VEDERE LABS

Royal Ransomware

Analysis of one of the most active ransomware groups in late 2022 and early 2023

Date: January 10, 2023

Contents

1. Executive Summary

The Royal ransomware threat actor group, initially tracked as DEV-0569, first emerged in early 2022 and has been especially active since the end of the same year. Royal ransomware was first observed by security researchers in September 2022 and since then multiple attacks were detected, targeting organizations across the globe, but mostly in U.S., Brazil and Europe. It was among the most active ransomware groups in December 2022 and has already announced its first victim of 2023: DSBJ, a Chinese company that manufactures components for IoT and telecommunications equipment.

Security researchers have noticed that the group was probably created by one of the former Conti teams ("Conti Team One") and used the Zeon encryptor in some attacks. The group employs the double extortion tactic by gaining access to a victim's environment, encrypting their data as well as exfiltrating sensitive data and demanding a ransom to decrypt files. The files are encrypted using the Advanced Encryption Standard (AES) and given the extension **.royal**. In recent attacks, the encrypted files also had the extension **.royal_***.

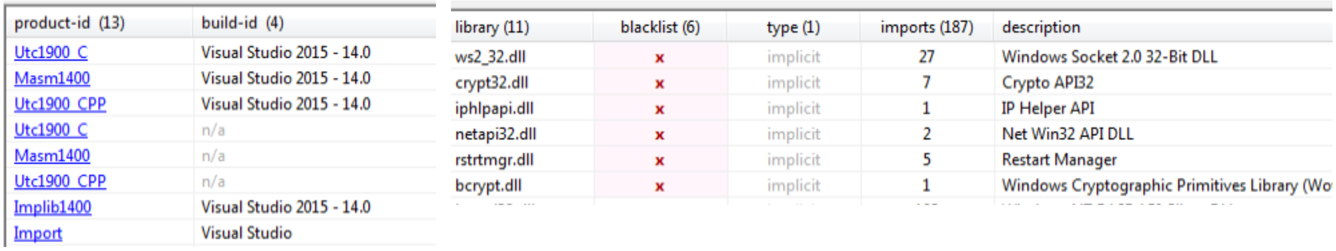
The initial attack vectors are specifically designed and tailored for individual targets, including some unusual techniques. Their techniques for initial infection include malicious advertisements, phishing links that point to a malware payload, fake software installers and fake forum pages to lure potential victims. The group's phishing techniques include callback phishing, where they impersonate various service providers and software providers in emails that look like subscription renewals. The phishing emails contain phone numbers that the victim should contact to cancel their subscription. Upon calling the number, the threat actors convince the victim to install remote access software. This remote access software would serve as initial access to the target network.

In a recent campaign, the ransomware actors used a compiled remote desktop malware, which was used to drop the tools that were later used to infiltrate the victim's system. There have been instances where the threat actor used QakBot and Cobalt Strike for lateral movement, while NetScan was used to look for any network connected systems. Once they infiltrated the system, the ransomware actors used tools like Nsudo, PowerTool and Process Hacker to disable any security-related services running in the system. The ransomware actors used PsExec to execute the malware and to spread the malware to other machines in the network. The group also relies heavily on defense evasion techniques such as using encrypted binaries and disabling antivirus solutions.

In this report, we analyze the Royal ransomware payload in Section 2; present threat hunt opportunities in Section 3; and share details of the Royal ransomware group's tactics, techniques and procedures (TTPs) in Section 4.

2. Technical Analysis

The Royal ransomware payload is a 64-bit executable written in C++ that is not packed and that imports several interesting DLLs, as shown in Figure 1.

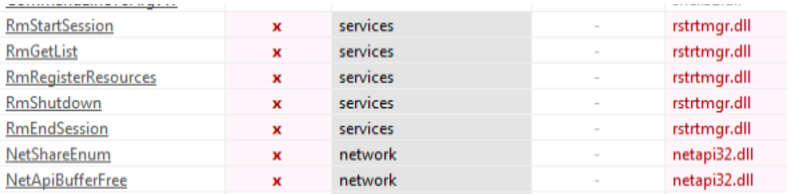


product-id (13)	build-id (4)
Utc1900_C	Visual Studio 2015 - 14.0
Masm1400	Visual Studio 2015 - 14.0
Utc1900_CPP	Visual Studio 2015 - 14.0
Utc1900_C	n/a
Masm1400	n/a
Utc1900_CPP	n/a
Implib1400	Visual Studio 2015 - 14.0
Import	Visual Studio

library (11)	blacklist (6)	type (1)	imports (187)	description
ws2_32.dll	x	implicit	27	Windows Socket 2.0 32-Bit DLL
crypt32.dll	x	implicit	7	Crypto API32
iphlpapi.dll	x	implicit	1	IP Helper API
netapi32.dll	x	implicit	2	Net Win32 API DLL
rstrtmgr.dll	x	implicit	5	Restart Manager
bcrypt.dll	x	implicit	1	Windows Cryptographic Primitives Library (Wo

Figure 1 – Royal ransomware executable and DLLs

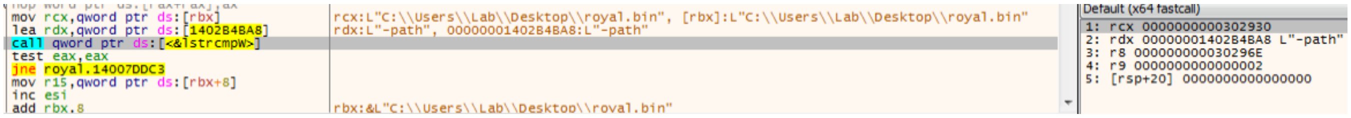
The ransomware uses the Windows Restart Manager DLL to check if any of the files targeted to be encrypted are being used by other processes. The malware uses API calls such as `RmStartSession`, `RmGetList` and `RmShutDown` (shown in Figure 2) to start the session, get the list of processes using the resource and kill those processes using the resource.



RmStartSession	×	services	-	rstrtmgr.dll
RmGetList	×	services	-	rstrtmgr.dll
RmRegisterResources	×	services	-	rstrtmgr.dll
RmShutdown	×	services	-	rstrtmgr.dll
RmEndSession	×	services	-	rstrtmgr.dll
NetShareEnum	×	network	-	netapi32.dll
NetApiBufferFree	×	network	-	netapi32.dll

Figure 2 – Windows Restart Manager APIs

The ransomware supports three arguments for execution: `-path`, `-ep` and `-id`. The last argument is mandatory while the other two are optional. The `-path` parameter (shown in Figure 3) is used to specify the path to be encrypted, `-ep` is used to specify the percentage of the file that needs be encrypted and `-id` is a unique number used by the ransomware group to identify its victims



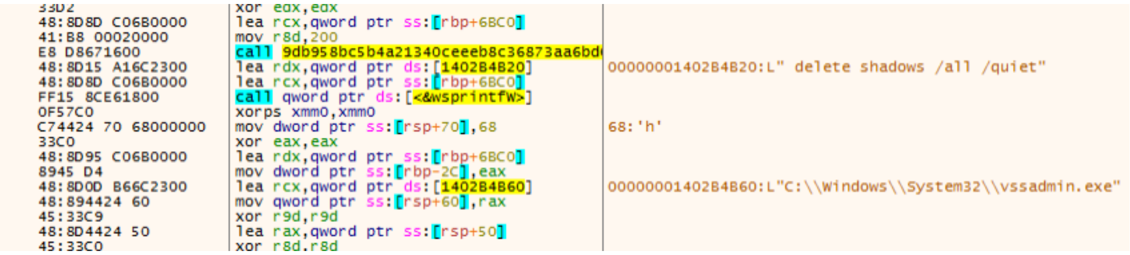
```
mov rcx,qword ptr ds:[rbx]             rcx:L"C:\\Users\\Lab\\Desktop\\royal.bin", [rbx]:L"C:\\Users\\Lab\\Desktop\\royal.bin"
lea rdx,qword ptr ds:[1402B4BA8]       rdx:L"-path", 00000001402B4BA8:L"-path"
call qword ptr ds:[<&lstrcmpW>]
test eax,eax
jne royal.14007DDC3
mov r15,qword ptr ds:[rbx+8]
inc esi
add rbx,8                              rbx:&L"C:\\Users\\Lab\\Desktop\\royal.bin"
```

```
Default (x64 fastcall)
1: rcx 0000000000302930
2: rdx 00000001402B4BA8 L"-path"
3: r8 000000000030296E
4: r9 0000000000000002
5: [rsp+20] 0000000000000000
```

Figure 3 – `-path` parameter

The command executed to run the payload is as follows: `cmd.exe /c "c:\windows\temp\royal.exe -id <32-bit victim ID>"`

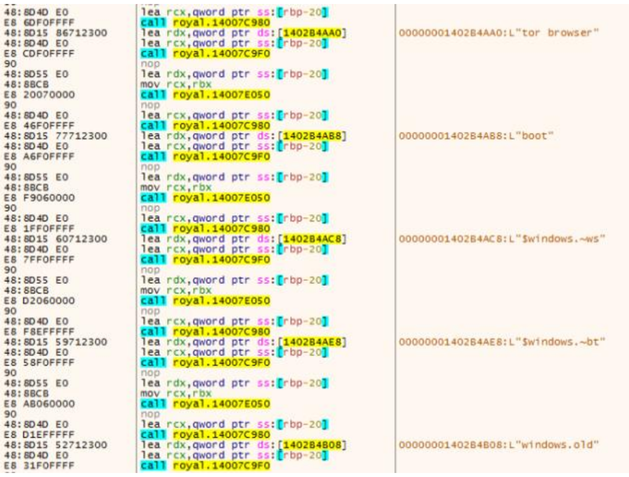
The ransomware will not run if no value is specified for the `-id` parameter. The ransomware will then attempt to delete volume shadow copies using the following command: `vssadmin.exe delete shadows /all /quiet`



```
33D2                   xor edx,edx
48:8D8D C06B0000       lea rcx,qword ptr ss:[rbp+6BC0]
41:B8 00020000         mov r8d,200
E8 D8671600            call 9db958bc5b4a21340ceeeb8c36873aa6bd
48:8D15 A16C2300       lea rdx,qword ptr ds:[1402B4B20]       00000001402B4B20:L" delete shadows /all /quiet"
48:8D8D C06B0000       lea rcx,qword ptr ss:[rbp+6BC0]
FF15 8CE61800          call qword ptr ds:[<&wsprintfW>]
0F57C0                 xorps xmm0,xmm0
C74424 70 68000000     mov dword ptr ss:[rsp+70],68           68:'h'
33C0                   xor eax,eax
48:8D95 C06B0000       lea rdx,qword ptr ss:[rbp+6BC0]
8945 D4                mov dword ptr ss:[rbp-2C],eax
48:8D0D B66C2300       lea rcx,qword ptr ds:[1402B4B60]       00000001402B4B60:L"C:\\Windows\\System32\\vssadmin.exe"
48:894424 60           mov qword ptr ss:[rsp+60],rax
45:33C9                xor r9d,r9d
48:8D4424 50           lea rax,qword ptr ss:[rsp+50]
45:33C0                xor r8d,r8d
```

Figure 4 – Shadow copy deletion

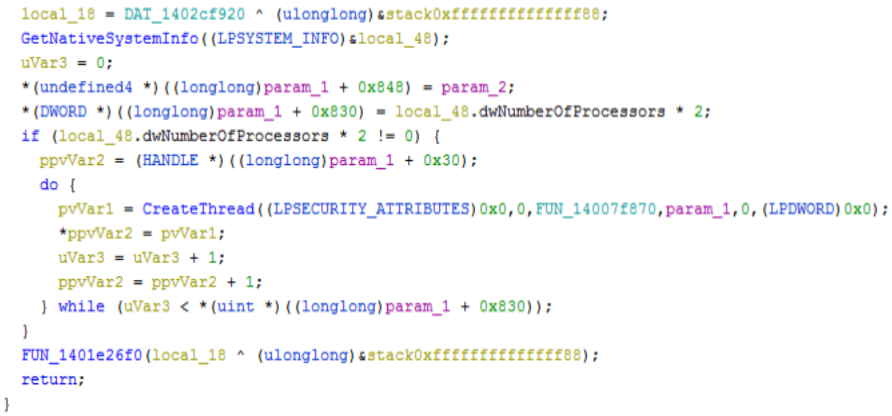
Once the shadow copies are deleted, the malware then decrypts a list of file extensions. Files with the following extensions would be excluded from encryption: `.exe`, `.dll`, `.bat`, `.lnk`, `.royal`. Similarly, a list of folders is also decrypted, which are to be excluded from encryption: `windows`, `$recycle.bin`, `google`, `royal perflogs`, `mozilla`, `tor browser`, `boot`, `$windows.~ws`, `$windows.~bt`, `windows.old`



```
48:8D4D E0          lea rcx,qword ptr ss:[rbp-20]
E8 6DF0FFFF         call royal.14007C980
48:8D15 86712300    lea rdx,qword ptr ds:[1402B4AA0]     00000001402B4AA0:L"tor browser"
48:8D4D E0          lea rcx,qword ptr ss:[rbp-20]
E8 CDF0FFFF         call royal.14007C9F0
90                  nop
48:8D55 E0          lea rdx,qword ptr ss:[rbp-20]
48:8BCB             mov rcx,rbx
E8 20070000         call royal.14007E050
90                  nop
48:8D4D E0          lea rcx,qword ptr ss:[rbp-20]
E8 46F0FFFF         call royal.14007C980
48:8D15 77712300    lea rdx,qword ptr ds:[1402B4AB8]     00000001402B4AB8:L"boot"
48:8D4D E0          lea rcx,qword ptr ss:[rbp-20]
E8 A6F0FFFF         call royal.14007C9F0
90                  nop
48:8D55 E0          lea rdx,qword ptr ss:[rbp-20]
48:8BCB             mov rcx,rbx
E8 F9060000         call royal.14007E050
90                  nop
48:8D4D E0          lea rcx,qword ptr ss:[rbp-20]
E8 1FF0FFFF         call royal.14007C980
48:8D15 60712300    lea rdx,qword ptr ds:[1402B4AC8]     00000001402B4AC8:L"$windows.~ws"
48:8D4D E0          lea rcx,qword ptr ss:[rbp-20]
E8 7FF0FFFF         call royal.14007C9F0
90                  nop
48:8D55 E0          lea rdx,qword ptr ss:[rbp-20]
48:8BCB             mov rcx,rbx
E8 D2060000         call royal.14007E050
90                  nop
48:8D4D E0          lea rcx,qword ptr ss:[rbp-20]
E8 F8EFFFFF         call royal.14007C980
48:8D15 59712300    lea rdx,qword ptr ds:[1402B4AE8]     00000001402B4AE8:L"$windows.~bt"
48:8D4D E0          lea rcx,qword ptr ss:[rbp-20]
E8 58F0FFFF         call royal.14007C9F0
90                  nop
48:8D55 E0          lea rdx,qword ptr ss:[rbp-20]
48:8BCB             mov rcx,rbx
E8 AB060000         call royal.14007E050
90                  nop
48:8D4D E0          lea rcx,qword ptr ss:[rbp-20]
E8 D1EFFFFF         call royal.14007C980
48:8D15 52712300    lea rdx,qword ptr ds:[1402B4B08]     00000001402B4B08:L"windows.old"
48:8D4D E0          lea rcx,qword ptr ss:[rbp-20]
E8 31F0FFFF         call royal.14007C9F0
```

Figure 5 – Directories excluded from encryption

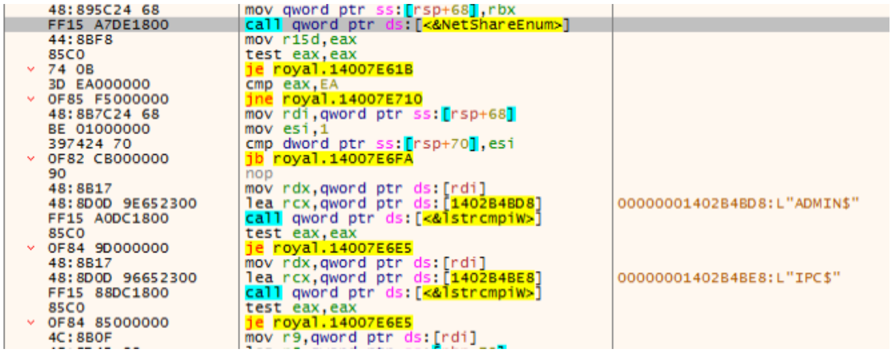
The Royal ransomware uses a multi-threaded encryption mechanism. The `GetNativeSystemInfo` API is used to get the number of processors available in a target machine. The threads for encryption are then created using this value.



```
  local_18 = DAT_1402cf920 ^ (ulonglong)&stack0xfffffffffffff88;
  GetNativeSystemInfo((LPSYSTEM_INFO)&local_48);
  uVar3 = 0;
  *(undefined4 *)((longlong)param_1 + 0x848) = param_2;
  *(DWORD *)((longlong)param_1 + 0x830) = local_48.dwNumberOfProcessors * 2;
  if (local_48.dwNumberOfProcessors * 2 != 0) {
    ppvVar2 = (HANDLE *)((longlong)param_1 + 0x30);
    do {
      pvVar1 = CreateThread((LPSECURITY_ATTRIBUTES)0x0,0,FUN_14007f870,param_1,0,(LPDWORD)0x0);
      *ppvVar2 = pvVar1;
      uVar3 = uVar3 + 1;
      ppvVar2 = ppvVar2 + 1;
    } while (uVar3 < *(uint *)((longlong)param_1 + 0x830));
  }
  FUN_1401e26f0(local_18 ^ (ulonglong)&stack0xfffffffffffff88);
  return;
}
```

Figure 6 – Thread creation for encryption

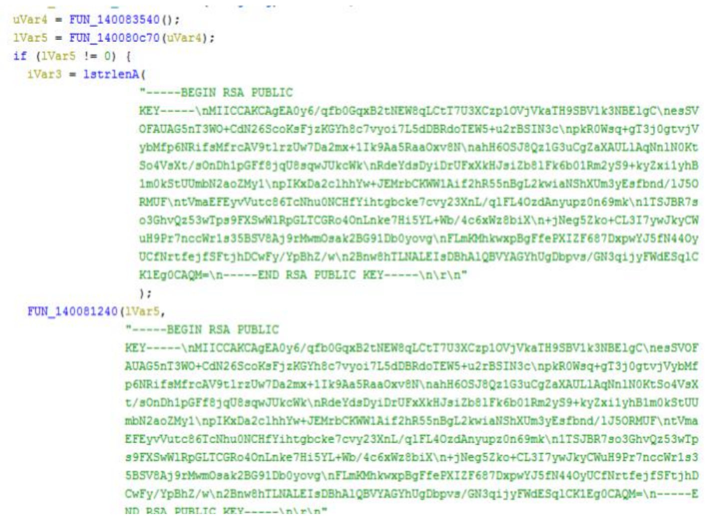
The ransomware then tries to enumerate the network shares available in the network using the `NetShareEnum` API. Shares `ADMIN$` and `IPC$` are excluded.



```
    48:895C24 68       mov qword ptr ss:[rsp+68],rbx
    FF15 A7DE1800      call qword ptr ds:[<&NetShareEnum>]
    44:8BF8            mov r15d,eax
    85C0               test eax,eax
  ˅ 74 0B              je royal.14007E61B
    3D EA000000        cmp eax,EA
  ˅ 0F85 F5000000      jne royal.14007E710
    48:8B7C24 68       mov rdi,qword ptr ss:[rsp+68]
    BE 01000000        mov esi,1
    397424 70          cmp dword ptr ss:[rsp+70],esi
  ˅ 0F82 CB000000      jb royal.14007E6FA
    90                 nop
    48:8B17            mov rdx,qword ptr ds:[rdi]
    48:8D0D 9E652300   lea rcx,qword ptr ds:[1402B4BD8]      00000001402B4BD8:L"ADMIN$"
    FF15 A0DC1800      call qword ptr ds:[<&lstrcmpiW>]
    85C0               test eax,eax
  ˅ 0F84 9D000000      je royal.14007E6E5
    48:8B17            mov rdx,qword ptr ds:[rdi]
    48:8D0D 96652300   lea rcx,qword ptr ds:[1402B4BE8]      00000001402B4BE8:L"IPC$"
    FF15 88DC1800      call qword ptr ds:[<&lstrcmpiW>]
    85C0               test eax,eax
  ˅ 0F84 85000000      je royal.14007E6E5
    4C:8B0F            mov r9,qword ptr ds:[rdi]
```

Figure 7 – Network share enumeration

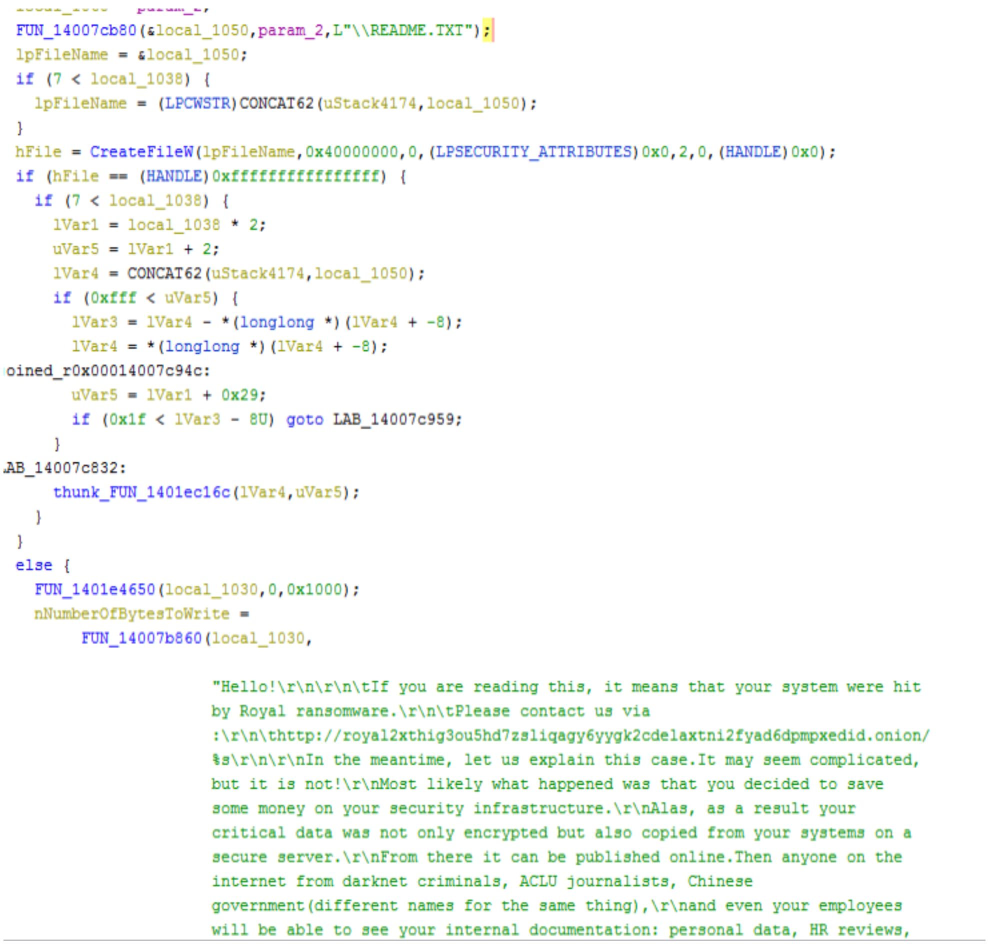
The ransomware then imports a hard-coded RSA public key that is embedded in the binary in plain text format (shown in Figure 8). This is used for encrypting the AES key used for file encryption.



```
  uVar4 = FUN_140083540();
  lVar5 = FUN_140080c70(uVar4);
  if (lVar5 != 0) {
    iVar3 = lstrlenA(
                     "-----BEGIN RSA PUBLIC
                     KEY-----\nMIICCAKCAgEA0y6/qfb0GqxB2tNEW8qLCtT7U3XCzp1OVjVkaTH9SBV1k3NBElgC\nesSV
                     OFAUAG5nT3WO+CdN26ScoKsFjzKGYh8c7vyoi7L5dDBRdoTEW5+u2rBSIN3c\npkR0Wsq+gT3j0gtvjV
                     ybMfp6NRifsMfrcAV9tlrzUw7Da2mx+1Ik9Aa5RaaOxv8N\nahH6OSJ8Qz1G3uCgZaXAULlAqNnlN0Kt
                     So4VsXt/sOnDh1pGFf8jqU8sqwJUkcWk\nRdeYdsDyiDrUFxXkHJsiZb8lFk6b01Rm2yS9+kyZxi1yhB
                     1m0kStUUmbN2aoZMy1\npIKxDa2clhhYw+JEMrbCKWW1Aif2hR55nBgL2kwiaNShXUm3yEsfbnd/1J5O
                     RMUF\ntVmaEFEyvVutc86TcNhu0NCHfYihtgbcke7cvy23XnL/qlFL4OzdAnyupz0n69mk\n1TSJBR7s
                     o3GhvQz53wTps9FXSwWlRpGLTCGRo4OnLnke7Hi5YL+Wb/4c6xWz8biX\n+jNeg5Zko+CL3I7ywJkyCW
                     uH9Pr7nccWr1s35BSV8Aj9rMwmOsak2BG91Db0yovg\nFLmKMhkwxpBgFfePXIZF687DxpwYJ5fN44Oy
                     UCfNrtfejfSFtjhDCwFy/YpBhZ/w\n2Bnw8hTLNALEIsDBhAlQBVYAGYhUgDbpvs/GN3qijyFWdESqlC
                     K1Eg0CAQM=\n-----END RSA PUBLIC KEY-----\n\r\n"
                     );
    FUN_140081240(lVar5,
                  "-----BEGIN RSA PUBLIC
                  KEY-----\nMIICCAKCAgEA0y6/qfb0GqxB2tNEW8qLCtT7U3XCzp1OVjVkaTH9SBV1k3NBElgC\nesSVOF
                  AUAG5nT3WO+CdN26ScoKsFjzKGYh8c7vyoi7L5dDBRdoTEW5+u2rBSIN3c\npkR0Wsq+gT3j0gtvjVybMf
                  p6NRifsMfrcAV9tlrzUw7Da2mx+1Ik9Aa5RaaOxv8N\nahH6OSJ8Qz1G3uCgZaXAULlAqNnlN0KtSo4VsX
                  t/sOnDh1pGFf8jqU8sqwJUkcWk\nRdeYdsDyiDrUFxXkHJsiZb8lFk6b01Rm2yS9+kyZxi1yhB1m0kStUU
                  mbN2aoZMy1\npIKxDa2clhhYw+JEMrbCKWW1Aif2hR55nBgL2kwiaNShXUm3yEsfbnd/1J5ORMUF\ntVma
                  EFEyvVutc86TcNhu0NCHfYihtgbcke7cvy23XnL/qlFL4OzdAnyupz0n69mk\n1TSJBR7so3GhvQz53wTp
                  s9FXSwWlRpGLTCGRo4OnLnke7Hi5YL+Wb/4c6xWz8biX\n+jNeg5Zko+CL3I7ywJkyCWuH9Pr7nccWr1s3
                  5BSV8Aj9rMwmOsak2BG91Db0yovg\nFLmKMhkwxpBgFfePXIZF687DxpwYJ5fN44OyUCfNrtfejfSFtjhD
                  CwFy/YpBhZ/w\n2Bnw8hTLNALEIsDBhAlQBVYAGYhUgDbpvs/GN3qijyFWdESqlCK1Eg0CAQM=\n-----E
                  ND RSA PUBLIC KEY-----\n\r\n"
```

Figure 8 – Embedded RSA public key

The target files are encrypted using the OpenSSL library and the AES256 algorithm. Finally, a ransom note named `README.txt` is created in every directory (shown in Figure 9).



```
  FUN_14007cb80(&local_1050,param_2,L"\\README.TXT");
  lpFileName = &local_1050;
  if (7 < local_1038) {
    lpFileName = (LPCWSTR)CONCAT62(uStack4174,local_1050);
  }
  hFile = CreateFileW(lpFileName,0x40000000,0,(LPSECURITY_ATTRIBUTES)0x0,2,0,(HANDLE)0x0);
  if (hFile == (HANDLE)0xffffffffffffffff) {
    if (7 < local_1038) {
      lVar1 = local_1038 * 2;
      uVar5 = lVar1 + 2;
      lVar4 = CONCAT62(uStack4174,local_1050);
      if (0xfff < uVar5) {
        lVar3 = lVar4 - *(longlong *)(lVar4 + -8);
        lVar4 = *(longlong *)(lVar4 + -8);
oined_r0x00014007c94c:
        uVar5 = lVar1 + 0x29;
        if (0x1f < lVar3 - 8U) goto LAB_14007c959;
      }
AB_14007c832:
      thunk_FUN_1401ec16c(lVar4,uVar5);
    }
  }
  else {
    FUN_1401e4650(local_1030,0,0x1000);
    nNumberOfBytesToWrite =
         FUN_14007b860(local_1030,

                       "Hello!\r\n\r\n\tIf you are reading this, it means that your system were hit
                       by Royal ransomware.\r\n\tPlease contact us via
                       :\r\n\thttp://royal2xthig3ou5hd7zsliqagy6yygk2cdelaxtni2fyad6dpmpxedid.onion/
                       %s\r\n\r\nIn the meantime, let us explain this case.It may seem complicated,
                       but it is not!\r\nMost likely what happened was that you decided to save
                       some money on your security infrastructure.\r\nAlas, as a result your
                       critical data was not only encrypted but also copied from your systems on a
                       secure server.\r\nFrom there it can be published online.Then anyone on the
                       internet from darknet criminals, ACLU journalists, Chinese
                       government(different names for the same thing),\r\nand even your employees
                       will be able to see your internal documentation: personal data, HR reviews,
```

Figure 9 – Ransom note creation

3. Threat Hunt Opportunities

- **PsExec Service Installation:** `event_id = 7045 OR 7036 && service_name contains “psexesvc”`
- **PsExec Remote Command Execution:** `process _process_name = psexesvc.exe && process _name = cmd.exe`
- **Shadow Copy Deletion:** `process _name = vssadmin.exe && Commadline contains “delete*shadows”`
- **Local Admin Account Created Using Net.exe:** `process_name = net.exe OR net1.exe && Commadline contains “* administr* /add*”`

4. TTPs

Tactic	Technique
Initial Access	T1566: Phishing
	T1078: Valid Accounts
Discovery	T1083: File and Directory Discovery
	T1016: System Network Configuration Discovery
	T1046: Network Service Discovery
	T1057: Process Discovery
	T1082: System Information Discovery
	T1135: Network Share Discovery
Execution	T1059: Command and Scripting Interpreter
	T1569: System Services
	T1204: User Execution
Defense Evasion	T1562: Impair Defenses
	T1036: Masquerading
Impact	T1486: Data Encrypted for Impact
	T1489: Service Stop
	T1490: Inhibit System Recovery

5. References

- https://www.bleepingcomputer.com/news/security/callback-phishing-attacks-evolve-their-social-engineering-tactics/
- https://www.trendmicro.com/en_us/research/22/l/conti-team-one-splinter-group-resurfaces-as-royal-ransomware-wit.html
- https://www.microsoft.com/en-us/security/blog/2022/11/17/dev-0569-finds-new-ways-to-deliver-royal-ransomware-various-payloads/