Proactive response: AnyDesk, any breach

stairwell.com/resources/proactive-response-anydesk-any-breach/

Research

Written by **Threat Research**at Stairwell

February 2, 2024

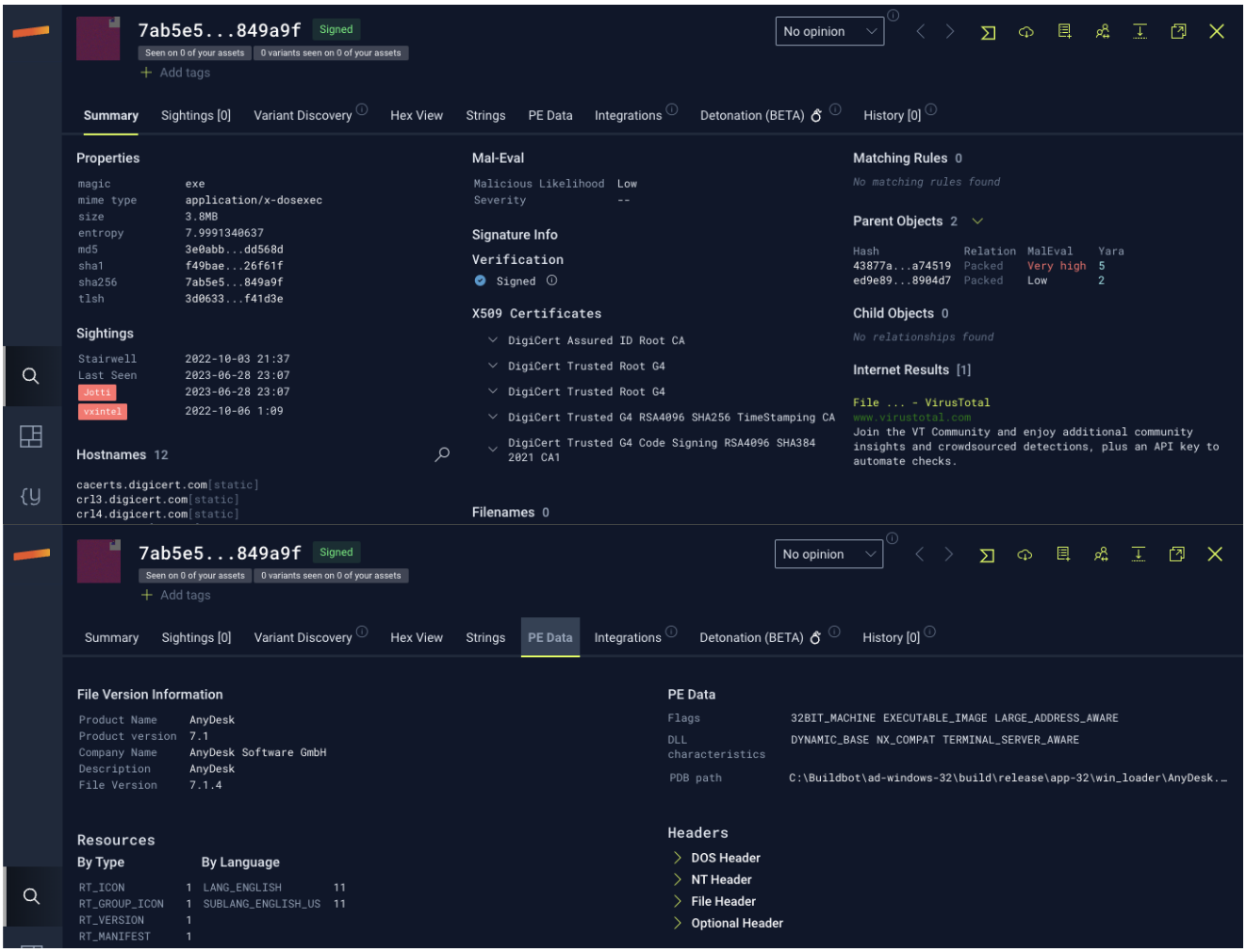
The Stairwell research team closely monitors news for security events that could potentially impact our customers and the world at large. Among rumors of a breach at AnyDesk, we started proactively working to develop YARA rules and hunting methods to help our customers rapidly respond. At the time of this blog, this breach remains unconfirmed – which is precisely why we are writing this report. During the window of time between rumor and confirmation, organizations can take proactive steps to evaluate their exposure.

This blog post covers our initial approach and potential detection methodologies that serve as a foundation upon which others can build.

YARA development approach

As part of our initial approach, we started to collect technical facts:

1. What do legitimate AnyDesk files look like?
2. What code signing certificates are used?



When looking at trusted copies of AnyDesk, the executables signed before `9 November 2023` were signed with a certificate from `philandro Software GmbH` (Serial number: `0d:bf:15:2d:ea:f0:b9:81:a8:a9:38:d5:3f:76:9d:b8`), the company's original name. After this time, AnyDesk started signing copies of their software with a new certificate with a serial number of `0a:81:77:fc:d8:93:6a:91:b5:e0:ed:df:99:5b:0b:a5`.

Potential detection: AnyDesk certificate used

To help equip hunt teams to answer quickly, "Is this used in our network?" the first rule we wrote looks for the serial numbers from AnyDesk's code signing certificates. This signature will match legitimate copies of AnyDesk and any potentially malicious files:

```
import "pe"

rule AnyDesk_certificates
{
    meta:
        author = "Silas Cutler (silas@stairwell)"
        description = "Detection for PE files with AnyDesk certificates"
        date = "2024-02-02"

    condition:
        uint16(0) == 0x5a4d and
        for any i in (0 .. pe.number_of_signatures): (
            pe.signatures[i].serial ==
"0a:81:77:fc:d8:93:6a:91:b5:e0:ed:df:99:5b:0b:a5" or
            pe.signatures[i].serial ==
"0d:bf:15:2d:ea:f0:b9:81:a8:a9:38:d5:3f:76:9d:b8"
        )
}
```

Potential detection: AnyDesk certificate used, but unrelated PE info

If the certificates were stolen and are used in the future to sign malicious executables, we can use the certificate serial numbers as a starting point. An easy starting point is to look for instances in which the files are signed, but the PE metadata does not match legitimate AnyDesk executables.

```
import "pe"

rule AnyDesk_certificates_invalid_pe_data
{
    meta:
        author = "Silas Cutler (silas@stairwell)"
        description = "Detection for PE files with AnyDesk certificates that do not
contain AnyDesk in the company name"
        date = "2024-02-02"

    condition:
        uint16(0) == 0x5a4d and
        for any i in (0 .. pe.number_of_signatures): (
            ( pe.signatures[i].serial ==
"0a:81:77:fc:d8:93:6a:91:b5:e0:ed:df:99:5b:0b:a5" or
            pe.signatures[i].serial ==
"0d:bf:15:2d:ea:f0:b9:81:a8:a9:38:d5:3f:76:9d:b8")
            and not pe.version_info["CompanyName"] icontains "AnyDesk"
        )
}
```

Potential detection: AnyDesk .NET

While reviewing the results of the previous two rules, we identified several malicious files (one example is SHA256 hash: `4e10c6fe5d0f656aab6d41c6a359bdbf658cafad4866583c8872ed60ed3018ed`) written in .NET bearing the AnyDesk certificate. As Anydesk is not written in .NET, signed files may be worth investigating.

```
import "pe"
import "dotnet"


rule AnyDesk_cert_and_DOTnet
{
    meta:
        author = "Silas Cutler (silas@stairwell)"
        description = "Detection for PE files with AnyDesk certificates and written
in .NET"
        date = "2024-02-02"

    condition:
        uint16(0) == 0x5a4d and
        for any i in (0 .. pe.number_of_signatures): (
            pe.signatures[i].serial ==
"0a:81:77:fc:d8:93:6a:91:b5:e0:ed:df:99:5b:0b:a5" or
            pe.signatures[i].serial ==
"0d:bf:15:2d:ea:f0:b9:81:a8:a9:38:d5:3f:76:9d:b8"
        )
        and dotnet.number_of_streams > 0
}
```

One of the most powerful features of the Stairwell platform is the ability for users to leverage YARA to dynamically hunt through their environment and through our sample feeds. Whether it's hunting for malware or tracking software, we want to enable our users to stay one step ahead of attackers.

Stairwell customers and users can find copies of these rules under the Stairwell Research ruleset.

Check out the Stairwell platform

Meet with the Stairwell team to learn how we can help your security team.

Get in touch