: And what did they present in general?

@chuck: If they don't know that you're basta, that's good

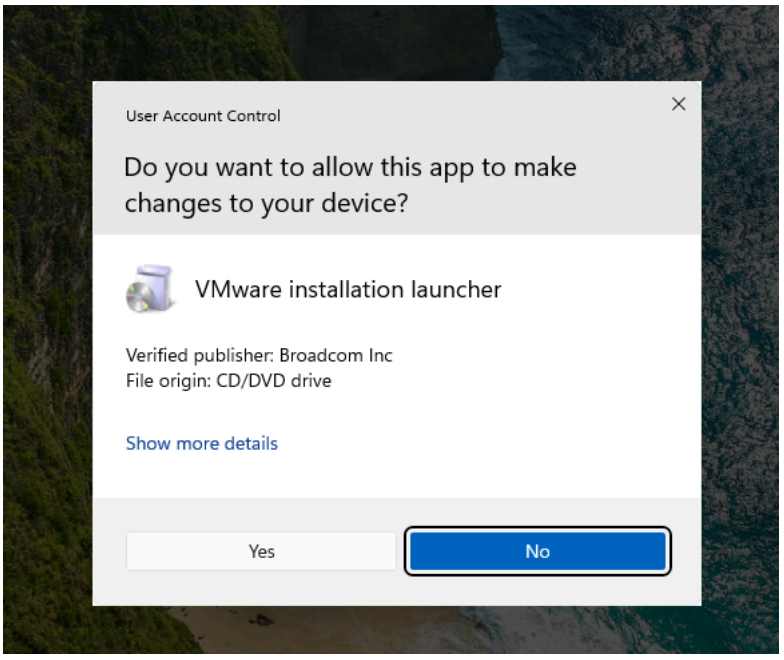
Core safety via code-signing certificates

Computers have become an essential part of our lives and businesses. To help make this possible, a lot of time has gone into helping ensure systems exist to make using computers safe. One of the core principles to help do this is code-signing certificates.

Code-signing certificates help solve some basic problems. They help answer the questions "*How can I trust this program?*" and "*How can I trust this program has not been tampered with?*"

Code-signing certificates are issued to vetted organizations. This vetting process results in a chain of trust: a root authority trusts a certificate authority, and the certificate authority vets and validates their customers. After validation, the customer is trusted because the certificate authority is trusted.

This chain of trust impacts how web browsers and operating systems handle many files. If the file isn't signed, many browsers and operating systems will show very clear warnings. However, if they are trusted, those warnings may not appear, or may only be informational.



This is an informational User Access Control message. The file is signed by Broadcom Inc.

This system helps us have confidence that a system file or downloaded application is legitimate, and that the developer has been vetted.

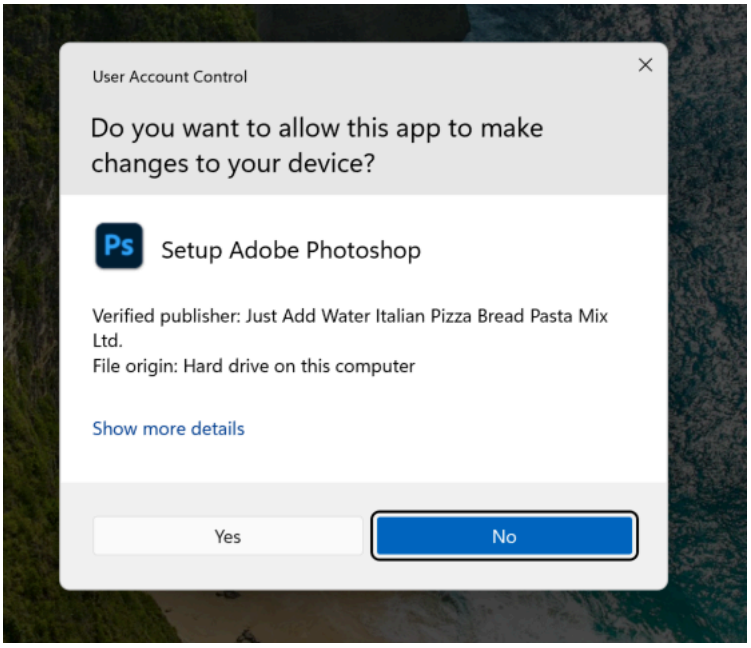
It uses public key cryptography to ensure the signatures are valid. When a file is signed, it requires the signing party to have a private key (often tied to a hardware token) to apply a signature to a file. This ensures entities without the private key can't sign as the software developer.

Signing also protects against file tampering. The signature tied to the file contains a cryptographic hash of the file, and this hash must match the computed file hash of the file, or the signature won't be seen as valid. If you aren't familiar with hashing, this is a technique of applying a one-way algorithm to a file to compute a small representation of the file. Any modifications to the file will impact the representation or hash.

These processes work pretty seamlessly: when you're downloading or running files from trusted sources, the chain of trust helps you feel safe about the file you're running. Without this system, the user is required to validate a lot of information on their own (such as the source of the download) and calculate their own file hashes to compare against a public list of file hashes.

Abusing code-signing

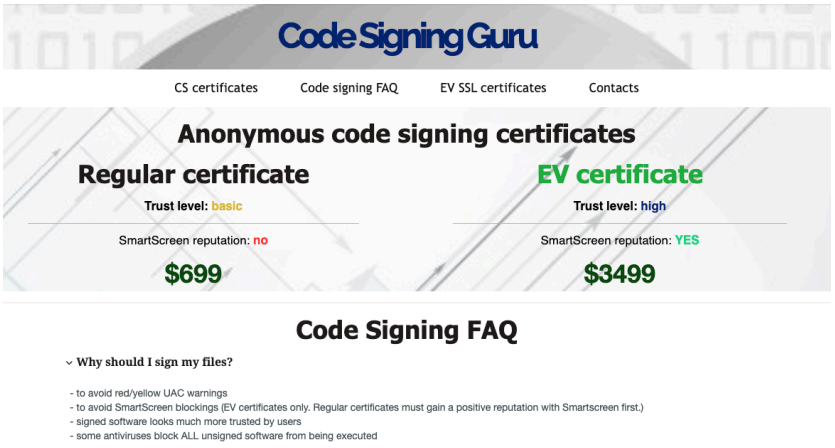
This code-signing process and transfer of trust works well, but just like anything within cybersecurity, attackers can *also* leverage the process. Cybercriminals also want to ensure browsers and operating systems trust their files, too, so they're incentivized to find ways to abuse this system.



This is an informational User Access Control warning. This Adobe Photoshop file is signed by Just Add Water Italian Pizza Bread Pasta Mix Ltd. (spoiler alert: it isn't a legitimate copy of Photoshop).

In very rare cases, cybercriminals can steal the private key and then sign files as an organization. There are some very well-known cases of this happening, but since the private key is connected to a physical hardware token, usually it's very uncommon.

The most common method of abuse today is to impersonate a company and be issued the keys directly. Instead of each bad actor needing to do this themselves, certain attackers specialize in this impersonation and resell the certificates. One well-known seller is Megatraffer, who has been selling certificates for a long time, and was also selling certificates to the Conti ransomware gang.



An image capture of Megatraffer's Code Signing Guru website as captured by the Wayback machine. Here, Megatraffer advertises why criminals need code-signing certificates for their malware.

This type of abuse has been previously written about and published by others. However, we recently got a good look at this type of abuse from the inside: from the chat logs from the Black Basta Ransomware gang.

Chat logs made public

On February 20, an individual going by ExploitWhispers announced they were publishing chats from the Black Basta ransomware gang. ExploitWhispers stated they did this because Black Basta had crossed the line by compromising a Russian bank.

It's generally taboo for Russian cybercriminals to target entities in the Commonwealth of Independent States (former USSR countries). Targeting these countries could result in legal action, and unlike the West, have a larger chance of manifesting into charges. Targeting entities in these countries also impacts the lives of fellow countrymen.

These types of leaks are valuable to defenders like us because they often expose uncensored discussions between cybercriminals. Several researchers and organizations have also already dug deep into these leaks, observing how bad actors handled the Ascension Healthcare ransomware attack internally, learning about how the gang leader escaped detention in Armenia, or documenting the vulnerabilities and tactics the gang discusses.

A picture of a translated chat message. A leading member shared his opinion on how to handle blowback from encrypting devices in the Ascension Healthcare ransomware attack. Ransomware gangs regularly try to rebrand to avoid heat from law enforcement.

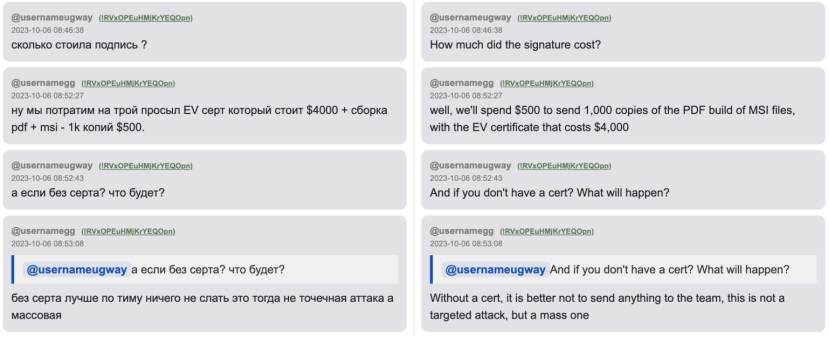
@usernamegg: But after such a f***, we need to rebrand or we will be taken by the a**

The leaks themselves cover time from 2023-09-18 to 2024-09-28. There are almost 200,000 messages, 50 unique users, and 79 chatrooms (source).

From our own visibility, we knew the Black Basta ransomware gang were frequent abusers of code-signing certificates. Specifically, our SOC frequently saw Black Basta sign Pikabot and Darkgate malware they used in phishing campaigns, so we dug in to see what we could learn.

Why sign?

In the early parts of the chatlog (2023-10-06), the bad actors discuss the cost and value of using code-signing certificates (the following machine-translated from Russian and edited for style and clarity.)



@usernamegg and @usernameeugway discuss the costs associated with the campaign and signing files.

@usernameeugway: How much did the signature cost?

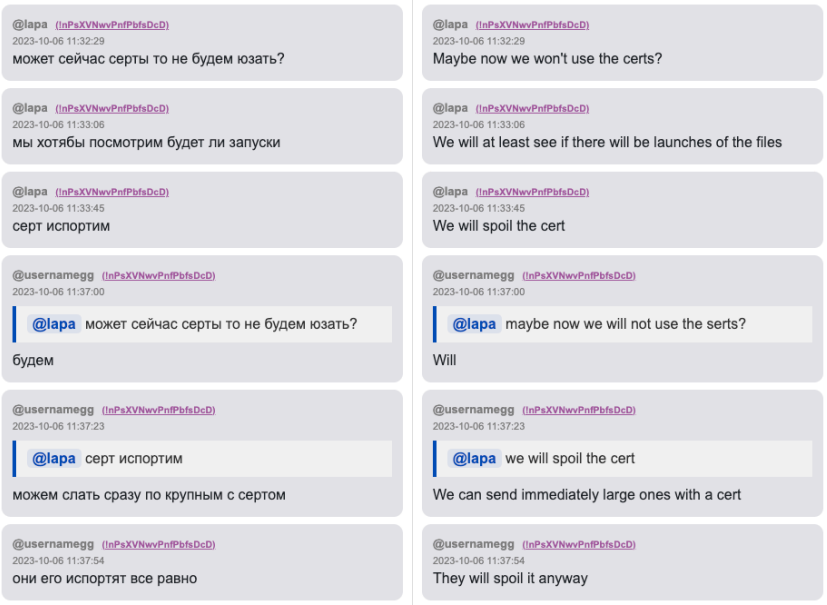
@usernamegg: Well, we'll spend $500 to send 1,000 copies of the PDF build with the MSI files, with the EV certificate that costs $4,000

@usernameeugway: And if you don't have a cert? What will happen?

@usernamegg: Without a cert, it is better not to send anything to the team. This is not a targeted attack, but a mass one

The main person handling the certificates goes by "gg" in the chats, and is known to be one of the main leaders of Black Basta. He had formerly been part of the Conti ransomware gang, and is the most active person in the chat logs.

Within the chatlogs, gg frequently talks about handling the signing of files. When he shares the files, they are always nicely labeled, which helps us get a good understanding of their contents. In general, they freely use code-signing certificates and don't hold back on buying or using them.



@lapa and @usernamegg discuss using a certificate for a test before a campaign.

@lapa: Maybe now we won't use the certs?

@lapa: We will at least see if there will be launches of the files

@lapa: We will spoil the cert

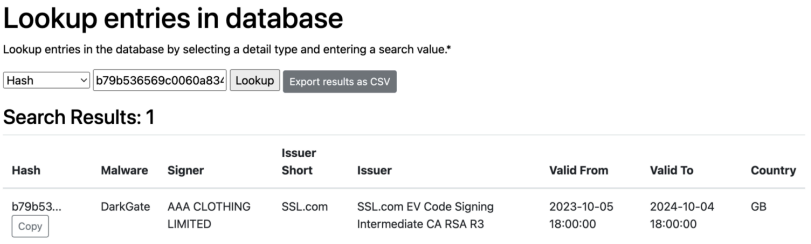
@usernamegg: Will

@usernamegg: We can send immediately large files with a cert

@usernamegg: They will spoil it anyway

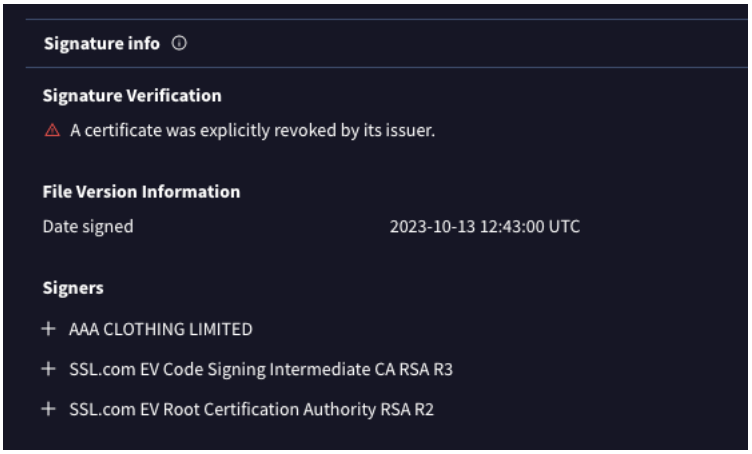
In most cases, signed files gg shares are labeled like this: "EV## Impersonated Organization name [Certificate Provider].rar" (for example, "EV44AAA_CLOTHING [SSL.COM].rar"). We can further correlate this particular certificate with a file in the The Cert Graveyard database.

The Cert Graveyard is a public database that tracks code-signing certificate abuse.



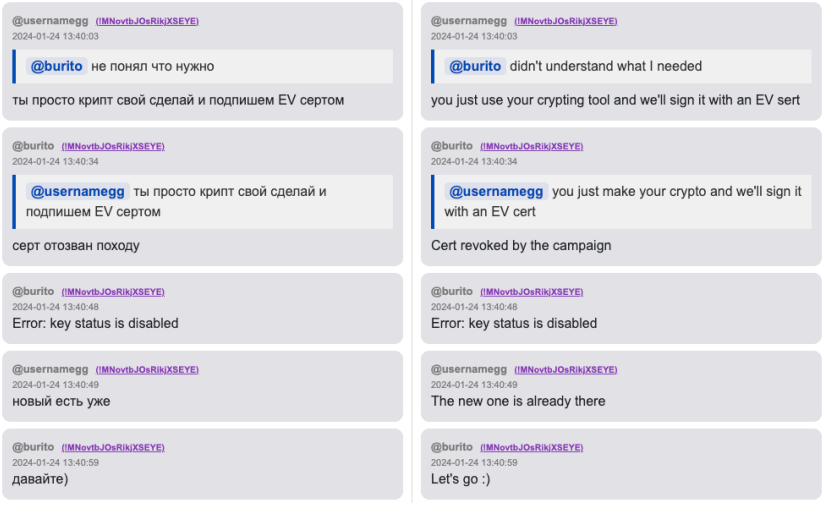
The file's entry in the The Cert Graveyard database.

The Cert Graveyard documented the abuse of a certificate issued by SSL.com to AAA Clothing Limited. They identified the malware as DarkGate malware, and provided the file hash b79b536569c0060a834e4001289a6700692d67df58e644779fababf0df22fc75. This file is also publicly available on VirusTotal.



VirusTotal indicates the certificate for AAA Clothing Limited was reported and revoked.

The chats mention *at least* 28 certificates. The entire list we identified is provided at the end of this document. Like the AAA Clothing Limited file, most are numbered and follow the pattern mentioned above. The file numbers range from 13 to 101, so the fact that we only have 28 may indicate many more are unaccounted for. Reporting certificates for revocation is important, but with gangs like these, they often have many on hand.



@usernamegg and @burito are discussing their ability to generate new certificates quickly when a previous one is revoked.

@usernamegg: You just use your crypting tool and we'll sign it with an EV cert

@burito: Cert revoked by the campaign

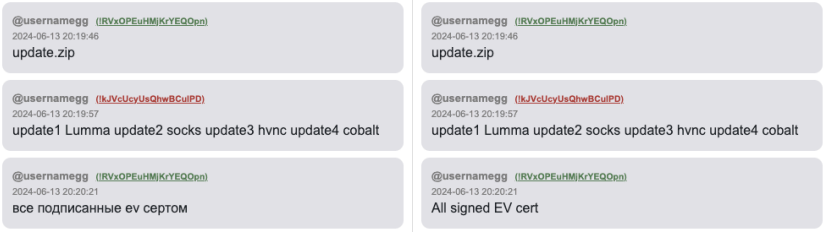
@burito: Error: key status is disabled

@usernamegg: The new one is already there

@burito: Let's go 🙂

A crypting tool is a common means to hide the functionality of a file.

The possibility that they could have had 100+ code-signing certificates isn't unusual. According to The Cert Graveyard, some malware families have been seen signed with code-signing many times. This includes Qakbot, which has been seen signed by impersonated organizations more than 100 times. The gang also signed many different files within a campaign—not just their initial access malware.



@usernamegg confirms the contents of files they plan to use in a cyber attack.

@usernamegg: update.zip

@usernamegg: update1 Lumma update2 socks update3 hvnc update4 cobalt

@usernamegg: All signed EV cert

The gang uses code-signing for multiple components of their campaign. This includes signing initial access tools (IATs) like Quakbot, Pikabot, and Darkgate; information stealing malware, like Lumma infostealer; tools to hide one's presence on a computer and network, like SOCKS proxy and Hidden VNC; and their featured remote access tool, Cobalt Strike.

Code-signing certificate abuse is often misunderstood. However, these chats give us a lot of great visibility into how this gang abuses them. We've seen how much they pay for them, how liberally they use them (due to their critical role in their campaign), and we've seen them used to sign a range of malware.

Leveraging the leaks for good

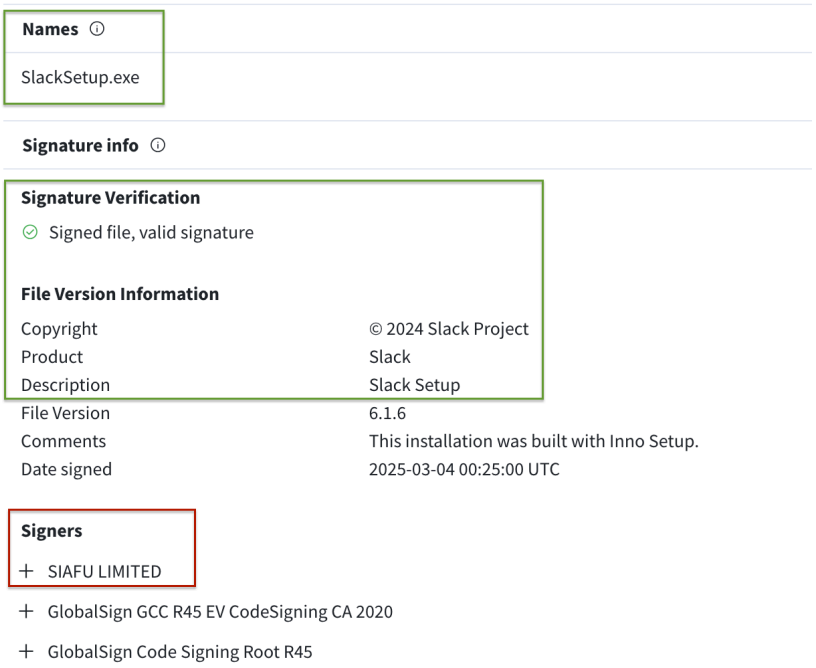
Just knowing ransomware gangs and other criminals abuse code-signing certificates isn't enough. It's important for organizations to leverage that knowledge for their defenses, too. Here's what you can do.

1. Investigate files with uncommon code-signing signatures.

Malware traffic generation teams (also known as traffers) often sell a service to hide the malicious code (using what is called crypting) and provide code-signing services. As a result, signed malware has a low detection rate, which increases the chances of success. Malware with high detection rates is unlikely to be signed to avoid "spoiling the cert."

At Expel, we frequently see malicious advertisements for common apps. These advertisements push low-detection malware. It's important to investigate the file, because the code-signing certificate can often be an indicator of malicious activity. One way to investigate these is to compare what the file claims to be with its signer.

In the following image, the file on VirusTotal was uploaded as SlackSetup.exe. It was downloaded from a malicious advertisement for Slack. The file information (which is basically just a text field an attacker can fill in) also claims the file is Slack. However, the certificate signer is "SIAFU LIMITED" and not the expected signer: **Slack**.



The file and certificate details of a file on VirusTotal. The items highlighted in green are inconsistent with the signer, making the signer suspicious.

By reviewing and evaluating what the file claims to be versus who has signed the file, analysts can quickly identify whether the file is legitimate, regardless of the detection rate. (At the time of this writing, the file was detected as malicious by four of 68 detection engines. But seeing completely clean malicious files is also fairly common.)

Organizations can manually review these features or use automation and AI. You do you. 🙂

2. Submit reports for abused code-signing certificates.

Certificate providers are really responsive to reports. The certificate provider previously vetted and trusted a customer, so receiving reports of abuse allows them to take action and revoke the certificates. To report a certificate, the file must be publicly available for the provider to validate your claim of abuse. The easiest way to make it available is to upload the file to VirusTotal and give that link to the provider. We also recommend providing a detailed report of the activity you observed indicating abuse, especially if the file is clean in terms of detection. The information you provide helps them identify malicious activity.

The maintainers of Cert Graveyard also have a tool called certReport, which can generate abuse reports in a few seconds and direct you on where to report the certificate. This tool leverages VirusTotal to collect the important details on the certificate, as well as any suspicious indicators identified by VirusTotal.

Reporting the certificate can also:

- **Costs criminals money.** As mentioned in the leaks, code-signing certificates regularly cost $4,000 or more, because obtaining the certificate takes a lot of work. Reporting the certificate causes criminals to have to spend even more money, or risk having a completely useless campaign.
- **Disrupt future downloads.** When a certificate is revoked, the file is now viewed as worse than an unsigned file. Both browsers and operating systems will reject a revoked certificate as explicitly untrusted. This can help protect users in your organization and outside your organization from downloading the same malware.
- **Disrupt malware delivery.** As seen in the chats, when a certificate is revoked, the certificate can't sign files anymore. This can disrupt campaigns where a bad actor is trying to deliver signed files. We regularly see malware traffic teams disrupted due to the revocation of their certificates.
- **Help defenders identify malware.** Since most certificates issued to impostors are used to sign multiple files, they all become suspect once one is reported. When certificates aren't revoked, they are reused across malware. Identifying other files with a malicious certificate can help identify low-detection malware that can then be investigated and analyzed to build new detections to find them the next time they are seen.

Black Basta's known certificates

The following table contains the certificates mentioned in the chats. We checked to see if they were publicly known in the Cert Graveyard database. If they were known, the file hash was provided.

This list was also provided to the certificate issuers listed in the table for their awareness.

Number from chat	Subscriber	Issuer	Date seen	Thumbprint	Hash example (if available)
EV1	AproFoods LLC	GlobalSign	*Unknown*	*Unknown*	*Unknown*
EV4	Avikser LLC	GlobalSign	*Unknown*	*Unknown*	*Unknown*
EV6	Aprima LLC	GlobalSign	*Unknown*	*Unknown*	*Unknown*
EV13	Stimul LLC	GlobalSign	1/31/24	F89A8B321959FED4963D8DF10996E1A9BD07119D	b758b935fc420e334d8afdff6d
EV23	LLC SERVER	GlobalSign	4/24/24	2B20EE6FB83FF52BDD2714741A8783981795B8E7	315e6d1736e2ec8465a172d28
EV24	LLC CESARIA	GlobalSign	5/31/24	239E18C2FF083DAB3546B83BE3CC00756442047D	ec3ca0877e599ae9c40cbcec51
EV32	Primak LLC	GlobalSign	10/2/24	*Unknown*	*Unknown*
EV37	MK ZN S.R.O.	SSL.com	9/28/23	0D762B095F6F2BA2DBEB00C5B8E9C93294FAD66F	4325d78175a803fb6a1d235e8
EV41	MK ZN S.R.O.	GlobalSign	10/12/23	*Unknown*	*Unknown*
EV42	AAA Bio Mass Services	SSL.com	*Unknown*	*Unknown*	*Unknown*
EV43	Fast Colibri	SSL.com	*Unknown*	*Unknown*	*Unknown*
EV44*	Media Box	SSL.com	*Unknown*	*Unknown*	*Unknown*
EV44*	AAA Clothing Limited	SSL.com	10/5/23	DF4E044C56147E7629B9C7781A5FE88996F91C5D	b79b536569c0060a834e40012

EV45	SIA “VIK CAR”	SSL.com	*Unknown*	*Unknown*	*Unknown*
EV47	Acacia Wood limited	SSL.com	*Unknown*	*Unknown*	*Unknown*
EV48	Andapak Corrugated Sales Limited	SSL.com	*Unknown*	*Unknown*	*Unknown*
EV53	Amazing Projects	SSL.com	*Unknown*	*Unknown*	*Unknown*
EV54	Stone Canvas	SSL.com	*Unknown*	*Unknown*	*Unknown*
EV56	Wallfort	SSL.com	*Unknown*	*Unknown*	*Unknown*
EV57	Freeze Me Ltd	SSL.com	*Unknown*	*Unknown*	*Unknown*
EV60	Soft Blanket	SSL.com	11/3/23	17E254F06BCF34A77A3797C5382E4BC064D2328D	f119f1e813cdb8dba30bd3348e
EV61	Soft Comm	SSL.com	*Unknown*	*Unknown*	*Unknown*
EV62	Sky Wine	SSL.com	*Unknown*	*Unknown*	*Unknown*
EV68	SSTextiles	SSL.com	*Unknown*	*Unknown*	*Unknown*
EV71	Share Holding	SSL.com	*Unknown*	*Unknown*	*Unknown*
EV75	Miniboss	SSL.com	*Unknown*	*Unknown*	*Unknown*
EV76	Dentinum	SSL.com	*Unknown*	*Unknown*	*Unknown*
EV77	Seed Plant	SSL.com	*Unknown*	*Unknown*	*Unknown*
EV78	Get Natural	SSL.com	*Unknown*	*Unknown*	*Unknown*
EV80	New Print	SSL.com	*Unknown*	*Unknown*	*Unknown*
EV81	Fisker Fashion	SSL.com	*Unknown*	*Unknown*	*Unknown*
EV85	SOFTWARE MEDICAL DEVICES LIMITED	SSL.com	12/15/23	7917A946ED473A0E81BD4501B0B1736FB1AC653D	fda2abd24764809fb36d4d2ee7
EV89	Kim Chick Sexing	SSL.com	*Unknown*	*Unknown*	*Unknown*
EV90	4leaf Holding Corp.	SSL.com	*Unknown*	*Unknown*	*Unknown*
EV93	ARCHIKADIA SP Z O O	SSL.com	1/15/24	566E7BCC466E79F9A21D4FF7DFF0A407D76B41F9	6c91b714aefef2438be04161d8
EV94	Talk Invest ApS	SSL.com	1/19/24	7B75394FF02197A21E6F683A717CB5A94C7C3DAE	1626880b917b7f5756109dcb6
EV95	A.P. Hernandez Consulting s.r.o.	SSL.com	1/25/24	2941D5F8758501F9DBC4BA158058C3B5	89dc50024836f9ad406504a3b
EV99	4leaf Holding Corp.	SSL.com	1/26/24	94BACD94876552AA683B8D9E4772A0E37C985E30	3a993c44e39c426239051b00a
EV101	Show Down	SSL.com	*Unknown*	*Unknown*	*Unknown*
Unknown	TAIM LLC	GlobalSign	10/5/23	4CB87577FA5B91346CCE30FB9FF3139D46DE3361	5be959722d8cd4bfd6f88a490
Unknown	Ken Friedman AB	SSL.com	12/26/23	BB296138FB75F5CEB45E36B85A8DF7CC82C6364C	8db0b8f45f726a963b34410c7
Unknown	Clover Field ApS	SSL.com	12/14/23	1C2C084FB6E18A4033B63E619868CF81819BF46E	e88610db05636a1476435ec1f

** While two certificates* can't *both be EV44, these numbers are directly from the chat and could not be confirmed.*
