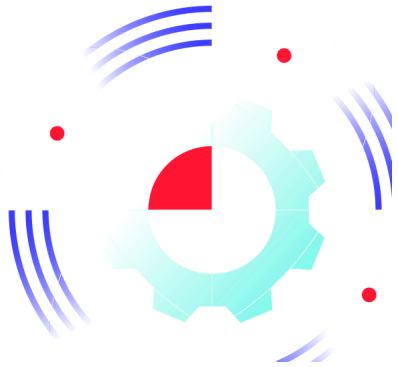
ZIP files, make it bigger to avoid EDR detection - Gatewatcher

Archived: 2026-04-05 22:37:12 UTC

Our Purple Team analysts have spotted a number of anomalies concerning zip files. ZIP is originally a file format for archiving and compressing data without loss of quality. These files attracted our attention because they were abnormally large compared to the size of the zip archive, suggesting the presence of malware, for example.

In this example, the zip file is 2.5 MB in size, and the unzipped malware is 2.1 GB. After some research on VT (virus total) and other tools, we decided to investigate further.

This article reports on what we found.



ZIP and its usefulness

Before any advanced analysis of this file and its contents, we find out about the associated compression rates. For a PE file, the maximum compression ratio is around 50%. In other words, if we have a 1GB executable file and we compress it, we should end up with a zip file of around 500MB.

Following this rule of proportionality, our file should be around 1GB, which is absolutely not the case. Starting with a 2.1GB file and arriving at a 2.5MB zip archive, we obtain a compression ratio of 99%.

It appears that from this address, a padding of 0 is applied to the end of the file. Once this padding has been identified, it can be removed, giving us a functional PE file of around 1 MB. The latter, after compression into a Zip file, is around 500KB, thus verifying for this file the proportionality mentioned earlier.

So, we've identified why the compression ratio was so high, but why would we need to artificially obtain such a high compression ratio?

The answer lies in the detection systems and their parameters.

As the PE file in the zip is most likely malicious, this technique would allow it to slip through the net of most EDRs. As EDRs have a default “limit”, files larger than 1GB, or even 2GB in some cases, are not analyzed. This technique is not new, but it is resurfacing with the rise of EDRs, which are just one of the many layers in a company’s attack detection and incident response system.

Identity and main characteristics of the file

Once the file had been extracted and sent to various sandboxes, we were able to get a more precise idea of the PE file and what it was doing.

Thanks to our research, we can now say that we’re dealing with an Asyncrat agent, a RAT (Remote Access Trojan) that can take control of a remote workstation via an encrypted connection and perform all kinds of malicious actions.

Following our research and the various reports we were able to find on the subject, we identified the attacker’s command server.

In order to find out more about the infrastructure, we set out to find out a little more about the server in question. When looking for information on 45.81.243.217, we find a number of open ports and certificates. Let’s take a look at the certificate with CN (Common Name) Asyncrat. This information allows us to conclude that Asyncrat embeds certificates by default.

There are over 47,300 servers worldwide with the same certificate. As these servers use an Asyncrat certificate by default, it is more than likely that they are deployed by inexperienced attackers, or pentest teams, with a default configuration. This makes them particularly recognizable.

ZIP and padding, or how to avoid detection techniques ?

During our investigations, we were able to report the use of a detection evasion technique, padding, to artificially increase the size of the malware when decompressed.

In the course of our research into this malware, we identified it as a Trojan of the Asyncrat family. We also found the attacker’s command server, enabling him to dialogue with the implant. By following the certificate we discovered a number of servers administered by attackers with a configuration using an Asyncrat certificate.

IOCs concerning the malware described and another malware using a similar padding technique:

Malware	hash
Asyncrat (archive)	f37bd1c01d5b6e9b5d0cab196b7808994af8daecfd25231846d1f49f7e1a092b
Asyncrat (padding)	b3ebb2ed5e94506334e792f524c8a411296886e0d978aced4446a419e575ae55
Asyncrat (sans padding)	a6690545f411e0d746d9b4738a597393a8775c9768d280cfe3fc483b0a325d40
AgentTesla (archive)	6faa0871251d2128d0374eadb6d82e77cab0a63d5ad925f349c578a3fca63966
AgentTesla (padding)	80d09eae137e35d64532c4ea9583fc7a3eae1a9141b0dda04746af18e1e4f82d
AgentTesla (sans padding)	a7b7711f4675f581f99a9715285ba966d642df24a560607836f7044181f31099

IOCs concerning Asyncrat's infrastructure:

Infrastructure	@ip
C&C Asyncrat	45.81.243.217

Source: https://www.gatewatcher.com/en/lab/zip-files-make-it-bigger-to-avoid-edr-detection/