Threat Analysis Report: Inside the Destructive PYSA Ransomware

cybereason.com/blog/threat-analysis-report-inside-the-destructive-pysa-ransomware

Written By
Cybereason Global SOC Team

September 27, 2021 | 10 minute read

The Cybereason Global Security Operations Center (GSOC) issues Cybereason Threat Analysis reports to inform on impacting threats. The Threat Analysis reports investigate these threats and provide practical recommendations for protecting against them.

In this Threat Analysis Report, the GSOC investigates the PYSA ransomware. The PYSA ransomware came into awareness earlier this year when the Federal Bureau of Investigation (FBI) reported on the ransomware's increased activity and high damaging impact.

The threat actors behind PYSA deploy the ransomware as part of attack operations with high-stake targets, such as government authorities, educational institutions, and the healthcare sector. This Threat Analysis report focuses on the implementation of the PYSA ransomware and the ransomware's internal working principles when deployed on a compromised system.

What is PYSA Ransomware?

- **Human-Operated**: PYSA is a human-operated ransomware that does not have self-propagation capabilities. Threat actors manually deploy the PYSA ransomware as part of full attack operations. The PYSA ransomware operators typically gain initial access to target systems by compromising credentials or through phishing emails. Prior to the deployment of the ransomware, the malicious actors use publicly available and/or open-source tools for credential theft, stealthiness, privilege escalation, lateral movement, and more.
- **Hybrid Encryption Approach:** The PYSA ransomware is implemented in the C++ programming language and uses the open-source CryptoPP C++ library for data encryption. The ransomware encrypts data by combining the use of the Advanced Encryption Standard-Cipher Block Chaining (AES-CBC) and the Rivest, Shamir, Adleman (RSA) encryption algorithms. This is to maximize both encryption performance and security.
- **Double Extortion**: The PYSA ransomware operators use a double extortion tactic - if the victim refuses to pay for data decryption, the malicious actor threatens to leak the data or sell it for profit.
- **Detected and Prevented**: The Cybereason Defense Platform effectively detects and prevents the PYSA ransomware.
- **Cybereason Managed Detection and Response (MDR):** The Cybereason GSOC has zero tolerance towards attacks that involve ransomware, such as PYSA, and categorizes such attacks as critical, high-severity incidents. The Cybereason GSOC MDR team issues a comprehensive report to customers when such an incident occurs. The report provides an in-depth overview of the incident, which helps to scope the extent of compromise and the impact on the customer's environment. In addition, the report provides attribution information when possible as well as recommendations for mitigating and isolating the threat.

Introduction

PYSA is a new variant of the Mespinoza ransomware that first came to prominence in October 2019 when it infected large corporate networks. The French national computer emergency response team (CERT) reported in April 2020 that the PYSA ransomware has also targeted French local authorities. This has significantly raised the profile of this ransomware in the threat landscape. In March 2021, the FBI issued an alert stating that they have observed an increase in the PYSA ransomware targeting education institutions in 12 US states and the United Kingdom.

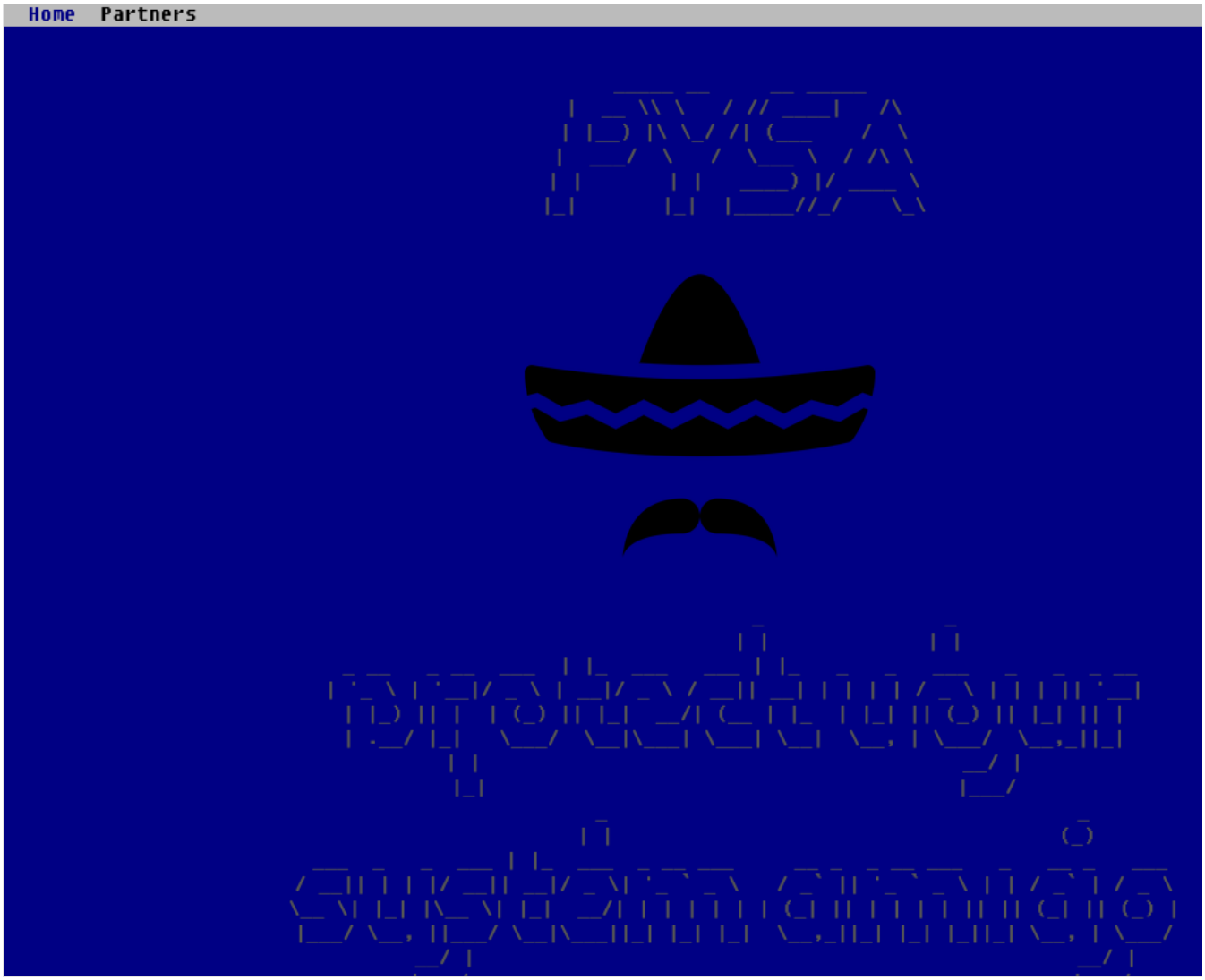
The operators of the PYSA ransomware have specifically targeted higher education, K-12 schools, and seminaries. In addition, the FBI reports on PYSA ransomware attacks targeting US and foreign government entities, private companies, and the healthcare sector since March 2020. In June 2021, the BlackBerry Threat Research and Intelligence SPEAR Team reported that it had observed the actors behind the PYSA ransomware conducting fully developed attack operations and deploying the ransomware at selected target organizations.

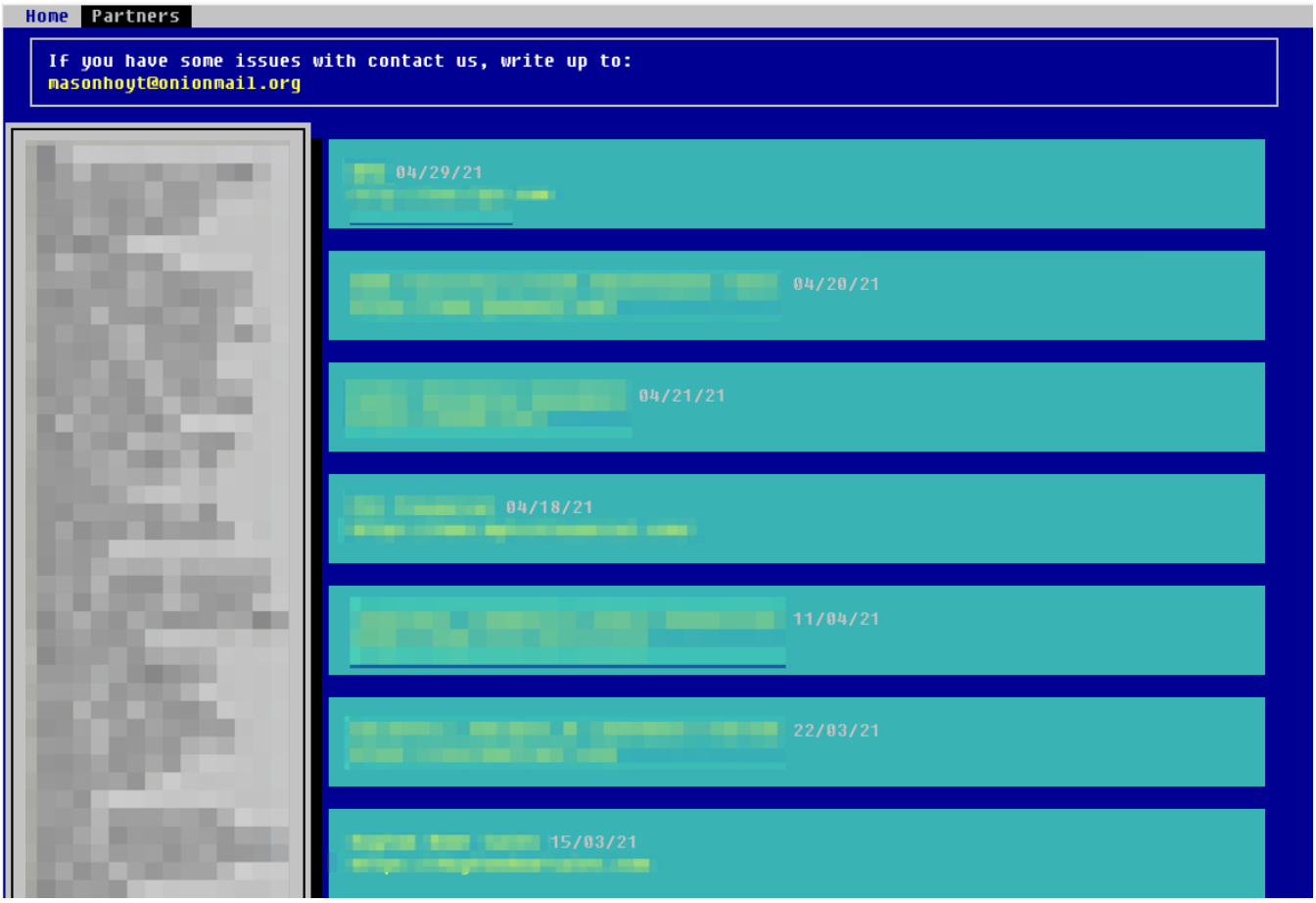
PYSA is a human-operated ransomware that does not have self-propagation capabilities. Threat actors manually deploy the PYSA ransomware as part of full attack operations. The FBI reports that the PYSA ransomware operators typically gain initial access to target systems through phishing email messages or by compromising credentials, such as brute-forcing Active Directory domain credentials or Remote Desktop Protocol (RDP) credentials.

Prior to the deployment of the PYSA ransomware on a compromised system, the malicious actors use publicly available and/or open-source tools for credential theft, stealthiness, privilege escalation, lateral movement, and so on. For example, they use the Advanced Port Scanner and the Advanced IP Scanner tools developed by Famatech Corp, which are port scanning and information gathering tools that enable users to discover and gather information on services running on network computers.

In addition, the ransomware operators use the tools PowerShell Empire, Koadic, PsExec, and Mimikatz for credential theft and lateral movement. Before deploying the PYSA ransomware, the actors execute PowerShell scripts that stop or remove system security mechanisms, such as Windows Defender. They also delete system restore snapshots and shadow copies so that victims cannot restore data encrypted by the ransomware.

Furthermore, the FBI reports that malicious actors use the WinScp tool for data exfiltration from victim systems before the data is encrypted. Also, the actors behind the PYSA ransomware use a double extortion tactic - if the victim refuses to pay for data decryption, the malicious actor threatens to leak the data online or sell it for profit:





Screenshot of the PYSA data leaks website

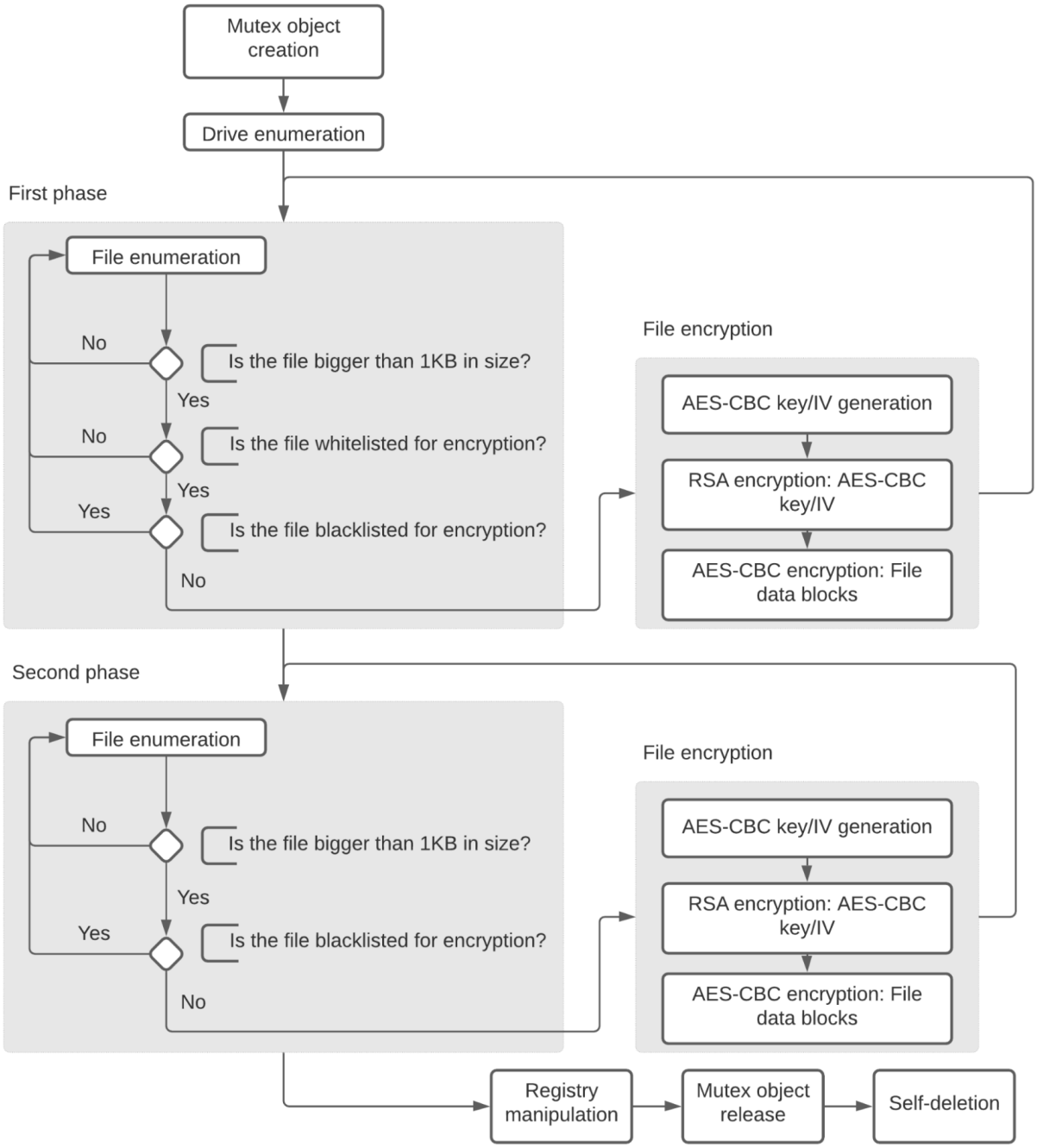
The operators of the PYSA ransomware communicate with their victims only via email. They refer to victims as “partners” and they do not use mechanisms typical for the currently trending Ransomware-as-a-Service (RaaS) business model, such as a ticketing system for communication with victims or online decryption services.

The PYSA ransomware is implemented in the C++ programming language and uses the open-source CryptoPP C++ library for data encryption. The ransomware encrypts data by applying a hybrid encryption approach that combines the use of the Advanced Encryption Standard-Cipher Block Chaining (AES-CBC) and the Rivest, Shamir, Adleman (RSA) encryption algorithms. This is to maximize both encryption performance and security.

The files that are encrypted by PYSA have the *.pysa* filename extension. The name PYSA may be derived from the *Protect your system amigo* slogan or from the Zanzibari coin with the same name. The *Protect your system amigo* slogan can be found in the ransom note that is left by the ransomware on compromised systems.

Pysa Ransomware Analysis

This section discusses the implementation and the operation of the PYSA ransomware. The following chart provides a summarizing overview of the operation of PYSA:



Summarizing overview of the operation of the PYSA ransomware

The PYSA ransomware process first detaches itself from the console, which closes the console. This allows the ransomware to operate without the console being a visual indicator of the ransomware's operation. The PYSA ransomware then creates a mutex object named *Pysa*. If this mutex object already exists, the ransomware terminates. This is to ensure that only one instance of the PYSA ransomware runs at a time:

```
FreeConsole();
if ( !OpenMutexA(0x1F0001u, 0, "Pysa") )
{
  v3 = CreateMutexA(0, 0, "Pysa");
```

Creation of a mutex object named Pysa

The PYSA ransomware then enumerates drives with a fixed media attached to the compromised system. These are drives for which the Windows API function *GetDriveTypeW* returns *0x3* (*DRIVE_FIXED*), such as hard disks. For each drive with a fixed media, the PYSA ransomware creates a process thread, in whose context the ransomware conducts file enumeration and encryption.

The PYSA ransomware does this in two phases. In the first phase, it encrypts files that are whitelisted for encryption, these are files that have one of the filename extensions that are hardcoded in the file that implements the ransomware. The following table lists the filename extensions of files that are whitelisted for encryption:

.doc	*.backupdb*	*.vrb*
.xls	*.bck*	*.win*
.docx	*.bkf*	*.pst*
.xlsx	*.bkup*	*.mdb*
.pdf	*.bup*	*.7z*
.db	*.fbk*	*.zip*
.db3	*.mig*	*.rar*
.frm	*.spf*	*.cad*
.ib	*.sql*	*.dsd*
.mdf	*.vhdx*	*.dwg*
.mwb	*.vfd*	*.pla*
.myd	*.avhdx*	*.pln*
.ndf	*.vmcx*	
.sdf	*.vmrs*	
.trc	*.pbf*	
.wrk	*.qic*	
.001	*.sqb*	
.acr	*.tis*	
.bac	*.vbk*	
.bak	*.vbm*	

Filename extensions of files that the PYSA ransomware encrypts in the first phase

In the second phase, PYSA encrypts the rest of the files stored on the drive and stores a *README.README* file in each directory on the drive. The *README.README* file contains the ransom note. The ransom note contains the following:

- The *Protect your system amigo* slogan, which the name PYSA may be derived from.
- Text informing the victims that the malicious actors have exfiltrated data from the compromised system and that they will expose this data to the public, or sell the data, if payment is not made. This is a double extortion tactic.
- A link to a data leaks website.
- A list of email addresses for communication with the attackers.

In both phases, the PYSA ransomware:

- Encrypts only files that are bigger than 1 KB in size.
- Does not encrypt files that are blacklisted for encryption. These are:
  - system-critical files, such as *pagefile.sys*, the Windows boot manager and files stored in system-critical directories, for example, *Windows*, *Boot*, and *System Volume Information*;
  - files that have one of the following filename extensions: *.exe*, *.dll*, *.search-ms*, *.sys*, *.README*, or *.pysa*.

PYSA does not encrypt the aforementioned files because encrypting system-critical files and files that have filename extensions typical for executable files (*.exe*, *.dll*, and *.sys*) renders the compromised system unbootable and unusable. In addition, the PYSA ransomware creates files itself with the filename extensions *.README* and *.pysa*. The encryption of these files means encrypting the ransom note and encrypting files already encrypted by PYSA:

```
Hi Company,

Every byte on any types of your devices was encrypted.
Don't try to use backups because it were encrypted too.

To get all your data back contact us:
DeborahTrask@onionmail.org
AlisonRobles@onionmail.org
NatanSchultz67@protonmail.com

Also, be aware that we downloaded files from your servers and in case of non-payment
we will be forced to upload them on our website, and if necessary, we will sell them on the darknet.
Check out our website, we just posted there new updates for our partners:
http://pysa[redacted].onion/

FAQ:

1.
   Q: How can I make sure you don't fooling me?
   A: You can send us 2 files(max 2mb).

2.
   Q: What to do to get all data back?
   A: Don't restart the computer, don't move files and write us.

3.
   Q: What to tell my boss?
   A: Protect Your System Amigo.
```

The ransom note left by the PYSA ransomware on compromised systems

Before encrypting a file, the PYSA ransomware first renames the file by appending the filename extension *.pysa* to the filename, for example, *test.txt* becomes *test.txt.pysa*. PYSA then encrypts the file by applying a hybrid encryption approach. This approach combines the use of the AES-CBC and the RSA encryption algorithms. This is to maximize both encryption performance and security.

The PYSA ransomware first encrypts a file with the symmetric encryption algorithm AES-CBC. AES-CBC is by design more performant but less secure than the RSA encryption algorithm. This algorithm relies on a symmetric encryption key and an initialization vector (IV) for encryption security. To compensate for this disadvantage of AES-CBC, the ransomware then encrypts the AES-CBC symmetric key and IV with the RSA encryption algorithm. The PYSA ransomware uses the CryptoPP C++ library for encryption.

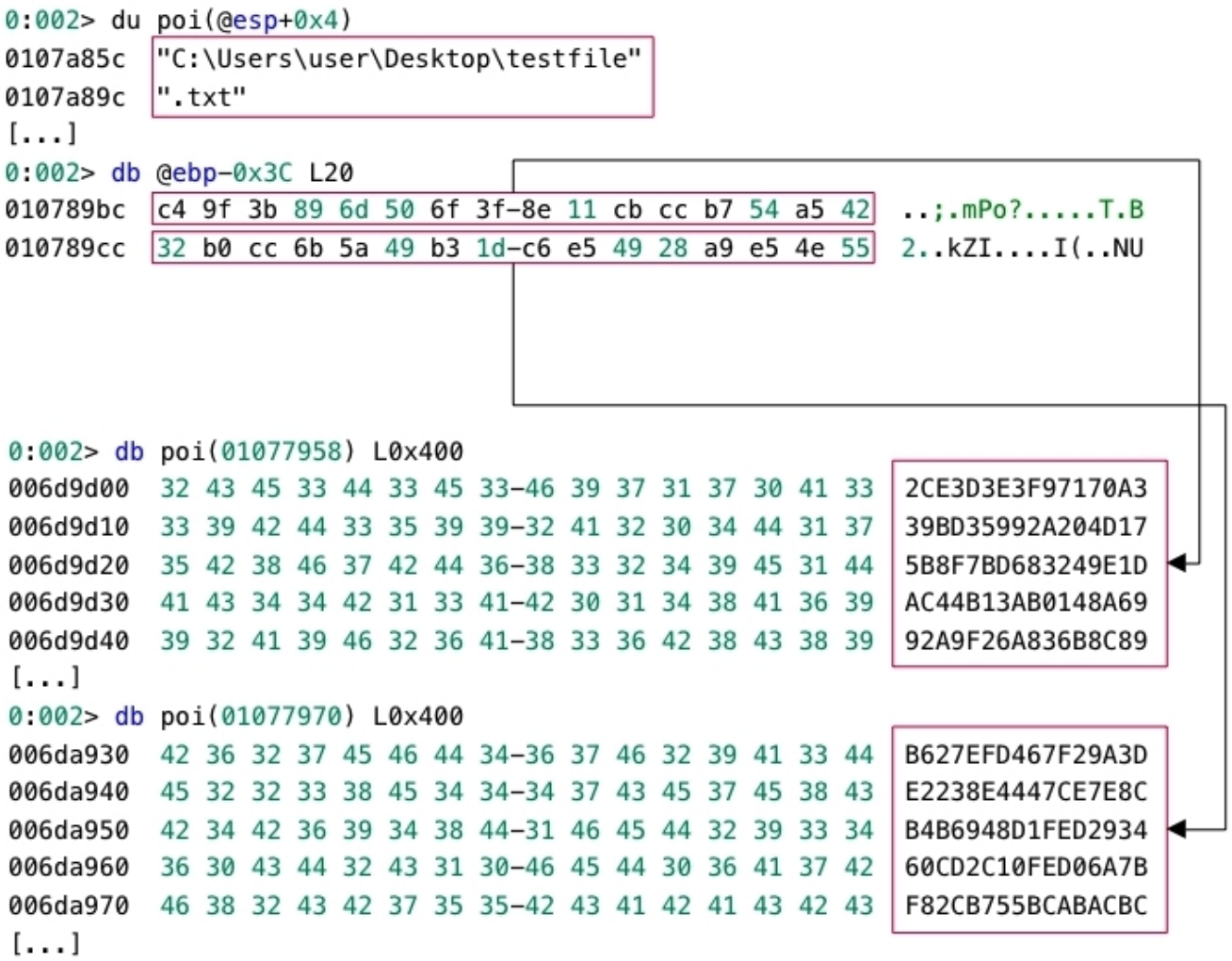
For each file being encrypted, PYSA first generates two random arrays of 16 bytes. The first byte array is an AES-CBC symmetric encryption key and the second is an initialization vector (IV). PYSA then encrypts the AES-CBC key and the IV using a 4096-bit RSA public key. This public key is Abstract Syntax Notation One (ASN.1)-encoded and is stored in Distinguished Encoding Rules (DER) format in the file that implements the PYSA ransomware:

```
Public-Key: (4096 bit)
Modulus:
    00:b8:65:d5:07:19:53:4c:5b:dc:4b:bb:2f:92:f6:
    cd:ee:f5:f7:ac:f4:75:fb:17:80:30:78:66:75:33:
    9e:52:a0:18:3a:ed:99:f5:bc:97:16:b8:0b:df:e3:
    3d:65:6b:bb:a2:6e:ea:6e:62:b7:4b:68:a0:a0:a3:
    90:f2:3a:8e:2c:0a:9e:d1:9f:b9:e7:7b:9e:19:3a:
    [...]
    70:c9:c9:b0:34:32:c3:39:90:48:8f:21:67:43:b0:
    98:d1:b9:ee:e8:3f:30:9a:36:32:b7:3e:48:8c:22:
    bf:01:33:1f:98:75:86:e8:3c:cb:3a:0d:57:75:90:
    df:ed:03:67:ec:8c:b0:55:18:81:10:a3:9c:f7:f9:
    ab:b1:8f:8b:26:62:69:de:18:2e:78:d3:1e:14:81:
    97:82:14:a7:e8:ce:aa:94:81:9c:0e:7c:29:b7:45:
    68:b7:74:b6:32:5c:c7:f8:3c:58:43:95:02:3f:a3:
    c2:78:38:1e:30:d6:3e:51:15:ab:fc:6b:87:fe:34:
    79:70:b7
Exponent: 17 (0x11)
```

The public key that the PYSA ransomware uses to encrypt AES-CBC keys and IVs

The PYSA ransomware then uses the *HexEncoder* class of CryptoPP library to encode in strings the data segments that are the encrypted AES-CBC key and IV. This encoding represents the digits of the hexadecimal representation of the bytes of these data segments as uppercase American Standard Code for Information Interchange (ASCII) characters.

The RSA-encrypted form of the AES-CBC key and IV is 512 bytes big due to the 4096-bit RSA key used for encryption. Therefore, the encoding operation results in two strings of 1024 bytes:



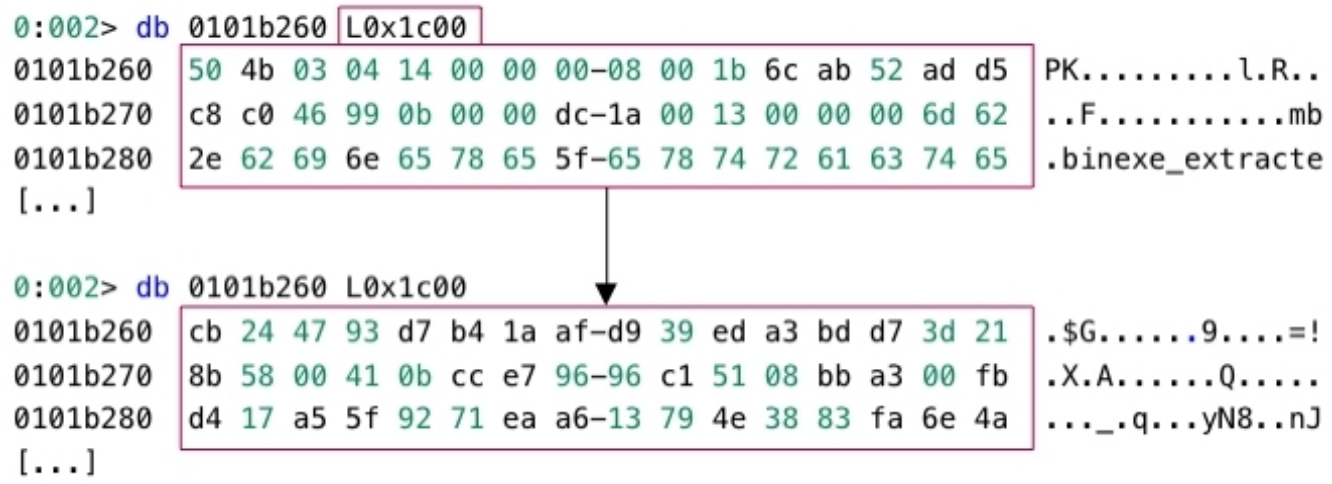
The unencrypted and RSA-encrypted form of an AES-CBC key and IV

The PYSA ransomware then encrypts 100 equal-sized data blocks of the file being encrypted, starting from the beginning of the file. For encrypting the data blocks, the ransomware uses the AES-CBC encryption algorithm with the previously generated AES-CBC key and IV. The ransomware calculates the size of a single data block for encryption (in bytes) by calculating:

$$[?][?][?]\frac{\frac{filesize}{100}}{1024}[?][?][?]\times 1024$$

where $\lfloor \rfloor$ is the *floor* function and *filesize* is the size of the file in bytes.

Since AES-CBC operates in a block cipher mode, the encrypted form of the data blocks is equal in size to the data blocks themselves. After encrypting a data block, the PYSA ransomware writes the encrypted form of the data block in the file, replacing the original data block. This encryption procedure normally results in some data at the end of the file being left unencrypted:



Unencrypted and encrypted form of a file data block (data block size: 7168 bytes)

The ransomware then appends to the end of the file the strings that store the encrypted forms of the AES-CBC key and IV. Since each of these strings is 1024 bytes big, the size of the file that PYSA has encrypted is greater by 2 KB than the size of the original, unencrypted file. The ransomware then proceeds to encrypt the next file designated for encryption:

```
00331650  00 00 00 00 01 00 18 00  c1 9a 61 de f3 ff d6 01  |..........a.....|
00331660  c9 73 e9 b8 f3 ff d6 01  c9 73 e9 b8 f3 ff d6 01  |.s.......s......|
00331670  50 4b 05 06 00 00 00 00  01 00 01 00 66 00 00 00  |PK..........f...|
00331680  0a 16 33 00 00 00 39 34  36 38 37 41 45 36 33 31  |..3...94687AE631|
00331690  46 42 46 39 41 32 32 37  33 45 34 36 36 31 43 44  |FBF9A2273E4661CD|
[...]
00331e60  30 34 45 35 36 30 35 32  39 41 31 37 41 45 30 33  |04E560529A17AE03|
00331e70  44 35 32 46 46 37 30 32  45 41 43 45 35 45 30 30  |D52FF702EACE5E00|
00331e80  30 44 44 43 37 42                               |0DDC7B|
00331e86
```

The encrypted form of an AES-CBC key and IV, appended to the end of a file

After it encrypts all files designated for encryption, the PYSA ransomware stores the value *PYSA* in the registry key *HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System\legalnoticecaption* and the ransom note in the registry key *HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System\legalnoticetext*. This displays the ransom note to users at system start-up, which effectively brings the users' attention to it.

The PYSA ransomware then releases the mutex *Pysa* and writes Windows batch script code into a file named *update.bat*. PYSA first places this file in the temporary directory of the user in whose context the ransomware executes (for example *C:\Users\user\AppData\Local\Temp*) and then executes it. *update.bat* deletes the file that implements the PYSA ransomware and the directory in which this file is stored. *update.bat* also deletes itself:

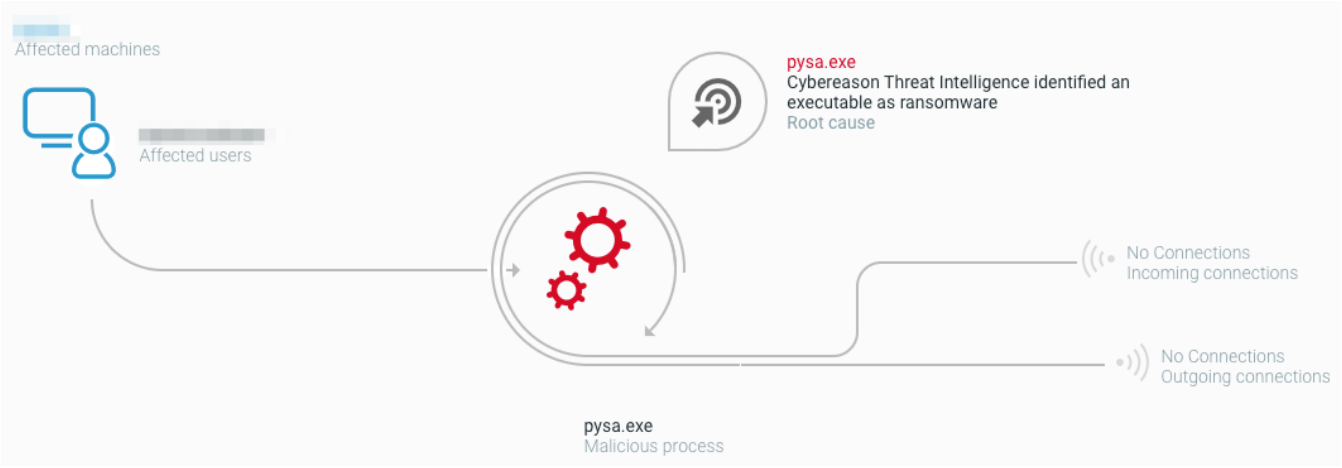
```
:Repeat
del "C:\Users\user\Desktop\pysa\pysa.exe"
if exist "C:\Users\user\Desktop\pysa\pysa.exe" goto Repeat
rmdir "C:\Users\user\Desktop\pysa"
del "C:\Users\user\AppData\Local\Temp\update.bat"
```

The content of update.bat

Detection and Prevention

Cybereason Prevents PYSA Ransomware

The Cybereason Defense Platform is able to detect and prevent the execution of the PYSA ransomware using multi-layer protection that detects and blocks ransomware with threat intelligence, machine learning and next-gen antivirus (NGAV) capabilities:



The Cybereason Defense Platform detects the PYSA ransomware based on threat intelligence

The *Anti-Malware* feature of the Cybereason Defense Platform detects and prevents the execution of the PYSA ransomware. Behavioral detection techniques in the platform are able to detect and prevent any attempt to encrypt files and automatically generates a MalOp™ for it:

The Anti-Malware feature of the Cybereason Defense Platform detects the PYSA ransomware

Cybereason GSOC MDR

In this section, the Cybereason GSOC provides additional, proactive ways for detecting the presence of the PYSA ransomware in systems, and defending against this threat.

YARA-Based Detection

The following YARA rule is useful for detecting the presence of the PYSA ransomware in the context of running processes or in the filesystem:

```
rule Pysa_ransomware
{
meta:
    description = "YARA rule for identifying the Pysa ransomware."
    author = "Aleksandar Milenkoski"
    date = "2021-07"

strings:
    $code = { 68 00 04 00 00 ?? ?? E8 7C BD 02 00 ?? ?? E8 A5 C2 02 00 ?? ?? ?? ?? ?? ?? ?? ??
    DD ?? ?? ?? ?? ?? ?? ?? DD ?? ?? E8 5D 81 03 00 59 ?? E8 B6 BE 02 00 }
    $s1 = "CryptoPP" ascii wide
    $s2 = "pysa" ascii wide nocase fullword
    $s3 = "Protect Your System Amigo" ascii wide nocase

condition:
    uint16(0) == 0x5A4D and uint32(uint32(0x3C)) == 0x00004550 and $s2 and 2 of ($code,$s1,$s3)
}
```

YARA rule for identifying the PYSA ransomware

Mutex Object Locking

The PYSA ransomware creates a mutex object named *Pysa*. If this mutex object already exists and is therefore locked, the ransomware terminates without encrypting any data. This is to the advantage of defenders such that a mutex object named *Pysa* can be locked by a legitimate process on a given system with the intention to stop any potential future execution of the PYSA ransomware on the system.

The PowerShell script below demonstrates this defense technique. The script creates, opens, and therefore locks a mutex object named *Pysa*, and releases the object when the user issues the *Ctrl+C* command. Users can execute the script by issuing the command *powershell.exe ./pysa_mutex_lock.ps1* in the directory where the script file is stored, where *pysa_mutex_lock.ps1* is the filename of the script file:

```
function create_pysa_mutex
{
    $created = $False
    $mutex = New-Object -TypeName System.Threading.Mutex($true, "Pysa", [ref]$created)
    Write-Host "Mutex object named Pysa created, opened, and locked: $created."
    return $mutex
}
function release_pysa_mutex
{
    param (
        $mutex
    )
    $mutex.ReleaseMutex()
    $mutex.Dispose()
}
$mutex = create_pysa_mutex
try
{
    while($true)
    {
        Start-Sleep -Seconds 1
    }
}
finally{
    release_pysa_mutex($mutex)
    Write-Host "Mutex object released."
}
```

PowerShell script that locks a mutex object named Pysa

General Recommendations

- Enable the *Anti-Ransomware* feature on the Cybereason NGAV and set the *Anti-Ransomware* protection mode to *Prevent.*
- Enable the *Anti-Malware* feature on the Cybereason NGAV and enable the *Detect* and *Prevent* modes of this feature.
- Make sure your systems are timely patched in order to minimize the risk of ransomware infections by vulnerability exploitation.
- Use secure passwords, regularly rotate passwords, and use multi-factor authentication where possible.
- Disable unused RDP services, properly secure used RDP services, and regularly monitor RDP log data for bruteforce attempts and other irregular activities.
- Regularly backup files to a secured remote location and implement a data recovery plan. Regular data backups ensure that you can restore your data after a ransomware attack.
- Securely handle email messages that originate from external sources. This includes disabling hyperlinks and investigating the content of email messages to identify phishing attempts.

Cybereason is dedicated to teaming with defenders to end cyber attacks from endpoints to the enterprise to everywhere - including modern ransomware. Learn more about ransomware defense here or schedule a demo today to learn how your organization can benefit from an operation-centric approach to security.

Indicators of Compromise

Executables	SHA-256 hash: *7FD3000A3AFBF077589C300F90B59864EC1FB716FEBA8E288ED87291C8FDF7C3* File size: 512512 bytes
Associated files	*Readme.README* *%TEMP%\update.bat*
Mutex objects	*Pysa*
Email domains	*protonmail.com* *onionmail.org*
Registry keys	*HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System\legalnoticecaption* *HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System\legalnoticetext*

MITRE ATT&CK Techniques

Execution	Defense Evasion	Discovery	Impact
Native API	Indicator Removal on Host: File Deletion	File and Directory Discovery	Data Encrypted for Impact
	Modify Registry		

About the Researcher:

Aleksandar Milenkoski, Senior Threat and Malware Analyst, Cybereason Global SOC

Aleksandar Milenkoski is a Senior Threat and Malware Analyst with the Cybereason Global SOC team. He is involved primarily in reverse engineering and threat research activities. Aleksandar has a PhD in system security. Prior to Cybereason, his work focussed on research in intrusion detection and reverse engineering security mechanisms of the Windows 10 operating system.

About the Author

Cybereason Global SOC Team

The Cybereason Global SOC Team delivers 24/7 Managed Detection and Response services to customers on every continent. Led by cybersecurity experts with experience working for government, the military and multiple industry verticals, the Cybereason Global SOC Team continuously hunts for the most sophisticated and pervasive threats to support our mission to end cyberattacks on the endpoint, across the enterprise, and everywhere the battle moves.

All Posts by Cybereason Global SOC Team