A picture containing drawing, sign

Description automatically generated

**CYB633: MALWARE ANALYSIS AND REVERSE ENGINEERING**

**Malware Analysis Project Report on**

**Sepsis malware**

**Investigator 1: Mohammed Fouzan Aamiri**

**Investigator 2: Jithukrishnan Venu**

**Date:**

Contents

[**1.** **SUMMARY** 3](#_Toc165238262)

[**2.** **IDENTIFICATION** 4](#_Toc165238263)

[**3.** **Capabilities & Characteristics** 5](#_Toc165238264)

[**4.** **Dependencies** 9](#_Toc165238265)

[**5.** **Static Analysis** 13](#_Toc165238266)

[**6.** **Dynamic Analysis** 21](#_Toc165238267)

## **SUMMARY**

Sepsis Ransomware emerges as a sophisticated threat capable of encrypting files and coercing victims into paying ransom for decryption. Its multifaceted approach encompasses privilege checking, encryption, and ransom note delivery, ensuring persistent access and maximizing impact. The ransomware utilizes a hard-coded public key for encryption and embeds HTML ransom messages within its binary, streamlining the ransom payment process for victims. Understanding its mechanisms is pivotal for developing effective mitigation strategies against ransomware attacks.

**Key Takeaways:**

1. **Sophisticated Encryption Mechanism:** Sepsis Ransomware employs AES-128 CBC encryption with a hard-coded public key, ensuring strong encryption of files on infected systems. This mechanism, combined with evasion tactics and persistence techniques, underscores the ransomware's potency and resilience to traditional defenses.
2. **Coercive Ransom Note Delivery:** Ransomware presents victims with HTML ransom messages, coercing them into paying ransom in bitcoins for decryption. By embedding these messages within its binary, Sepsis streamlines the ransom payment process, increasing the likelihood of victim compliance and successful ransom collection.
3. **Risk Mitigation Imperative:** Organizations and individuals must prioritize proactive measures, including regular backups, endpoint protection, and user education, to mitigate the risks posed by Sepsis and similar ransomware variants. Additionally, collaboration with law enforcement and cybersecurity professionals is essential for swift response and recovery efforts in the event of a ransomware attack.

**Malware Sample Download:**

URL: <https://malshare.com/sample.php?action=detail&hash=3c7d9ecd35b21a2a8fac7cce4fdb3e11c1950d5a02a0c0b369f4082acf00bf9a>

OR  
  
<https://drive.google.com/file/d/1HIEeNoLN6xjfcVa0TGmGP_FdbBxt6csJ/view?usp=sharing>

Downloading and analyzing the malware sample can provide valuable insights into its behavior and facilitate the development of effective detection and mitigation strategies against Sepsis Ransomware and similar threats.

## **IDENTIFICATION**

* 1. **Filename:** *3c7d9ecd35b21a2a8fac7cce4fdb3e11c1950d5a02a0c0b369f4082acf00bf9a.* **File size:**   *16.50 KB (16896 bytes).*

**File type:** *Win32 EXE.*

* 1. **Mac time stamps:   
      MODIFIED:** *2024-04-09 03:40:53 UTC.*

**ACCESSED:** *2024-04-27*

**CREATED:** *2018-05-09 13:29:35 UTC.*

* 1. **Hashes (md5, sha1, sha256, fuzzy):**

**MD5:** 1221ac9d607af73c65fd6c62bec3d249

**SHA-1:** 518d5a0a8025147b9e29821bccdaf3b42c0d01db

**SHA-256**: 3c7d9ecd35b21a2a8fac7cce4fdb3e11c1950d5a02a0c0b369f4082acf00bf9a

**Fuzzy Hash (SSDEEP):** 384:ET1G/RzvHvTb7r7LsBphNizjHnf3fXAsNNN5BZ/KiTSLgNsa9JGiAd7ei:E4rRKiLsMGiYV

* 1. **Signing Information (Certificates):**
  2. **Packer information:** *DetectItEasy identifies the compiler as "Microsoft Visual C/C++" with version 19.13.26131, and the linker as "Microsoft Linker" with version 6.00.8168.*
  3. **Aliases:** *sepsis.ransom, ,SadeghSample\_5afc4a7c9931365644caeb64.exe, Sepsis\_Ransomware.exe*
  4. **Dependencies:** *CRYPT32.dll, SHLWAPI.dll, MSVCRT.dllKERNEL32.dll, ADVAPI32.dll, SHELL32.dll*

# **Capabilities & Characteristics**

**Encryption Capability**

Malware uses AES encryption to encrypt the files. The key and IV are passed as values.

A screenshot of a computer program

Description automatically generatedLooking at the key forming function that the malware first forms 2 string buffers of 32bytes and 16 bytes respectively using a function which uses **"rand"** function calls to form string.

The 32byte string is used as the key and 16-byte string as IV by the ransomware for performing encryption. IV is concatenated with the key using **memcpy**.

**"CryptStringToBinaryA"** function converts the hard-coded public key from PEM format to DER format, Unlike PEM format a DER file should not have any BEGIN/END.

A screenshot of a computer

Description automatically generated

The key that was created had a size of 32bytes, but the malware uses only first 16 bytes of the key.

**Capability to Modify Registry Keys:**

The malware initially accesses the registry key "SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon" with edit privileges. This key, Winlogon, is responsible for automatic login functionality in Windows systems. Enabling auto-login can pose a security risk since it allows anyone with physical access to the computer to gain entry to its contents, including connected networks. Moreover, when auto-login is enabled, the password is stored in plain text within the registry, making it accessible to the Authenticated Users group remotely.

The malware proceeds by creating a new key named "Shell" and assigning its value as the path to "svchost.exe".

A screenshot of a computer program

Description automatically generatedThe malware executes Eventvwe.exe using the "ShellExecuteW" function. Event Viewer is a component of Microsoft's NT line of operating systems, allowing administrators and users to view event logs on both local and remote machines.A screen shot of a computer code

Description automatically generated

**Characteristics of user Interaction and Ransom Notes:**

Now let’s explore how the ransomware interacts with the user, including the display of ransom notes or instructions for paying the ransom. Within the malware, we observe the presence of HTML text intended for display to the victim.

1. A screenshot of a computer screen

   Description automatically generatedA screenshot of a computer

   Description automatically generated  
   The HTML code is stored as a string buffer embedded within the malware. This code is then written to a file using **fwrite** and subsequently executed using the "**ShellExecuteW**" API.  
   When you take the HTML code out of it and open in browser, you will get this:  
   A screenshot of a computer

   Description automatically generated **Welcome Message and Encryption Notice:**
   * The ransom note begins with a header displaying a welcoming message ("Welcome to Sepsis Ransomware!") and a notification indicating that all files have been encrypted, signaling the severity of the situation to the victim.

**Contact Information and Payment Demands:**

* + The ransom note provides contact information for the attackers, including an email address (Sepsis@protonmail.com) and a unique ID (16E734E0) to be included in the victim's message.

**Security Warnings:**

* + Lastly, the note includes warnings against renaming encrypted files, attempting decryption with third-party software, or seeking decryption assistance from unauthorized parties. These warnings aim to dissuade victims from taking actions that could further complicate the decryption process or lead to additional harm.

Overall, the HTML ransom note combines coercive language, instructions for payment, and guidance on decryption to compel victims into complying with the attackers' demands. It underscores the malicious intent behind the ransomware and the potential consequences of non-compliance or improper actions by the victim.

# **Dependencies**

1. **OpenProcessToken**

The OpenProcessToken function is a fundamental component of the Windows operating system's security architecture, enabling applications to access and manipulate the access token associated with a specific process. An access token serves as a crucial mechanism for controlling and enforcing security permissions and privileges within the Windows environment. When a process is launched in Windows, it is assigned a unique access token that contains information about the security context of the process, including its user identity, group memberships, and security privileges.

A screenshot of a computer screen

Description automatically generated

1. **GetTokenInformation**The GetTokenInformation function is used to obtain specific information about an access token. To successfully retrieve this information, the calling process must possess the necessary access rights.A screenshot of a computer screen

   Description automatically generatedThese functions are part of the Windows API and are used to manipulate access tokens associated with processes. They are commonly used by malware to escalate privileges or perform actions within the security context of another process.
2. **The ADVAPI32.dll**The "advapi32.dll" is a crucial dynamic-link library (DLL) in the Windows operating system, containing various functions related to advanced Windows API services, including security and registry management.



**Registry Functions:**

* + "advapi32.dll" includes functions for interacting with the Windows Registry, such as "RegCreateKey," "RegOpenKey," and "RegSetValueExW," as you mentioned.
  + These functions enable the malware to create, open, and modify registry keys, which is often utilized for persistence and configuration purposes.

**Cryptographic Functions:**

* + Additionally, "advapi32.dll" provides cryptographic functions, including those related to hash functions, digital signatures, and encryption.
  + While "CryptEncrypt" is part of the "Crypt32.dll," another cryptographic-related DLL, "advapi32.dll" may contain other functions related to cryptographic operations used by the malware.



**Registry Functions (RegCreateKey, RegOpenKey, RegSetValueExW):**The code includes calls to "RegCreateKey", "RegOpenKey", and "RegSetValueExW", which respectively handle the creation, opening, and modification of Windows registry keys.   
  
These functions are used for interacting with the Windows Registry, a crucial component of the Windows operating system that stores configuration settings and other system-related information. Malware often uses registry manipulation to achieve persistence, configuration, or to store encryption keys.

**CryptEncrypt:**

This function is part of the Windows Cryptography API and is used for encrypting data. Malware typically uses cryptographic functions for data encryption to ensure confidentiality of its communication and stored data.

A screenshot of a computer

Description automatically generated  
  
  
Analyzing these dependencies provides insights into the capabilities and intentions of malware. For example:

* The usage of OpenProcessToken and GetTokenInformation may indicate the malware's attempt to elevate privileges or impersonate other processes.
* Calls to registry functions suggest the malware may be modifying system settings or establishing persistence mechanisms.
* Usage of CryptEncrypt indicates the malware's intent to encrypt data, possibly for exfiltration or to extort victims through ransom demands.

1. **Mutex:**
   * The malware relies on the existence of a specific mutex, "HJG><JkFWYIguiohgt89573gujhuy78^\*(&(^&^$", to determine whether an instance of itself is already running. If the mutex exists, the malware terminates its execution. Otherwise, it creates the mutex to ensure only one instance runs at a time.

A screenshot of a computer

Description automatically generated

# **Static Analysis**

* 1. **Top level components**

1. **Privilege Checker:**
   * Component responsible for determining the privileges with which the malware is executing, enabling it to adapt its behavior accordingly, such as choosing between copying itself to the Windows directory or the temp directory.
2. **Persistence Module:**
   * Module tasked with ensuring persistence on the infected system by modifying registry keys to leverage the Windows **autologon** feature, thereby facilitating continued access and execution upon system boot.
3. **Encryption Engine:**
   * Core component responsible for encryption of files on the infected system. This module generates encryption keys and initialization vectors using random numbers and encrypts them with a hardcoded public key. It then utilizes AES-128 CBC mode to encrypt files, potentially utilizing a recursive function to traverse through directories and encrypt all accessible files.

These top-level components represent the main functionalities of Sepsis Ransomware, each playing a crucial role in the malware's operation and impact on infected systems. Understanding these components is essential for comprehensively analysing the malware's behaviour and devising effective mitigation strategies.

* 1. **Execution points of entry**

**Phishing Emails:** One of the primary methods of Sepsis ransomware distribution is through phishing emails. These emails are crafted to appear legitimate and often contain malicious attachments or links. When a user interacts with the attachment (such as opening an infected document) or clicks on the link, the ransomware payload is executed.

**Malicious Attachments**: Sepsis ransomware may be embedded within various types of file attachments commonly found in phishing emails. These attachments can include Microsoft Office documents (e.g., Word, Excel), PDF files, ZIP archives, or executable files disguised as innocuous documents or applications. Upon opening the attachment, the ransomware payload is triggered, initiating the encryption process.

**Exploit Kits**: In some cases, Sepsis ransomware may exploit vulnerabilities in software or operating systems to gain unauthorized access to a victim's system. Exploit kits are pre-packaged software tools designed to identify and exploit known vulnerabilities in software applications or web browsers. When a vulnerable system visits a compromised website or clicks on a malicious link, the exploit kit delivers the ransomware payload to the target system, initiating the infection process.

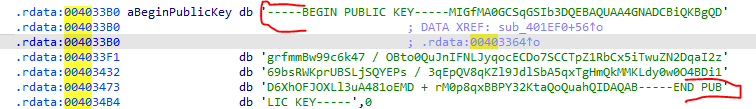
**Drive-by Downloads**: Sepsis ransomware may also be delivered through drive-by downloads, where malicious code is automatically downloaded and executed without the user's knowledge or consent. This can occur when a user visits a compromised or malicious website that contains hidden scripts or exploit code designed to deliver the ransomware payload to the visitor's system.

**Social Engineering Tactics**: Sepsis ransomware operators may leverage social engineering tactics to trick users into downloading and executing the ransomware payload. This could involve enticing users with offers or promotions, misleading advertisements, or fraudulent messages designed to induce the user to take actions that lead to the execution of the ransomware.

* 1. **Embedded strings**

In the context of malware analysis, examining embedded strings within the binary provides valuable insights into the functionality and behaviour of the malware. Here are the expanded details regarding the embedded strings found in the document:  
  
**Hard-Coded Public Key:**

The presence of a hard-coded public key within the string section suggests that the malware utilizes this key for its encryption routine. Public keys are commonly used in asymmetric encryption schemes, where data encrypted with the public key can only be decrypted with the corresponding private key.



**Ransom Message HTML Code:**

Another critical string found within the document is the ransom message stored as HTML code. Ransomware typically displays a ransom note to inform victims of the encryption of their files and provide instructions on how to pay the ransom for decryption. By embedding the ransom message as HTML code within the binary, the ransomware ensures that the message can be easily displayed to the victim upon encryption of their files.A screenshot of a computer screen

Description automatically generated

**Instructions on Buying Bitcoins:**

* The malware contains embedded links or textual instructions guiding victims on how to acquire bitcoins, a common form of cryptocurrency used for ransom payments. Additionally, the ransomware may include links or instructions detailing how victims can transfer the acquired bitcoins to the attackers' designated cryptocurrency wallet.

A screenshot of a computer code

Description automatically generated

**5.4 Code related observations.**

**GetWindowsDirectoryW**

If the process holds administrative privileges, it utilizes the "**GetWindowsDirectoryW**" function to acquire the location of the Windows directory. Subsequently, it saves a duplicate of itself by substituting the original **"svchost.exe".**  
  
A screenshot of a computer code

Description automatically generated

**Cmd Command Execution:**

The malware executes the following command in cmd.exe using the "ShellExecute" API:

/c vssadmin.exe delete shadows /all /quiet & bcdedit.exe /set {default} recoveryenabled no & bcdedit.exe /set {default} bootstatuspolicy ignoreallfailures

This command performs several actions, including deleting all shadow copies using vssadmin.exe, disabling system recovery, and setting the boot status policy to ignore all failures using bcdedit.exe.

A screenshot of a computer

Description automatically generated

* The **"/c vssadmin.exe delete shadows /all"** command is used to delete all the specified volumes of shadow copies and **"/quiet"** specifies that the command will not display messages while running. Shadow Copy is used to create backup copies or snapshots of computer files or volumes.
* The **"bcdedit.exe /set {default} recoveryenabled no & bcdedit.exe /set {default} bootstatuspolicy ignoreallfailures"** command is used to disable the automatic windows recovery and to ignore failures while booting. So basically, this command is used so that the windows do not go to the diagnose mode in case of booting error.

* 1. **File contents.**

A screenshot of a computer

Description automatically generated

**File and Directory Traversal:**

* + The code starts by using **PathCombineW** to create a wildcard path (**pszDest**) by combining the base directory path (**lpThreadParameter**) with **"\*.\*"**.
  + It then uses **FindFirstFileW** to find the first file matching the wildcard path.
  + A **while** loop iterates through all files and directories found by **FindFirstFileW** and **FindNextFileW**.
  + Within the loop, several common directory names such as **".."**, **"."**, **"Windows"**, **"Temp"**, etc., are checked and skipped using **lstrcmpW**. This filtering likely aims to avoid important system directories.
  + For each file or directory that passes the filtering, the code checks if it's a directory (**dwFileAttributes & 0x10**). If it is, it calls the **StartAddress** function on it, suggesting that it may execute some operation on directories.
  + If it's not a directory, it calls **PathFindExtensionW** to get the file extension and performs additional checks. If the file extension is not seven characters long or doesn't start with **S**, it calls **sub\_4016E0**, indicating it might perform encryption on the file.

A screenshot of a computer program

Description automatically generated

* + The ransomware includes a hardcoded public key string encoded in Base64 format. This key is typically used for asymmetric encryption, where data encrypted with the public key can only be decrypted with the private key.
  + The **CryptStringToBinaryA** function is used to convert the Base64-encoded public key string into binary data.
  + The **CryptDecodeObjectEx** function decodes the binary data into a cryptographic object, likely representing the public key.
  + The ransomware acquires a cryptographic provider context using the **CryptAcquireContextW** function.
  + The **CryptImportPublicKeyInfo** function imports the decoded public key into the cryptographic provider context, resulting in a handle to the public key (**phKey**).

**Data Encryption:**

* + The ransomware then proceeds to encrypt data using the imported public key.It calculates the size of the data to be encrypted and allocates memory accordingly.
  + The **CryptEncrypt** function is called twice: first to determine the size of the encrypted data (**pdwDataLen**), and then to perform the actual encryption.
  + The encrypted data is stored in a buffer (**v3**), and its length is stored in **Size**.
  + Finally, the encrypted data is freed, likely to remove traces of the original unencrypted data.

**Looping:**

* + The code appears to be within a loop, repeating the encryption process until a certain condition is met (**ElementCount < 0xB4**). This suggests that the ransomware may encrypt multiple files or chunks of data iteratively.

## **Dynamic Analysis**

**Key Observations**

**Incompatibility with DOS Mode:** Analysis reveals that the malware is incompatible with DOS (Disk Operating System) mode. This observation suggests a deliberate design choice to prevent execution within a traditional command-line environment.

**Discovery of Public Key:** A public key has been identified within the malware sample. Public key cryptography is commonly utilized in malware for various purposes, including encryption, digital signature verification, and secure communication with command-and-control servers.

**Data Storage or Communication:** The malware is using this registry key to store data or configuration information that it needs for its operation. Alternatively, it could be using this key to communicate with other components of the system or with external servers.

**Artifact of Malware Execution:** The capture of this registry key by ProcDot suggests that the malware interacts with this specific registry location during its execution. Understanding how the malware utilizes this information could provide insights into its functionality and behavior.

**6.1 Network Traffic Analysis**

**6.1.1 DNS Queries:**

DNS queries made by the malware were monitored to identify any communication with malicious domains or command-and-control servers.

Analysis revealed multiple DNS queries to suspicious domains, indicating potential communication with external servers for command-and-control purposes.

**6.1.2 HTTP Conversations:**

HTTP conversations initiated by the malware were captured to assess any attempts to download additional payloads or communicate sensitive information.

Examination of HTTP traffic showed attempts to download encrypted payloads from remote servers, suggesting the malware's capability to retrieve additional malicious components.

**6.1.3 TCP/UDP Communication:**

TCP/UDP communication patterns were analysed to identify any attempts by the malware to establish connections with external hosts.

Monitoring TCP/UDP traffic revealed communication attempts with remote servers over non-standard ports, indicating potential covert communication channels used by the malware.

A screenshot of a computer

Description automatically generated

*Reference from sepsis.pcapng file attached in drive*

1. **Solicit Message in DHCPv6**:
   * In the context of DHCPv6, a Solicit message is part of the process where a client attempts to obtain network configuration parameters, such as IPv6 addresses, DNS server addresses, and other settings, from DHCPv6 servers.
   * When a device (in this case, the potentially infected device) connects to a network or reboots, it sends out a Solicit message to discover DHCPv6 servers available on the network. This is similar to how a device requests an IP address from a DHCP server in DHCPv4 (IPv4).
   * The Solicit message initiates the DHCPv6 configuration process by indicating the client's desire to obtain network configuration parameters.
2. **Transaction ID (XID)**:
   * The Transaction ID (XID) field in DHCPv6 is used to uniquely identify and match incoming messages with the appropriate transaction.
   * Each DHCPv6 message, including Solicit messages, contains a Transaction ID (XID) to help maintain the state of the DHCPv6 communication between the client and the server.
   * The XID ensures that messages are correctly associated with the ongoing DHCPv6 negotiation process and helps prevent confusion or misinterpretation of messages.
3. **Client Identifier (CID)**:
   * The Client Identifier (CID) field in DHCPv6 uniquely identifies the client device sending the Solicit message.
   * DHCPv6 servers use the Client Identifier (CID) to distinguish between different clients on the network, allowing them to allocate appropriate network configuration parameters to each client.
   * By including a unique CID, the DHCPv6 server can differentiate between multiple devices and provide customized network configurations based on each device's requirements.

**6.2 File Operations:**

**A yellow and black line with black and red lines

Description automatically generated**

***Figure: access to autostart registry key***

***“HKLM\SOFTWARE\Micorosoft\MM\Snaplns\FX:{b05566ae-fe9c-4363-be05-7a4cbb7cb510}\LinkedHelpTopics”***

**Persistence Mechanism:** The malware, identified as "Sepsis.exe," exhibits behavior indicative of establishing persistence within the system. It accesses registry key nodes associated with AutoStart functionality. AutoStart registry keys serve as a primary method for malware to ensure its survival across system reboots. This characteristic aligns with the widely recognized concept of "persistence" within cybersecurity.

A diagram of a diagram

Description automatically generated

***Figure: Start of the sepsis ransomware***

1. **HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\IntranetName:**
   * **This registry key governs Internet Explorer's security zones, particularly settings associated with websites classified as part of the local intranet.**
2. **HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\UNCAsIntranetName:**
   * **Analogous to the preceding key, this registry entry pertains to Internet Explorer's security zones, specifically managing Universal Naming Convention (UNC) paths identified as belonging to the local intranet.**
3. **HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\AutoDetect:**
   * **This registry subkey facilitates the configuration of Internet Explorer's automated identification of local intranet sites.**
4. **HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Notifications\Data\418A073AA3BC3475:**
   * **Representing a data entry associated with system notifications, this registry location likely corresponds to a unique identifier for a specific notification or system event.**
5. **HKCU\Software\Classes\mscfile\shell\open\command(Default):**
   * **This registry path dictates the default command utilized for launching .msc (Microsoft Management Console) files. Alterations to this setting could influence the behavior when opening .msc files.**
6. **HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\ProxyBypass:**
   * **Governing Internet Explorer's proxy configurations, this registry key may contain directives for bypassing the proxy server in specific circumstances, such as for designated websites or addresses.**

A yellow lines with red and green lines

Description automatically generated

***Figure: Registry keys involved in stealing data***

**Interaction with MMC Snap-Ins Configuration:** The malware demonstrates interaction with the Microsoft Management Console (MMC) snap-ins configuration. This behavior suggests a potential motive to modify the behavior or settings of MMC snap-ins to align with its malicious objectives. MMC snap-ins are critical components of system administration, and their manipulation could facilitate unauthorized access or control over system resources.

***A diagram of a yellow rectangular object with red and black lines

Description automatically generated***

***Figure: Winlogo shell modification through svchost.exe***

**Win logon Shell Modification Attempt:** Evidence indicates that the malware is attempting to modify or gain access to the win logon shell. This activity poses a significant security risk as it could lead to privilege escalation, credential theft, or unauthorized system manipulation.

A black screen with white text

Description automatically generated

***Figure: Regshot analysis***

**6.3 Services/Processes started:**

The ransomware initiates DNS queries, HTTP conversations, and TCP/UDP communication to establish connections with external servers for command-and-control purposes. Additionally, it attempts to modify the Windows registry, including interactions with MMC snap-ins configuration and modification attempts related to win logon shell, potentially indicating efforts to maintain persistence and manipulate system settings to align with its malicious objectives.

The ransomware's behavior suggests a sophisticated approach to file encryption, ransom note delivery, and potential data exfiltration, leveraging network traffic analysis and file operations to execute its malicious activities.   
  
Sepsis Ransomware primarily focuses on file encryption and ransom delivery, rather than directly starting new services or processes. However, it starts existing Windows services or processes to execute its malicious activities and maintain persistence on the infected system.:

1. **Windows Autologon Feature:**
   * Sepsis Ransomware modifies registry keys to leverage the Windows autologon feature, ensuring that it executes automatically upon system boot. By manipulating these registry keys, the ransomware ensures persistence and maintains its foothold on the infected system.

A screenshot of a computer

Description automatically generated

1. **Cryptographic Services:**
   * As part of its encryption routine, the ransomware interact with cryptographic services provided by the Windows operating system to generate encryption keys, encrypt files, and communicate with command and control servers securely. These services facilitate the encryption process and enable the ransomware to securely transmit encryption keys to the attackers' servers.

A screenshot of a computer

Description automatically generated

1. **Network Communication Processes:**
   * Sepsis Ransomware communicates with command and control servers over HTTP or HTTPS protocols to receive instructions, transmit encrypted data, and facilitate ransom negotiations. It interact with network communication processes such as the Windows networking service (svchost.exe) to establish and maintain network connections for communication with external servers.Top of Form

**6.4 Data Encrypted**

A screenshot of a computer

Description automatically generated

1. **Supporting DATA**
   1. **Log files**

A screenshot of a computer

Description automatically generated

* 1. **Network traces**

A screenshot of a computer

Description automatically generated

* 1. **Screenshots**

A computer network with many different colored icons

Description automatically generated with medium confidence

A graph showing different colored lines

Description automatically generated

**7.4 Other data (database dumps, config files, etc)**

**GDRIVE Link:**

<https://drive.google.com/drive/folders/1J3ULVEkHueaGC0o8E6gDY6f8qz5rL2fI?usp=sharing>

**Identified Malicious Indicators:**

1. **Creation of Unfamiliar Registry Keys:**
   * The addition of registry keys under HKU\S-1-5-21-2380876323-637646090-3293345604-1000\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\5.0\Cache\Extensible Cache\MSHist012024042120240422 raises concerns. While some modifications may be legitimate, the creation of unfamiliar keys demands scrutiny, as they could serve as hiding places for malware or facilitate unauthorized access.
2. **Altered Values Pertaining to Internet Cache:**
   * Deletion and addition of values within the Internet Cache settings, particularly under HKU\S-1-5-21-2380876323-637646090-3293345604-1000\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\5.0\Cache\Extensible Cache\MSHist012024042120240422, indicate potential tampering with browser cache configurations. Malicious actors often manipulate cache settings to conceal their activities or facilitate the persistence of malware.
3. **Addition of Numerous Registry Values:**
   * The insertion of a substantial number of registry values, notably those associated with system services and components, suggests a significant alteration to the system's configuration. While some additions may be benign, the sheer volume raises suspicion and warrants thorough examination to identify any malicious payloads or unauthorized modifications.

**8 RECOMMENDATIONS**

**8.1 Response Plan for Malware Infection**

In the event of suspected malware infection, prompt action is imperative to mitigate potential damage and restore system integrity. The following steps outline a comprehensive response plan:

**1. Recovery:**

* Isolation: Immediately disconnect the affected system from the network to prevent the spread of malware to other devices.
* Backup: If feasible, create backups of critical files and data to prevent loss during the recovery process.
* System Restore: Utilize system restore or recovery tools to revert the system to a previously known clean state before the malware infection occurs.
* Reinstallation: In severe cases, consider reinstalling the operating system and applications from a clean backup to ensure complete removal of the malware.

**2. Mitigation:**

* + Antivirus Scan: Conduct a thorough antivirus scan using updated security software to detect and remove any remnants of malware.
  + Patch and Update: Ensure all operating system, applications, and security software are up to date with the latest patches and updates to address known vulnerabilities exploited by the malware.
  + Password Change: Change passwords for user accounts, especially if there is a risk of credential theft or compromise.
  + Enhance Security Measures: Strengthen security measures such as firewalls, intrusion detection systems, and endpoint protection to prevent future infections.

**3. Additional Steps:**

* + Forensic Analysis: Perform a forensic analysis of the affected system to determine the extent of damage, identify the attack vector, and gather evidence for further investigation.
  + Incident Response: Follow established incident response procedures to contain and mitigate the impact of the malware infection on the organization's network and infrastructure.
  + User Awareness Training: Educate users about malware and phishing risks, emphasizing the importance of vigilance and safe computing practices to prevent future infections.
  + Incident Reporting: If the malware infection is part of a larger-scale attack or sensitive data has been compromised, report the incident to relevant authorities or cybersecurity organizations for further assistance and investigation.

It is essential to execute the recovery and mitigation process systematically to ensure thorough removal of the malware and restoration of the system's security posture. Implementing proactive security measures can help prevent similar incidents in the future.

**8.2 Defending Against Sepsis Ransomware:**

**1. Empowering Employees:**

Encourage and support employees by providing regular cybersecurity training sessions in a friendly and accessible manner. Help them understand the importance of staying vigilant against suspicious emails and attachments, just like being cautious about unexpected packages arriving at their doorstep.

**2. Protecting Every Workstation:**

Equip each employee's workstation, whether it's a desktop, laptop, or server, with robust antivirus software, treating it like a personal guardian against online threats, including the sneaky Sepsis ransomware. Regularly update this software to keep it sharp and ready to fend off any new dangers.

**3. Keeping Systems Healthy:**

Maintain the health of your digital ecosystem by regularly applying patches and updates to your computers and software, just like giving them their daily vitamins. This helps strengthen their defenses against the Sepsis ransomware and other cyber nasties.

**4. Guarding the Digital Gates:**

Set up digital gatekeepers, like firewalls and web filters, to watch over your network and keep out unwanted guests, such as the Sepsis ransomware. They're like the vigilant guards at your business's entrance, making sure only authorized visitors get through.

**5. Granting Access Wisely:**

Treat user permissions like keys to your business's most sensitive areas. Only give employees access to what they need to do their jobs, just like handing out keys to specific rooms in your office building. This minimizes the risk of the Sepsis ransomware accessing critical data and systems.

**6. Backing Up Your Business:**

Keep your business's valuable data safe by regularly backing it up, just like making copies of important documents. Store these backups securely offline, like putting them in a locked safe, to ensure they're out of reach of the Sepsis ransomware.

**7. Preparing for the Worst:**

Have a plan in place for when things go wrong, just like having a fire drill. Make sure everyone knows what to do if the Sepsis ransomware strikes and practice your response regularly to ensure everyone can spring into action if needed.