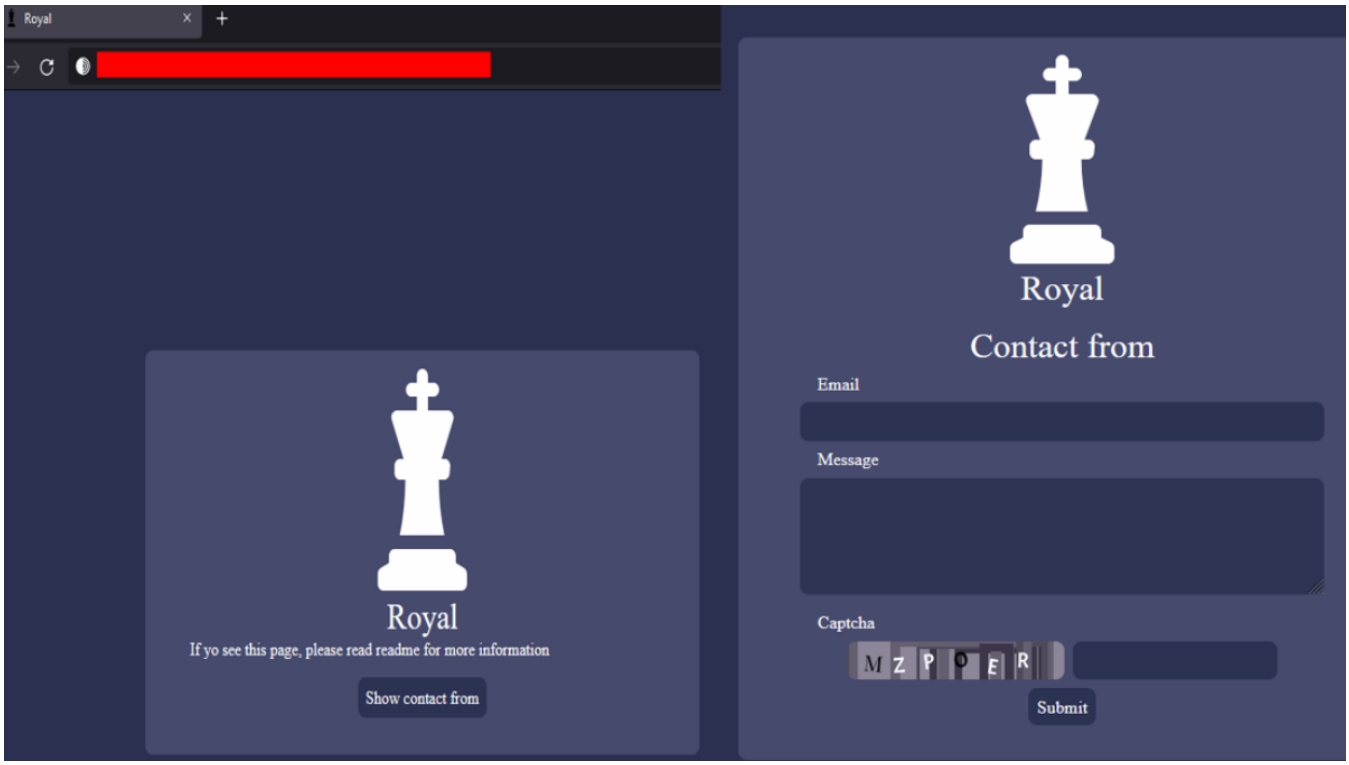
Reconstructing the last activities of Royal Ransomware

yoroi.company/research/reconstructing-the-last-activities-of-royal-ransomware/

November 17, 2022



11/17/2022

Introduction

Royal Ransomware is a new group first spotted on Bleeping Computer last September, where the cybersecurity news site revealed a connection with another malware known as Zeon.

At the moment, we don't have much information about the group and all its actual TTPs, but we know that they use the Double Extortion model to threaten the victims, as stated inside the ransom note.

```
Hello!

    If you are reading this, it means that your system were hit by Royal ransomware.
    Please contact us via :
    http://royal2xthig3ou5hd7zsliqazy6yygk2cdelaxtni2fyad6dgmpxedid.onion/dddddddddddddddddddddddddddddddd

In the meantime, let us explain this case.It may seem complicated, but it is not!
Most likely what happened was that you decided to save some money on your security infrastructure.
Alas, as a result your critical data was not only encrypted but also copied from your systems on a secure server.
From there it can be published online.Then anyone on the internet from darknet criminals, ACLU journalists, Chinese government(different names for the same thing),
and even your employees will be able to see your internal documentation: personal data, HR reviews, internal lawsuitsand complains, financial reports, accounting, intellectual property, and more!

    Fortunately we got you covered!

Royal offers you a unique deal.For a modest royalty(got it? got it ? ) for our pentesting services we will not only provide you with an amazing risk mitigation service,
covering you from reputational, legal, financial, regulatory, and insurance risks, but will also provide you with a security review for your systems.
To put it simply, your files will be decrypted, your data restoredand kept confidential, and your systems will remain secure.

    Try Royal today and enter the new era of data security!
    We are looking to hearing from you soon!
```

Figure 1: Ransom Note

About Royal Ransomware Group

The Cyber intelligence community has proof that the group started its malicious activities since January, with other ransomware payloads. So, we can say they started their malicious career as affiliated with other Ransomware-as-a-Service providers. But it only the last two months, it started to apply the Double Extorsion model, with an ad-hoc website in the Dark Web.

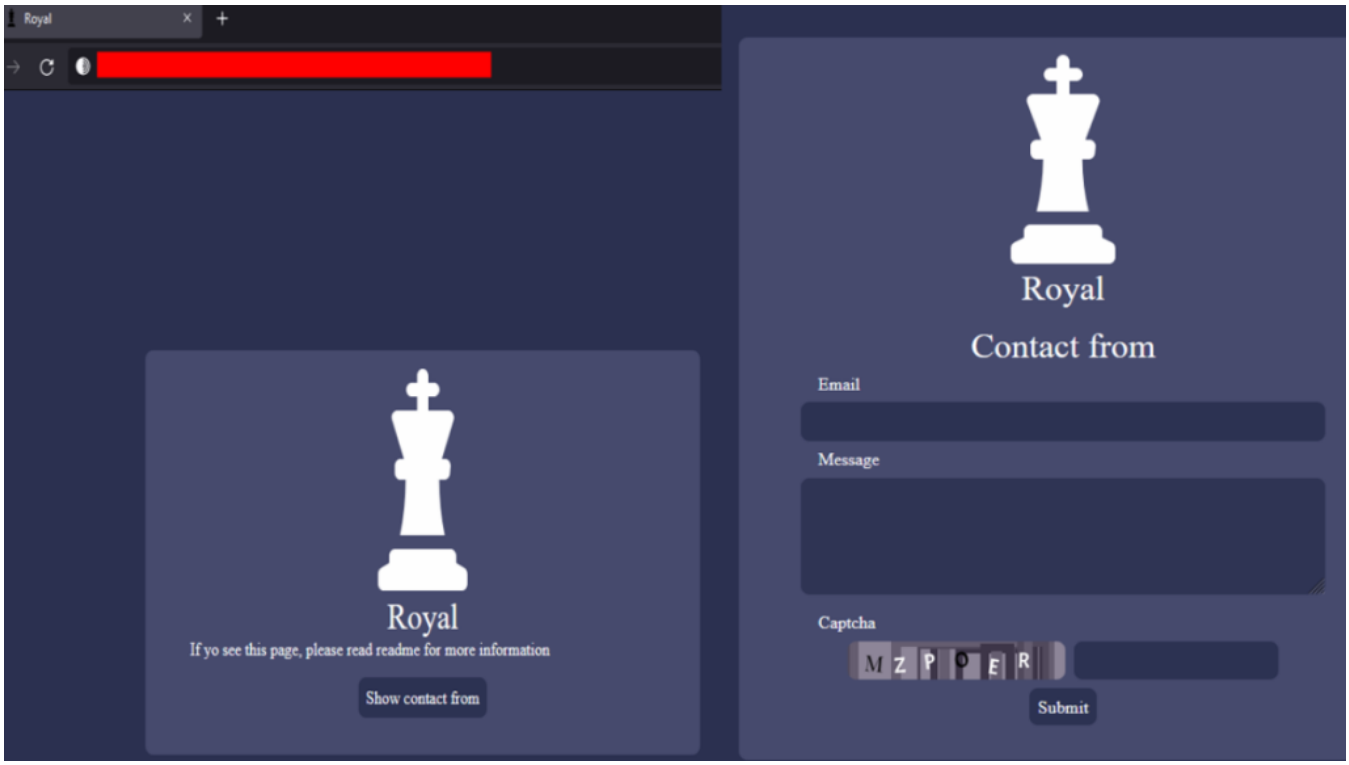


Figure 2: Royal Ransomware main page

During its existence, it seems that the ransomware group didn't adopt the Ransomware-as-a-Service model to recruit other affiliates to infect victims. The reason might be that the core team wants initially to create a malicious "brand positioning" inside the threat landscape.

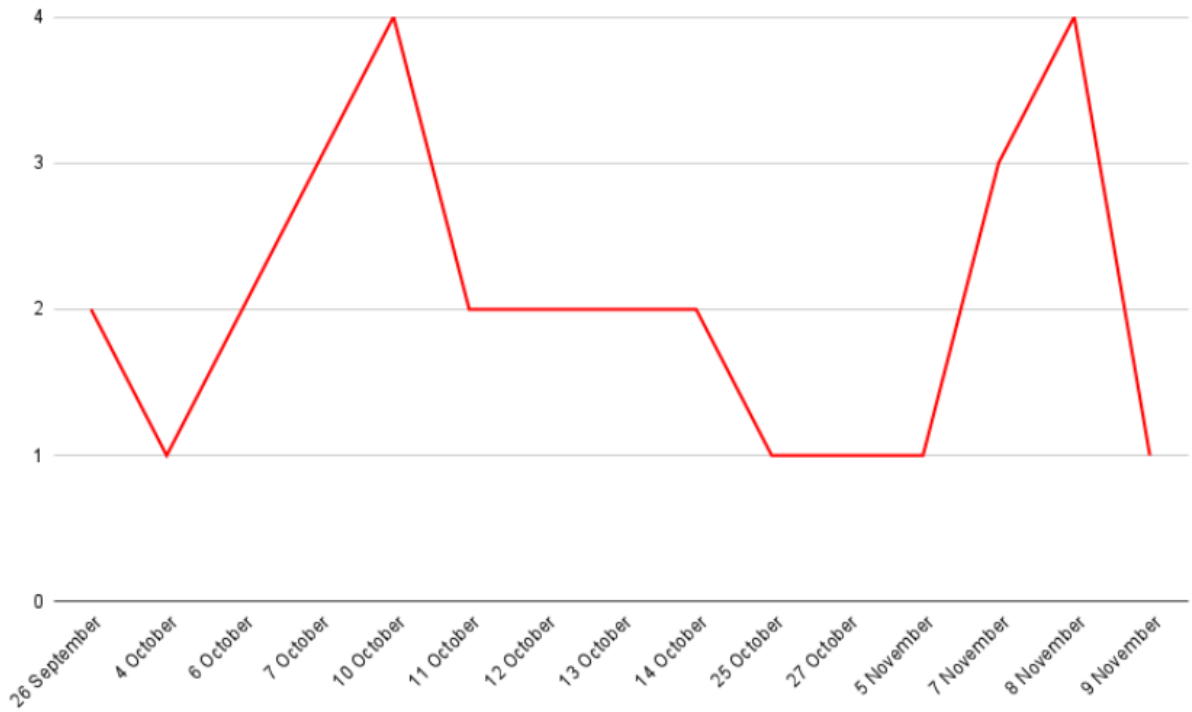


Figure 3: Distribution of the Attacks

At the same time, we don't know which toolkit is used to implement the exfiltration capability. We don't know if the group uses some custom malware, or if it leverages public storage platforms, such as Mega, Dropbox, etc.

Technical Analysis

We managed to obtain a recent specimen of this threat and analyzed its features and malicious capabilities, in order to create signatures and provide technical insight to better detection.

Tha analyzed sample has the following static information:

Hash	9db958bc5b4a21340ceeeb8c36873aa6bd02a460e688de56ccbba945384b1926
Threat	Royal Ransomware
Brief Description	Ransomware payload
SSDEEP	49152:cDVwASOLGtlqrRIU6i9+vazNqQlJZP1BMU2thA8mNtNCiJlrRUFcJ7HlPcLzk+5c:wm+GaNqqJJ12vlZol8cJ7rcl

property	value
md5	AFD5D656A42A746E95926EF07933F054
sha1	04028A0A1D44F81709040C31AF026785209D4343
sha256	9DB958BC5B4A21340CEEEB8C36873AA6BD02A460E688DE56CCBBA945384B1926
first-bytes-hex	4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00 B8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00 00
first-bytes-text	M Z .. .. @ .. ..
file-size	3085312 bytes
entropy	6.603
imphash	n/a
signature	Microsoft Visual C++
tooling	wait...
entry-point	48 83 EC 28 E8 07 06 00 00 48 83 C4 28 E9 7A FE FF FF CC CC CC CC CC CC CC CC CC CC CC CC CC...
file-version	n/a
description	n/a
file-type	executable
cpu	64-bit
subsystem	GUI
compiler-stamp	Mon Aug 15 14:53:34 2022 | UTC
debugger-stamp	Mon Aug 15 14:53:34 2022 | UTC
resources-stamp	0x00000000
import-stamp	0x00000000
exports-stamp	n/a

Figure 4: Static Information about the sample

Royal Ransomware is written in C/C++ and it is launched by command line. That behavior suggests that there is a previous and totally human-operated intrusion performed by a pen-testing team, which gained access to the internal network and performed privilege escalation and lateral movement operation.

When the executable is launched, it needs three parameters, otherwise the infection doesn't start:

```
lea     rdx, aPath      ; "-path"
call    cs:lstrcmpW
test    eax, eax
jnz     short loc_14007DDC3
mov     r15, [rbx+8]
inc     esi
add     rbx, 8
jmp     loc_14007DE4C
-----------------------------------------------------------

                        ; CODE XREF: WinMain+C2↑j
mov     rcx, [rbx]      ; lpString1
lea     rdx, aId        ; "-id"
call    cs:lstrcmpW
test    eax, eax
jnz     short loc_14007DE1D
mov     rdi, [rbx+8]
add     rbx, 8
mov     rcx, rdi        ; lpString
inc     esi
call    cs:lstrlenW
mov     [rsp+6EF0h+lpUsedDefaultChar], r12 ; lpUsedDefaultChar
mov     r8, rdi         ; lpWideCharStr
mov     r9d, eax        ; cchWideChar
mov     [rsp+6EF0h+lpDefaultChar], r12 ; lpDefaultChar
lea     rax, [rbp+6DF0h+MultiByteStr]
mov     [rsp+6EF0h+cbMultiByte], 21h ; '!' ; cbMultiByte
xor     edx, edx        ; dwFlags
mov     [rsp+6EF0h+lpMultiByteStr], rax ; lpMultiByteStr
mov     ecx, 0FDE9h     ; CodePage
call    cs:WideCharToMultiByte
jmp     short loc_14007DE4C
-----------------------------------------------------------

                        ; CODE XREF: WinMain+E5↑j
mov     rcx, [rbx]      ; lpString1
lea     rdx, aEp        ; "-ep"
call    cs:lstrcmpW
```

Figure 5: Parameters needed by Royal Ransomware

We constructed a table with the three parameters found inside the sample and we provide a small description of that:

Parameter	Description
-path	Specifies an exact path where to encrypt files
-id	Victim's ID, needed to run the sample, must be 32 characters
-ep	Encryption percentage (feature not implemented in this sample)

Table: Parameters description

After that, the sample starts the preparation of the ransomware operation by deleting the shadow copies:

```
lea     rdx, aDeleteShadowsA ; " delete shadows /all /quiet"
lea     rcx, [rbp+6DF0h+CommandLine] ; LPWSTR
call    cs:wsprintfW
xorps   xmm0, xmm0
mov     [rsp+6EF0h+StartupInfo.cb], 68h ; 'h'
xor     eax, eax
lea     rdx, [rbp+6DF0h+CommandLine] ; lpCommandLine
mov     dword ptr [rbp+6DF0h+StartupInfo.hStdError+4], eax
lea     rcx, ApplicationName ; "C:\\Windows\\System32\\vssadmin.exe"
mov     qword ptr [rsp+6EF0h+ProcessInformation.dwProcessId], rax
xor     r9d, r9d        ; lpThreadAttributes
lea     rax, [rsp+6EF0h+ProcessInformation]
xor     r8d, r8d        ; lpProcessAttributes
mov     [rsp+6EF0h+lpProcessInformation], rax ; lpProcessInformation
lea     rax, [rsp+6EF0h+StartupInfo]
mov     [rsp+6EF0h+lpStartupInfo], rax ; lpStartupInfo
mov     [rsp+6EF0h+lpUsedDefaultChar], r12 ; lpCurrentDirectory
mov     [rsp+6EF0h+lpDefaultChar], r12 ; lpEnvironment
mov     [rsp+6EF0h+cbMultiByte], r12d ; dwCreationFlags
mov     dword ptr [rsp+6EF0h+lpMultiByteStr], r12d ; bInheritHandles
movups  xmmword ptr [rsp+6EF0h+StartupInfo+4], xmm0
movups  xmmword ptr [rbp+6DF0h+StartupInfo.lpDesktop+4], xmm0
movups  xmmword ptr [rbp+6DF0h+StartupInfo.dwY], xmm0
movups  xmmword ptr [rbp+6DF0h+StartupInfo.dwYCountChars], xmm0
movups  xmmword ptr [rbp+6DF0h+StartupInfo+44h], xmm0
movups  xmmword ptr [rbp+6DF0h+StartupInfo.hStdInput+4], xmm0
movups  xmmword ptr [rsp+6EF0h+ProcessInformation.hProcess], xmm0
call    cs:CreateProcessW
```

Figure 6: Deleting Shadow Copies

Then, the malware starts the preparation for the encryption processes, by creating the lists of the elements to be excluded during the fetching of files and folders. For the files' extensions, the exclusions are:

```
.exe,.dll,.bat,.lnk,.royal,
```

```
9db958bc5b4a2134...  StrStrIW ( "C:\Tools\FakeNet-NG\fakenet1.4.11\CHANGELOG.txt", ".exe" )
9db958bc5b4a2134...  StrStrIW ( "C:\Tools\FakeNet-NG\fakenet1.4.11\CHANGELOG.txt", ".dll" )
9db958bc5b4a2134...  StrStrIW ( "C:\Tools\FakeNet-NG\fakenet1.4.11\CHANGELOG.txt", ".bat" )
9db958bc5b4a2134...  StrStrIW ( "C:\Tools\FakeNet-NG\fakenet1.4.11\CHANGELOG.txt", ".lnk" )
9db958bc5b4a2134...  StrStrIW ( "C:\Tools\FakeNet-NG\fakenet1.4.11\CHANGELOG.txt", ".royal" )
9db958bc5b4a2134...  StrStrIW ( "C:\Tools\FakeNet-NG\fakenet1.4.11\CHANGELOG.txt", "README.TXT" )
```

Figure 7: Checking file extension

Instead, for the folders exclusion there are the following:

```
windows, royal, $recycle.bin, google, perflogs, mozilla, tor browser, boot, $windows.~ws, $windows.~bt, windows.old
```

```
9db958bc5b4a2134...  StrStrIW ( "C:\Tools\FakeNet-NG\fakenet1.4.11\listeners", "windows" )
9db958bc5b4a2134...  StrStrIW ( "C:\Tools\FakeNet-NG\fakenet1.4.11\listeners", "royal" )
9db958bc5b4a2134...  StrStrIW ( "C:\Tools\FakeNet-NG\fakenet1.4.11\listeners", "$recycle.bin" )
9db958bc5b4a2134...  StrStrIW ( "C:\Tools\FakeNet-NG\fakenet1.4.11\listeners", "google" )
9db958bc5b4a2134...  StrStrIW ( "C:\Tools\FakeNet-NG\fakenet1.4.11\listeners", "perflogs" )
9db958bc5b4a2134...  StrStrIW ( "C:\Tools\FakeNet-NG\fakenet1.4.11\listeners", "mozilla" )
9db958bc5b4a2134...  StrStrIW ( "C:\Tools\FakeNet-NG\fakenet1.4.11\listeners", "tor browser" )
9db958bc5b4a2134...  StrStrIW ( "C:\Tools\FakeNet-NG\fakenet1.4.11\listeners", "boot" )
9db958bc5b4a2134...  StrStrIW ( "C:\Tools\FakeNet-NG\fakenet1.4.11\listeners", "$windows.~ws" )
9db958bc5b4a2134...  StrStrIW ( "C:\Tools\FakeNet-NG\fakenet1.4.11\listeners", "$windows.~bt" )
9db958bc5b4a2134...  StrStrIW ( "C:\Tools\FakeNet-NG\fakenet1.4.11\listeners", "windows.old" )
```

Figure 8: Checking folder name

Royal Ransomware has also the capability to infect and encrypt the shared resources inside the internal network. It uses the NetShareEnum seeking the "ADMIN$" and "IPC$" records and then, it proceeds to encrypt the files contained inside the shared folders

```
call    cs:NetShareEnum
mov     r15d, eax
test    eax, eax
jz      short loc_14007E61B
cmp     eax, 0EAh ; 'ê'
jnz     loc_14007E710

                        ; CODE XREF: sub_14007E510+FE↑j
mov     rdi, [rsp+300h+bufptr]
mov     esi, 1
cmp     [rsp+300h+entriesread], esi
jb      loc_14007E6FA
nop

                        ; CODE XREF: sub_14007E510+1DF↓j
mov     rdx, [rdi]      ; lpString2
lea     rcx, String1    ; "ADMIN$"
call    cs:lstrcmpiW
test    eax, eax
jz      loc_14007E6E5
mov     rdx, [rdi]      ; lpString2
lea     rcx, aIpc       ; "IPC$"
call    cs:lstrcmpiW
test    eax, eax
jz      loc_14007E6E5
mov     r9, [rdi]
lea     r8, [rbp+200h+szAddressString]
lea     rdx, aSS        ; "\\\\%s\\%s"
lea     rcx, [rbp+200h+var_240] ; LPWSTR
call    cs:wsprintfW
```

Figure 9: Network shares encryption

At this point, we have the encryption phase. Royal Group uses a mixture of RSA and AES algorithms. The RSA public Key is hardcoded in the sample, and it is easy to retrieve that:

```
                dq offset aHelloIfYouAreR ; "Hello!\r\n\r\n\tIf you are reading this"...
; const CHAR String[]
String          db '-----BEGIN RSA PUBLIC KEY-----',0Ah
                                        ; DATA XREF: sub_14007F870+49↑o
                                        ; sub_14007F870+59↑o ...
                db 'MIICCAKCAgEA0y6/qfb0GqxB2tNEW8qLCtT7U3XCzp1OVjVkaTH9SBV1k3NBElgC',0Ah
                db 'esSVOFAUAG5nT3WO+CdN26ScoKsFjzKGYh8c7vyoi7L5dDBRdoTEW5+u2rBSIN3c',0Ah
                db 'pkR0Wsq+gT3j0gtvjVybMfp6NRifsMfrcAV9tlrzUw7Da2mx+1Ik9Aa5RaaOxv8N',0Ah
                db 'ahH6OSJ8Qz1G3uCgZaXAULlAqNnlN0KtSo4VsXt/sOnDh1pGFf8jqU8sqwJUkcWk',0Ah
                db 'RdeYdsDyiDrUFxXkHJsiZb8lFk6b01Rm2yS9+kyZxi1yhB1m0kStUUmbN2aoZMy1',0Ah
                db 'pIKxDa2clhhYw+JEMrbCKWW1Aif2hR55nBgL2kwiaNShXUm3yEsfbnd/lJ5ORMUF',0Ah
                db 'tVmaEFEyvVutc86TcNhu0NCHfYihtgbcke7cvy23XnL/qlFL4OzdAnyupz0n69mk',0Ah
                db '1TSJBR7so3GhvQz53wTps9FXSwWlRpGLTCGRo4OnLnke7Hi5YL+Wb/4c6xWz8biX',0Ah
                db '+jNeg5Zko+CL3I7ywJkyCWuH9Pr7nccWr1s35BSV8Aj9rMwmOsak2BG91Db0yovg',0Ah
                db 'FLmKMhkwxpBgFfePXIZF687DxpwYJ5fN44OyUCfNrtfejfSFtjhDCwFy/YpBhZ/w',0Ah
                db '2Bnw8hTLNALEIsDBhAlQBVYAGYhUgDbpvs/GN3qijyFWdESqlCK1Eg0CAQM=',0Ah
                db '-----END RSA PUBLIC KEY-----',0Ah
```

Figure 10: Public RSA Key

In the following scheme, we highlight the encryption routine and the result of the encryption of a test file:

Figure 11: Encryption routine

The encryption routine is assisted by the OpenSSL library. An AES key is randomly generated and then it is protected with the RSA public key. In this way, the ransomware operator the encryption conserves the RSA private key to decrypt the original AES key and then the files could be restored.

Conclusion

The market of cyber extorsion is still growing and other threat actors are riding the wave of the most infamous 2022 trend. The case Royal Ransomware Group is representative because it started by adopting the affiliation on the RaaS market, and when they reached an appropriate expertise and experience, it created an independent group with a “proprietary” ransomware payload.

As stated, at the moment we have no proof they adopt the Ransomware-as-a-Service model, but the group is formed only by talented pen-testers and malware developers aimed at making money through the Double Extorsion model. However, this doesn't mean that in the near future the group will reach such a maturity that they will be capable of implementing the RaaS model.

Indicators of Compromise

- 9db958bc5b4a21340ceeeb8c36873aa6bd02a460e688de56ccbba945384b1926
- c24c59c8f4e7a581a5d45ee181151ec0a3f0b59af987eacf9b363577087c9746
- 5fda381a9884f7be2d57b8a290f389578a9d2f63e2ecb98bd773248a7eb99fa2
- 312f34ee8c7b2199a3e78b4a52bd87700cc8f3aa01aa641e5d899501cb720775
- 491c2b32095174b9de2fd799732a6f84878c2e23b9bb560cd3155cbdc65e2b80
- 2598e8adb87976abe48f0eba4bbb9a7cb69439e0c133b21aee3845dfccf3fb8f
- f484f919ba6e36ff33e4fb391b8859a94d89c172a465964f99d6113b55ced429
- 7cbfea0bff4b373a175327d6cc395f6c176dab1cedf9075e7130508bec4d5393

Yara Rules

```
rule royal_ransomware {

meta:
   author = "Yoroi Malware ZLab"
   description = "Rule for Royal Ransomware"
   last_updated = "2022-11-09"
   tlp = "WHITE"
   category = "informational"

strings:
   // x32
   $1 = {8d 84 ?? ?? ?? ?? ?? 50 ff 15 ?? ?? ?? ?? 83 f8 20 74 ?? 6a 00 ff 15 ?? ?? ?? ??}
   $2 = {68 ?? ?? ?? ?? ff 30 89 44 ?4 20 ff 15 ?? ?? ?? ?? 85 c0 75 ?? 8b 44 ?4 10 46 8b 0c b0 89 4c ?4 1c e9 ?? ?? ??
?? 8b 44 ?4 18 68 ?? ?? ?? ?? ff 30 ff 15 ?? ?? ?? ??}
   // x64
   $3 = {4? 8d ?? ?? ?? ?? ?? ff 15 ?? ?? ?? ?? 83 f8 20 74 ?? 33 c9 ff 15 ?? ?? ?? ??}
   $4 = { 4? 8d 15 ?? ?? ?? ?? ff 15 ?? ?? ?? ?? 85 c0 75 ?? 4? 8b 7b 08 ff c6 4? 83 c3 08 e9 ?? ?? ?? ?? 4? 8b 0b 4? 8d
15 ?? ?? ?? ?? ff 15 ?? ?? ?? ??}

condition:
    (($1 and $2) or ($3 and $4)) and uint16(0) == 0x5A4D
}
```

This blog post was authored by Luigi Martire, Carmelo Ragusa of Yoroi Malware ZLAB