Examining Code Reuse Reveals Undiscovered Links Among North Korea's Malware Families

By Jay Rosenberg and Christiaan Beek on Aug 09, 2018

This research is a joint effort by Jay Rosenberg, senior security researcher at Intezer, and

Christiaan Beek, lead scientist and senior principal engineer at McAfee. Intezer has also posted this story.

Attacks from the online groups Lazarus, Silent Chollima, Group 123, Hidden Cobra, DarkSeoul, Blockbuster, Operation Troy, and 10 Days of Rain are believed to have come from North Korea. But how can we know with certainty? And what connection does a DDoS and disk-wiping attack from July 4, 2009, have with WannaCry, one of the largest cyberattacks in the history of the cyber sphere?

From the Mydoom variant Brambul to the more recent Fallchill, WannaCry, and the targeting of cryptocurrency exchanges, we see a distinct timeline of attacks beginning from the moment North Korea entered the world stage as a significant threat actor.

Bad actors have a tendency to unwittingly leave fingerprints on their attacks, allowing researchers to connect the dots between them. North Korean actors have left many of these clues in their wake and throughout the evolution of their malware arsenal.

This post reflects months of research; in it we will highlight our code analysis illustrating key similarities between samples attributed to the Democratic People's Republic of Korea, a shared networking infrastructure, and other revealing data hidden within the binaries. Together these puzzle pieces show the connections between the many attacks attributed to North Korea and categorize different tools used by specific teams of their cyber army.

Valuable context

This article is too short to dig deeply into the history, politics, and economic changes of recent years. Nonetheless, we must highlight some events to put past and present cyber events into perspective.

Email Address *

Country *

--Please Select--

Submit

McAfee on Twitter

Follow us on Twitter

mcafee_labs

Threat actors continue to evolve their tools, increasing complexity and functionality. Learn how these cybercrimes… https://t.co/ibrUsYyn7V

21 hours ago

Reply · Retweet · Favorite

mcafee_labs

ICYMI: Our team analyzed Operation GhostSecret, a Hidden

The DPRK, like any country, wants to be as self-sufficient and independent as possible. However, for products such as oil, food, and foreign currency for trading, the country lacks resources and has to find ways of acquiring them. What can a nation do when legal international economics are denied? To survive, it must gain foreign currency for trading. One of the oldest ways to do this is to join the worlds of gambling (casinos) and drugs. In 2005, the United States wanted to shut down North Korean enterprises involved in illegal operations. They investigated a couple of banks in Asia that seemed to have ties with North Korea and operated as money laundering sites. One bank in particular is controlled by a billionaire gambling mogul who started a casino in Pyongyang and has close ties to Pyongyang. That bank, based in Macau, came back into the picture during an attack on the SWIFT financial system of a bank in Vietnam in 2015. The Macau bank was listed twice in the malware's code as a recipient of stolen funds:

Cobra campaign that leveraged multiple #malware implants i… https://t.co/FLgQ6Vn4yc

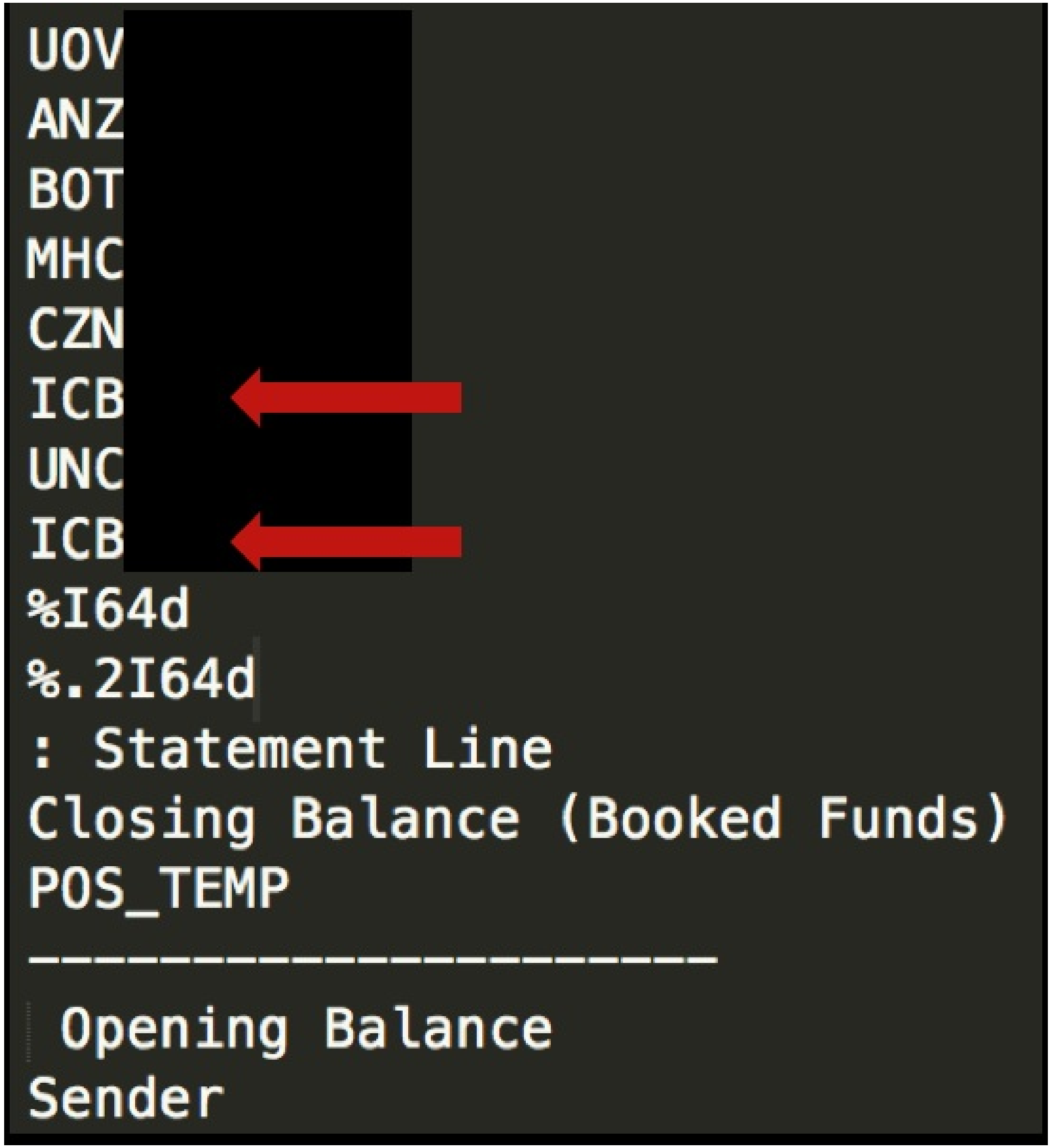
22 hours ago
Reply · Retweet · Favorite

mcafee_labs

Bad actors are exploiting connected home devices to find new innocent targets. Which connected device do you think… https://t.co/pxWq9B6Rqo

2 days ago
Reply · Retweet · Favorite

Next Article



```
UOV [redacted]
ANZ
BOT
MHC
CZN
ICB
UNC
ICB
%I64d
%.2I64d
: Statement Line
Closing Balance (Booked Funds)
POS_TEMP
---------------------
 Opening Balance
Sender
```

Figure 1: SWIFT code in malware.

Code reuse

There are many reasons to reuse malware code, which is very common in the world of cybercrime. If we take an average ransomware campaign, for example, once the campaign becomes less successful, actors often change some of basics such as using a different packer to bypass defenses. With targeted campaigns, an adversary must keep its tools undetected for as long as possible. By identifying reused code, we gain valuable insights about the "ancestral relations" to known threat actors or other campaigns. Our research was heavily focused on this type of analysis.

In our years of investigating cyber threats, we have seen the DPRK conduct multiple cyber campaigns. In North Korea, hackers' skills determine which cyber units they work for. We are aware two major focuses of DPRK campaigns: one to raise money, and one to pursue nationalist aims. The first workforce gathers money for the nation, even if that means committing cybercrime to hack into financial institutions, hijack gambling sessions, or sell pirated and cracked software. Unit 180 is responsible for illegally gaining foreign currency using hacking techniques. The second workforce operates larger campaigns motivated by nationalism, gathering intelligence from other nations, and in some cases disrupting rival states and military targets. Most of these actions are executed by Unit 121.

We focused in our research on the larger-scale nationalism-motivated campaigns, in which we discovered many overlaps in code reuse. We are highly confident that nation-state–sponsored groups were active in these efforts.

Timeline

We created a timeline of most of the malware samples and noticeable campaigns that we examined. We used primarily open-source blogs and papers to build this timeline and used the malware artifacts as a starting point of our research.

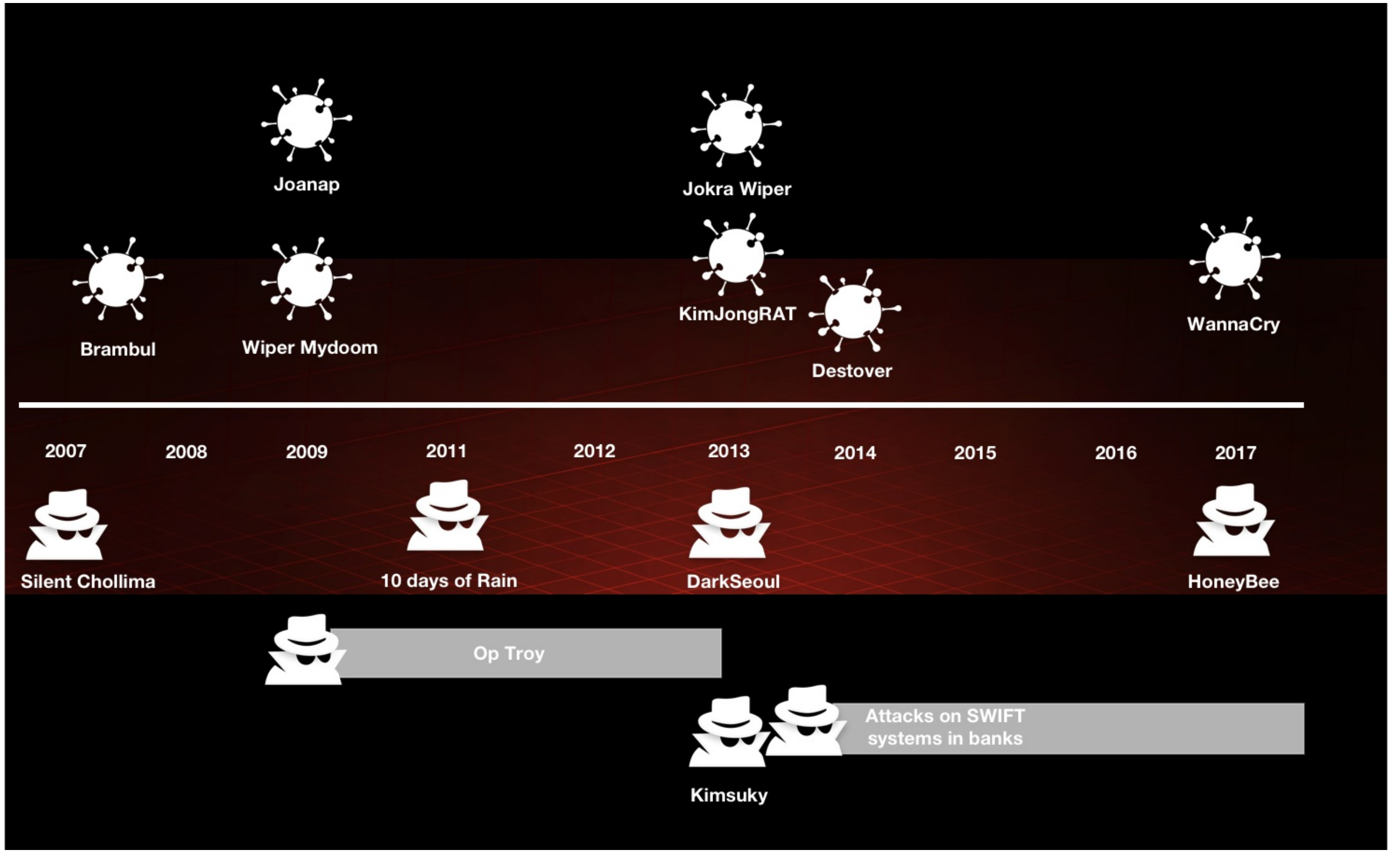


Figure 2: Timeline of malware and campaigns.

Analysis and observations

Similarities

During our research, we found many malware family names that are believed to be associated with North Korea's cyber operations. To better understand this threat actor and the similarities between the campaigns, we have used Intezer's code similarity detection engine to plot the links between a vast number of these malware families.

The following graph presents a high-level overview of these relations. Each node represents a malware family or a hacking tool (“Brambul,” “Fallchill,” etc.) and each line presents a code similarity between two families. A thicker line correlates to a stronger similarity. In defining similarities, we take into account only unique code connections, and disregard common code or libraries. This definition holds both for this graph and our entire research.

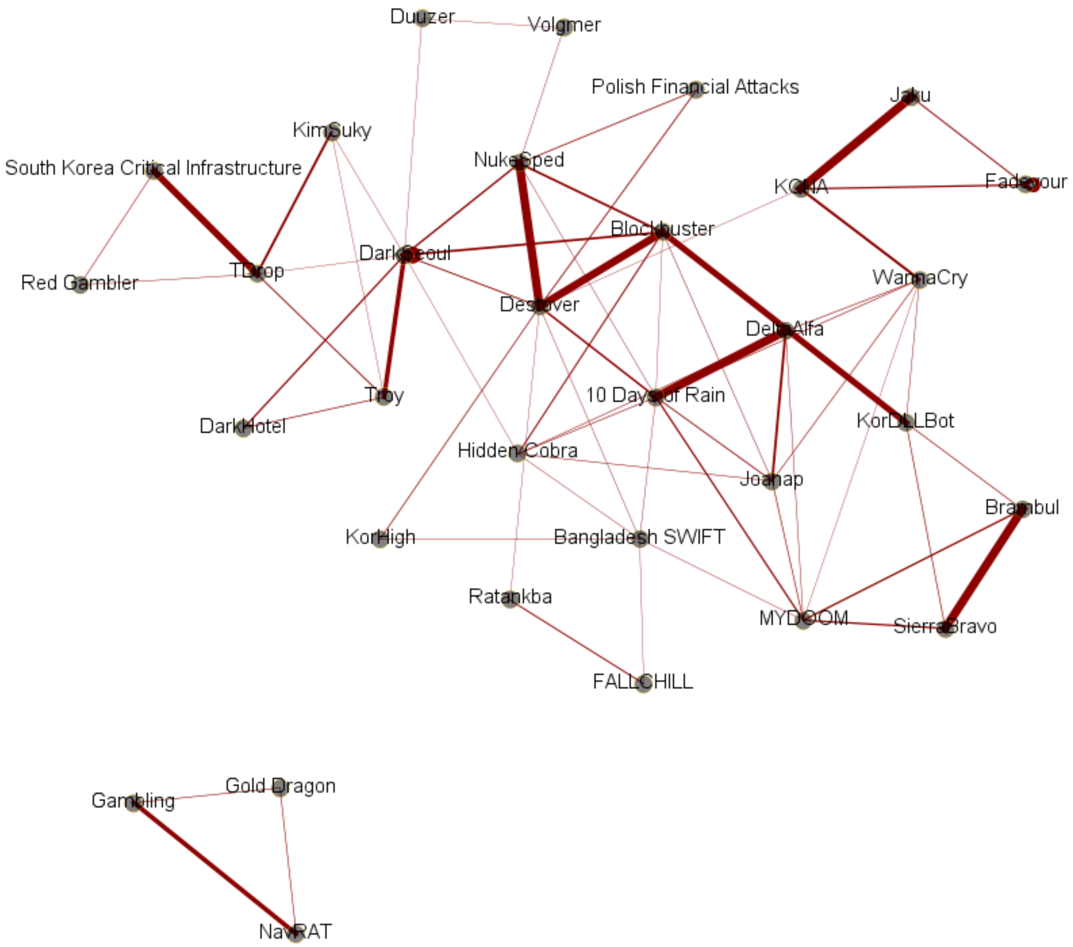


Figure 3: Code similarities between North Korean–associated malware families.

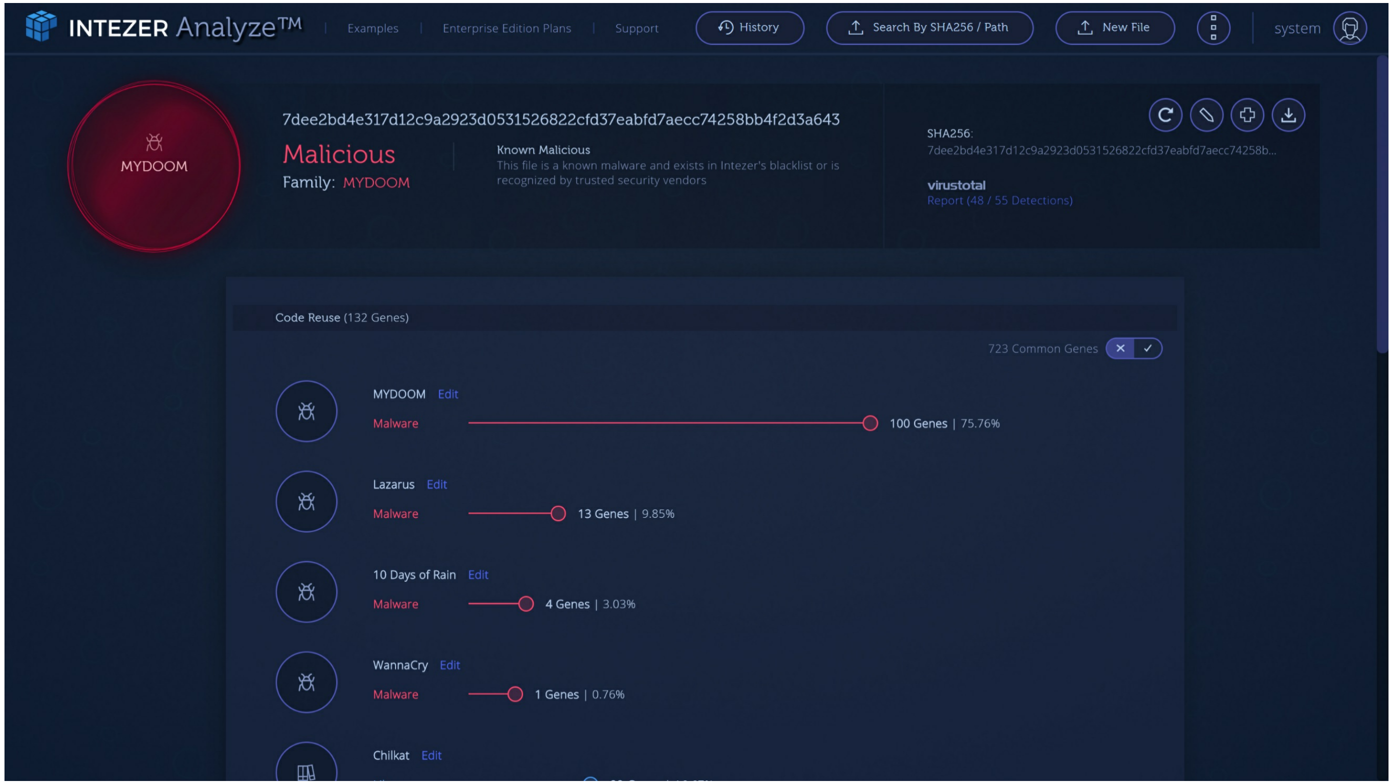
We can easily see a significant amount of code similarities between almost every one of the attacks associated with North Korea. Our research included thousands of samples, mostly unclassified or uncategorized. This graph was plotted using a data set of only several hundred samples, so there might be more connections than displayed here.

Deep technical analysis

During our research, we came across many code similarities between North Korean binaries that had not been seen before. Some of these attacks and malware have not been linked to one another, at least publicly. We will showcase four examples of reused code that has been seen only in malware attributed to North Korea.

1. Common SMB module

The first code example appeared in the server message block (SMB) module of WannaCry in 2017, Mydoom in 2009, Joanap, and DeltaAlfa. Further shared code across these families is an AES library from CodeProject. These attacks have been attributed to Lazarus; that means the group has reused code from at least 2009 to 2017.



7dee2bd4e317d12c9a2923d0531526822cfd37eabfd7aecc74258bb4f2d3a643

Malicious

Family: MYDOOM

Known Malicious
This file is a known malware and exists in Intezer's blacklist or is recognized by trusted security vendors

SHA256:
7dee2bd4e317d12c9a2923d0531526822cfd37eabfd7aecc74258b...

virustotal
Report (48 / 55 Detections)

Code Reuse (132 Genes)

723 Common Genes

MYDOOM Edit
Malware 100 Genes | 75.76%

Lazarus Edit
Malware 13 Genes | 9.85%

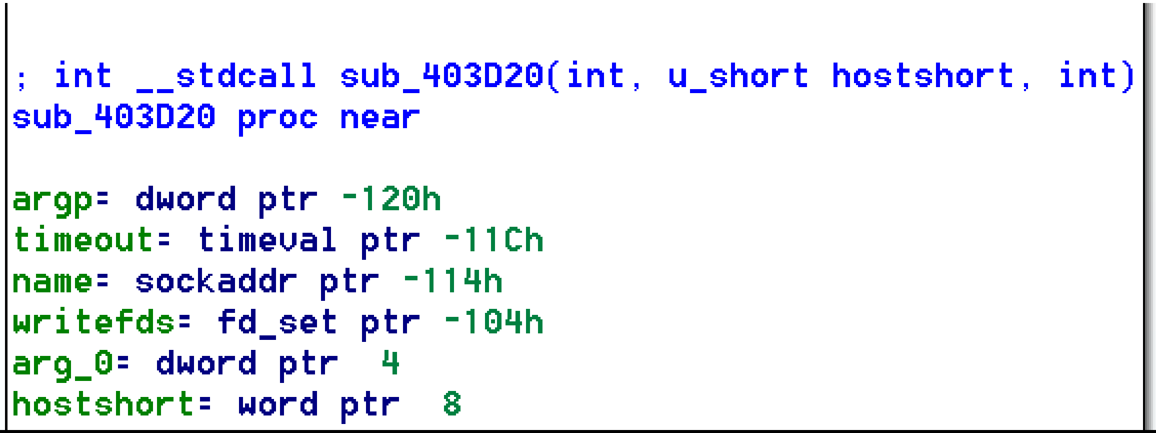
10 Days of Rain Edit
Malware 4 Genes | 3.03%

WannaCry Edit
Malware 1 Genes | 0.76%

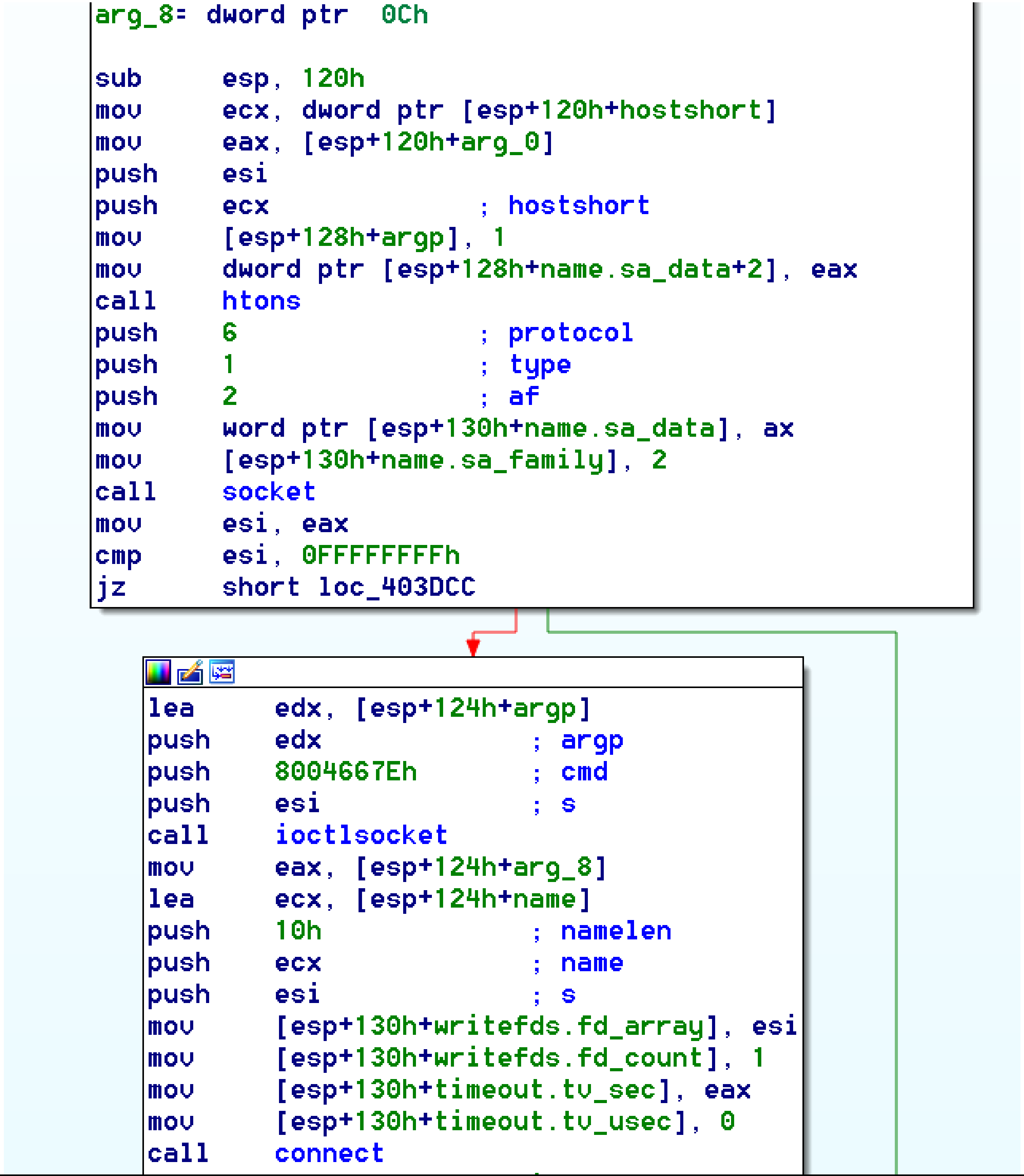
Chilkat Edit

Figure 4: Code overlap of a Mydoom sample.

In the next screenshots we highlight the exact code block that reflects the SMB module we found in campaigns other than WannaCry and Mydoom.

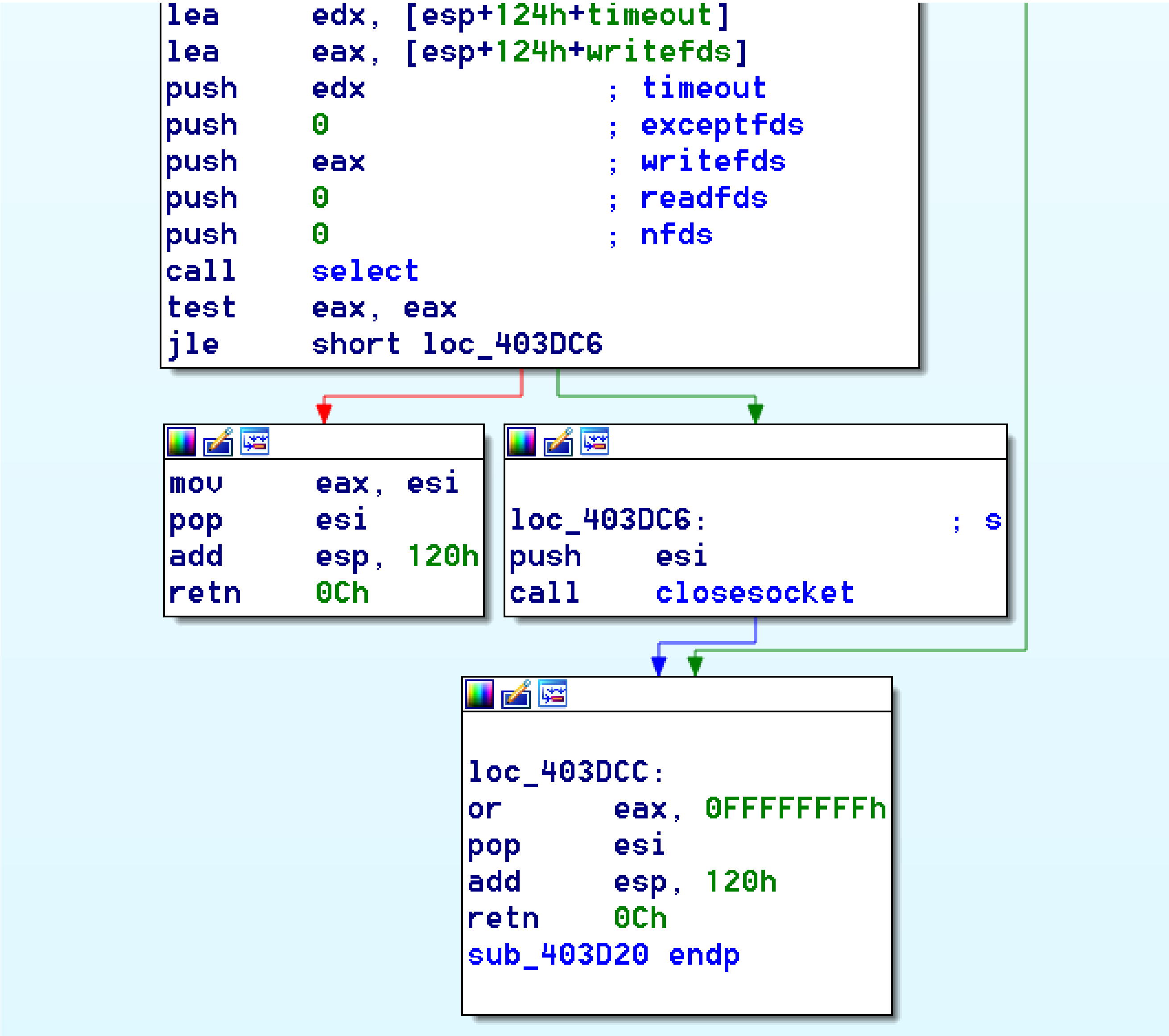


```
; int __stdcall sub_403D20(int, u_short hostshort, int)
sub_403D20 proc near

argp= dword ptr -120h
timeout= timeval ptr -11Ch
name= sockaddr ptr -114h
writefds= fd_set ptr -104h
arg_0= dword ptr  4
hostshort= word ptr  8
```



```
arg_8= dword ptr  0Ch

sub     esp, 120h
mov     ecx, dword ptr [esp+120h+hostshort]
mov     eax, [esp+120h+arg_0]
push    esi
push    ecx             ; hostshort
mov     [esp+128h+argp], 1
mov     dword ptr [esp+128h+name.sa_data+2], eax
call    htons
push    6               ; protocol
push    1               ; type
push    2               ; af
mov     word ptr [esp+130h+name.sa_data], ax
mov     [esp+130h+name.sa_family], 2
call    socket
mov     esi, eax
cmp     esi, 0FFFFFFFFh
jz      short loc_403DCC
```

```
lea     edx, [esp+124h+argp]
push    edx             ; argp
push    8004667Eh       ; cmd
push    esi             ; s
call    ioctlsocket
mov     eax, [esp+124h+arg_8]
lea     ecx, [esp+124h+name]
push    10h             ; namelen
push    ecx             ; name
push    esi             ; s
mov     [esp+130h+writefds.fd_array], esi
mov     [esp+130h+writefds.fd_count], 1
mov     [esp+130h+timeout.tv_sec], eax
mov     [esp+130h+timeout.tv_usec], 0
call    connect
```



```
lea     edx, [esp+124h+timeout]
lea     eax, [esp+124h+writefds]
push    edx                 ; timeout
push    0                   ; exceptfds
push    eax                 ; writefds
push    0                   ; readfds
push    0                   ; nfds
call    select
test    eax, eax
jle     short loc_403DC6
```

```
mov     eax, esi
pop     esi
add     esp, 120h
retn    0Ch
```

```

loc_403DC6:                 ; s
push    esi
call    closesocket
```

```

loc_403DCC:
or      eax, 0FFFFFFFFh
pop     esi
add     esp, 120h
retn    0Ch
sub_403D20 endp
```

Figure 5: An SMB module common to several attacks.

A lot has been written about WannaCry. As we analyze the code against our databases, we can draw the following overview:

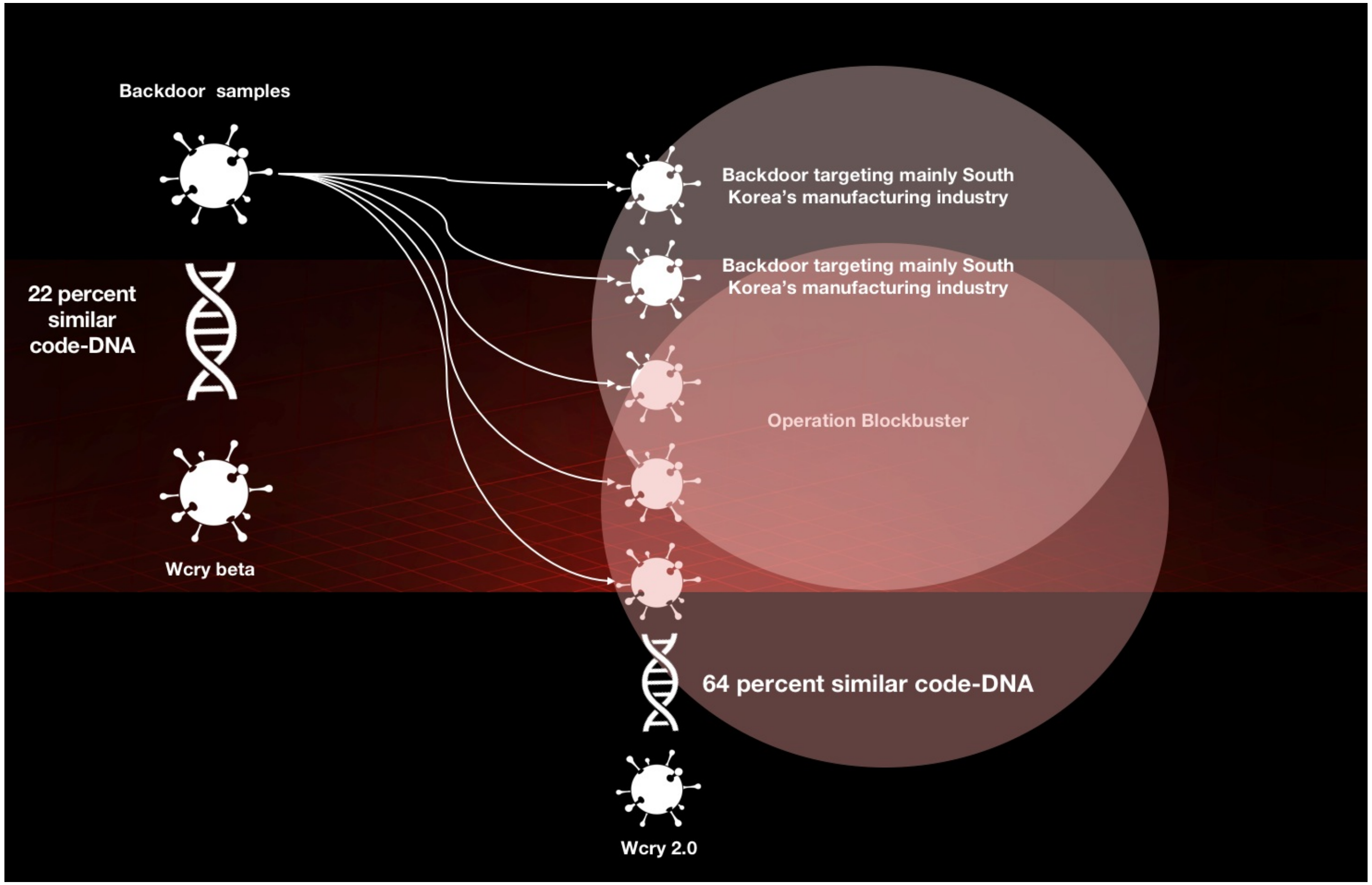
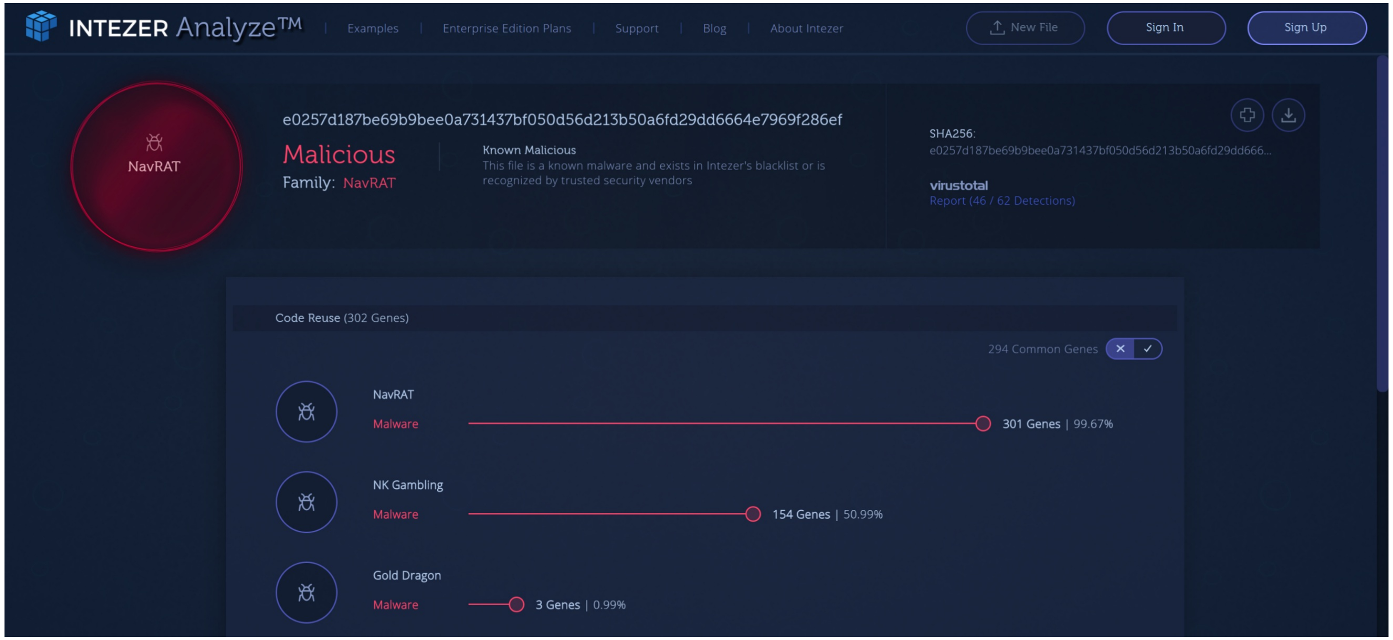

Backdoor samples
22 percent similar code-DNA
Wcry beta
Backdoor targeting mainly South Korea's manufacturing industry
Backdoor targeting mainly South Korea's manufacturing industry
Operation Blockbuster
64 percent similar code-DNA
Wcry 2.0

Figure 6: WannaCry code comparison overview.

For our research we compared the three major variants of WannaCry. An early release, called a beta, from February 2017, one from April, and the infamous one that hit the world in May.

2. Common file mapping

The second example demonstrates code responsible for mapping a file and using the XOR key 0xDEADBEEF on the first four bytes of the file. This code has appeared in the malware families NavRAT and Gold Dragon, plus a certain DLL from the South Korean gambling hacking campaign. These three RATs are thought to be affiliated with North Korea’s Group 123. NavRAT and the gambling DLL share more code, making them closer variants.



e0257d187be69b9bee0a731437bf050d56d213b50a6fd29dd6664e7969f286ef

Malicious
Family: NavRAT

Known Malicious
This file is a known malware and exists in Intezer's blacklist or is recognized by trusted security vendors

SHA256:
e0257d187be69b9bee0a731437bf050d56d213b50a6fd29dd666...

virustotal
Report (46 / 62 Detections)

Code Reuse (302 Genes)

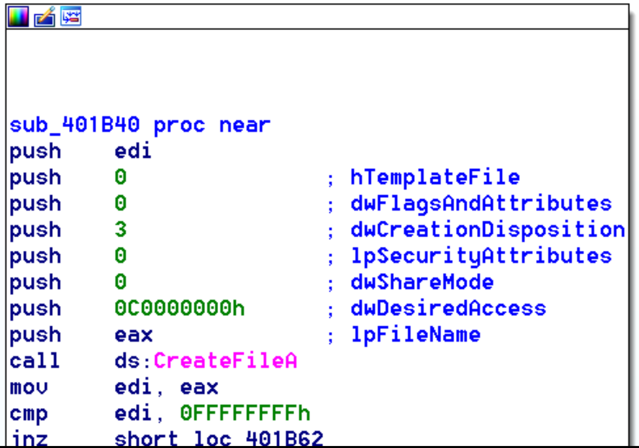
294 Common Genes

NavRAT
Malware
301 Genes | 99.67%

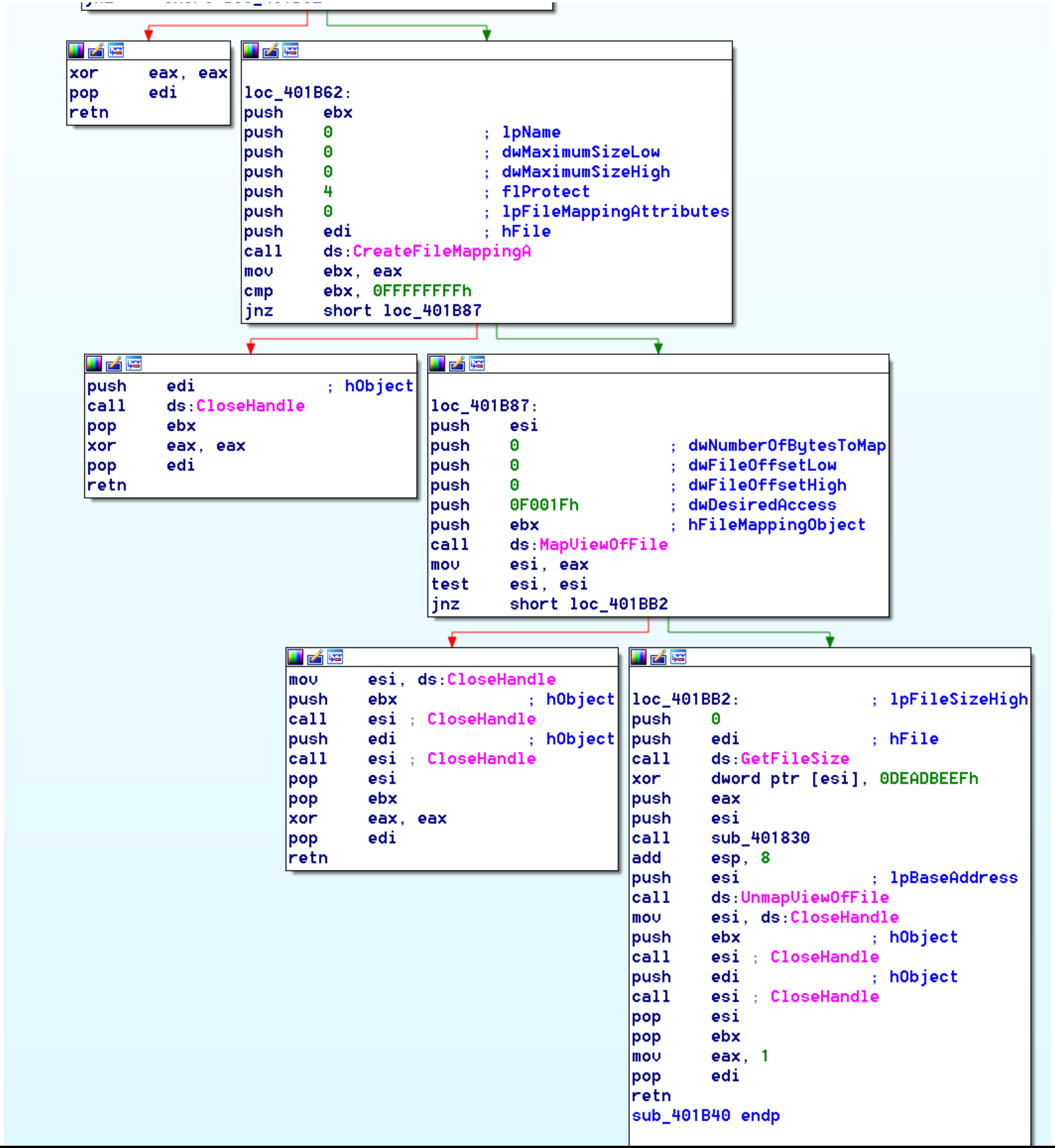
NK Gambling
Malware
154 Genes | 50.99%

Gold Dragon
Malware
3 Genes | 0.99%

Figure 7: Code overlap in a NavRAT sample.



```
sub_401B40 proc near
push    edi
push    0               ; hTemplateFile
push    0               ; dwFlagsAndAttributes
push    3               ; dwCreationDisposition
push    0               ; lpSecurityAttributes
push    0               ; dwShareMode
push    0C0000000h      ; dwDesiredAccess
push    eax             ; lpFileName
call    ds:CreateFileA
mov     edi, eax
cmp     edi, 0FFFFFFFFh
jnz     short loc_401B62
```



```
jnz     short loc_401B62
```

```
xor     eax, eax
pop     edi
retn
```

```
loc_401B62:
push    ebx
push    0               ; lpName
push    0               ; dwMaximumSizeLow
push    0               ; dwMaximumSizeHigh
push    4               ; flProtect
push    0               ; lpFileMappingAttributes
push    edi             ; hFile
call    ds:CreateFileMappingA
mov     ebx, eax
cmp     ebx, 0FFFFFFFFh
jnz     short loc_401B87
```

```
push    edi             ; hObject
call    ds:CloseHandle
pop     ebx
xor     eax, eax
pop     edi
retn
```

```
loc_401B87:
push    esi
push    0               ; dwNumberOfBytesToMap
push    0               ; dwFileOffsetLow
push    0               ; dwFileOffsetHigh
push    0F001Fh         ; dwDesiredAccess
push    ebx             ; hFileMappingObject
call    ds:MapViewOfFile
mov     esi, eax
test    esi, esi
jnz     short loc_401BB2
```

```
mov     esi, ds:CloseHandle
push    ebx             ; hObject
call    esi ; CloseHandle
push    edi             ; hObject
call    esi ; CloseHandle
pop     esi
pop     ebx
xor     eax, eax
pop     edi
retn
```

```
loc_401BB2:             ; lpFileSizeHigh
push    0
push    edi             ; hFile
call    ds:GetFileSize
xor     dword ptr [esi], 0DEADBEEFh
push    eax
push    esi
call    sub_401830
add     esp, 8
push    esi             ; lpBaseAddress
call    ds:UnmapViewOfFile
mov     esi, ds:CloseHandle
push    ebx             ; hObject
call    esi ; CloseHandle
push    edi             ; hObject
call    esi ; CloseHandle
pop     esi
pop     ebx
mov     eax, 1
pop     edi
retn
sub_401B40 endp
```

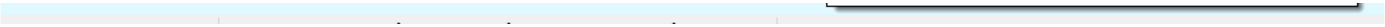
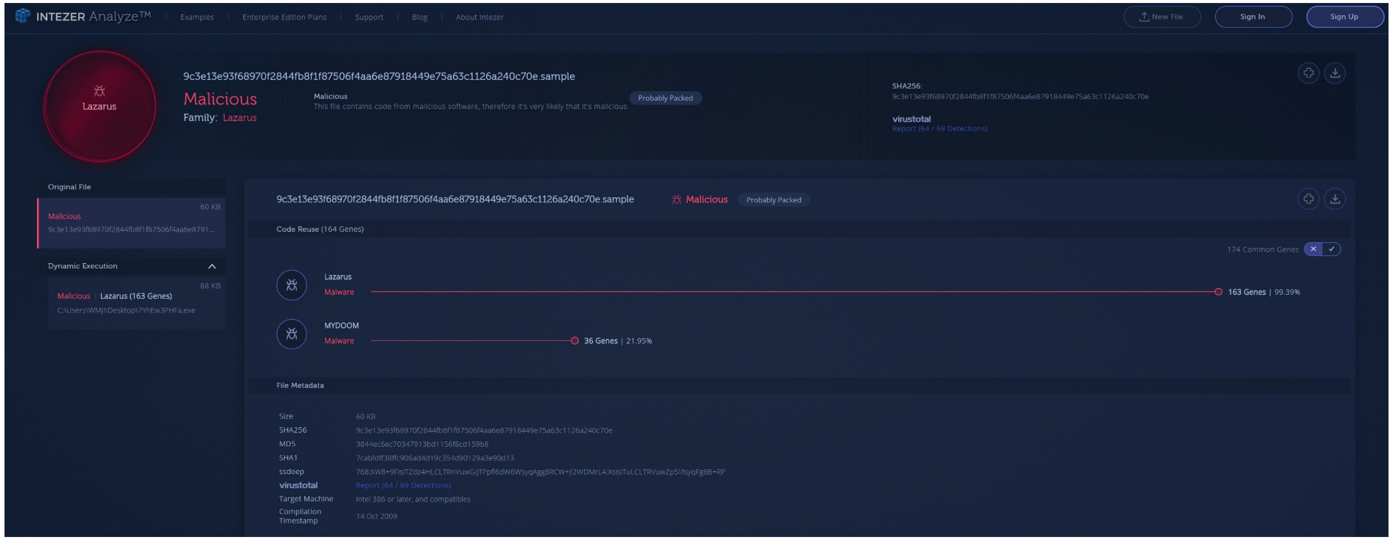


Figure 8: File-mapping code

3. Unique net share

The third example, responsible for launching a cmd.exe with a net share, has been seen in 2009's Brambul, also known as SierraBravo, as well as KorDllBot in 2011. These malware families are also attributed to the Lazarus group.



9c3e13e93f68970f2844fb8f1f87506f4aa6e87918449e75a63c1126a240c70e.sample

Malicious

Family: Lazarus

Malicious
This file contains code from malicious software, therefore it's very likely that it's malicious.

Probably Packed

SHA256:
9c3e13e93f68970f2844fb8f1f87506f4aa6e87918449e75a63c1126a240c70e

virustotal
Report (64 / 69 Detections)

Original File
60 KB
Malicious
9c3e13e93f68970f2844fb8f1f87506f4aa6e8791...

Dynamic Execution
88 KB
Malicious | Lazarus (163 Genes)
C:\Users\WMJI\Desktop\7YhEw3PHFa.exe

9c3e13e93f68970f2844fb8f1f87506f4aa6e87918449e75a63c1126a240c70e.sample Malicious Probably Packed

Code Reuse (164 Genes)

174 Common Genes

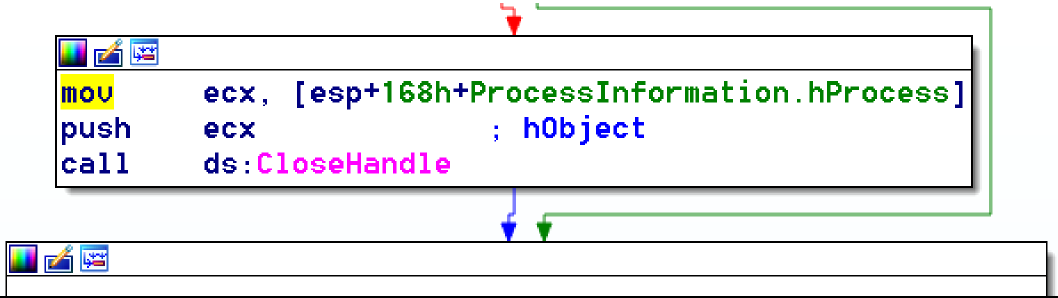
Lazarus
Malware
163 Genes | 99.39%

MYDOOM
Malware
36 Genes | 21.95%

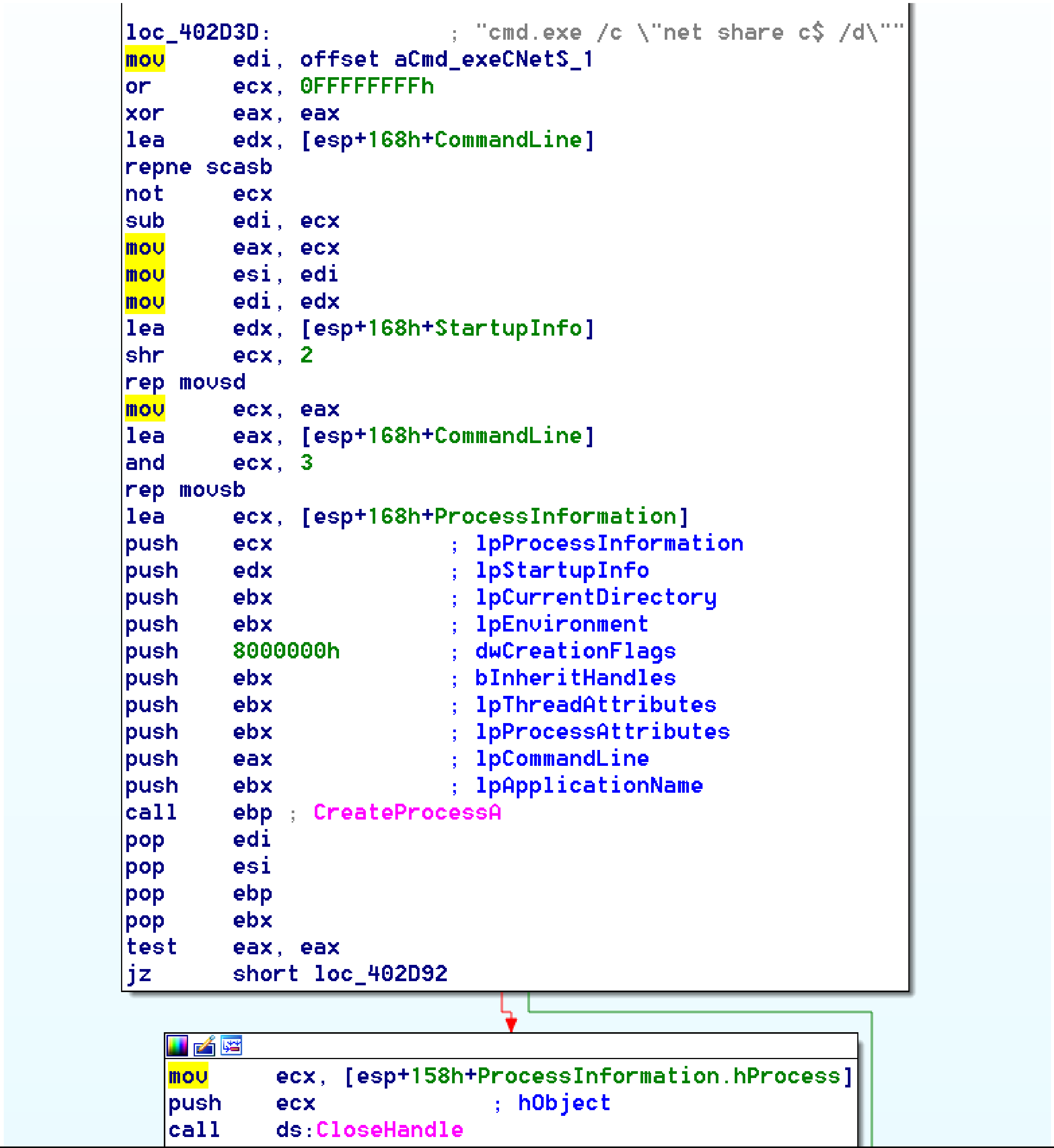
File Metadata

Size	60 KB
SHA256	9c3e13e93f68970f2844fb8f1f87506f4aa6e87918449e75a63c1126a240c70e
MD5	3844ec6ec70347913bd1156f8cd159b8
SHA1	7cabfdff38ffc906ad4d19c354d90129a3e90d13
ssdeep	768:XW8+9FisiTZdz4HLCLTRnVuwGiJTPpfl6dW6WsyqAgg8RCW+jl2WDMrL4:XsisiTuLCLTRVuwZp5l/lsyqFg8B+RP
virustotal	Report (64 / 69 Detections)
Target Machine	Intel 386 or later, and compatibles
Compilation Timestamp	14 Oct 2009

Figure 9: Code overlap of a SierraBravo (Brambul) sample.

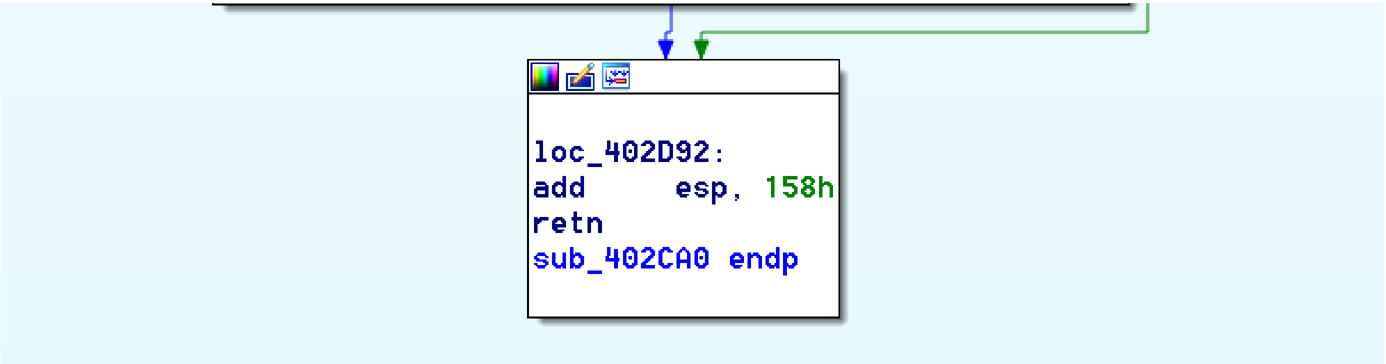


```
mov     ecx, [esp+168h+ProcessInformation.hProcess]
push    ecx             ; hObject
call    ds:CloseHandle
```



```
loc_402D3D:                     ; "cmd.exe /c \"net share c$ /d\""
mov     edi, offset aCmd_exeCNet$_1
or      ecx, 0FFFFFFFFh
xor     eax, eax
lea     edx, [esp+168h+CommandLine]
repne scasb
not     ecx
sub     edi, ecx
mov     eax, ecx
mov     esi, edi
mov     edi, edx
lea     edx, [esp+168h+StartupInfo]
shr     ecx, 2
rep movsd
mov     ecx, eax
lea     eax, [esp+168h+CommandLine]
and     ecx, 3
rep movsb
lea     ecx, [esp+168h+ProcessInformation]
push    ecx                     ; lpProcessInformation
push    edx                     ; lpStartupInfo
push    ebx                     ; lpCurrentDirectory
push    ebx                     ; lpEnvironment
push    8000000h                ; dwCreationFlags
push    ebx                     ; bInheritHandles
push    ebx                     ; lpThreadAttributes
push    ebx                     ; lpProcessAttributes
push    eax                     ; lpCommandLine
push    ebx                     ; lpApplicationName
call    ebp ; CreateProcessA
pop     edi
pop     esi
pop     ebp
pop     ebx
test    eax, eax
jz      short loc_402D92
```

```
mov     ecx, [esp+158h+ProcessInformation.hProcess]
push    ecx                     ; hObject
call    ds:CloseHandle
```



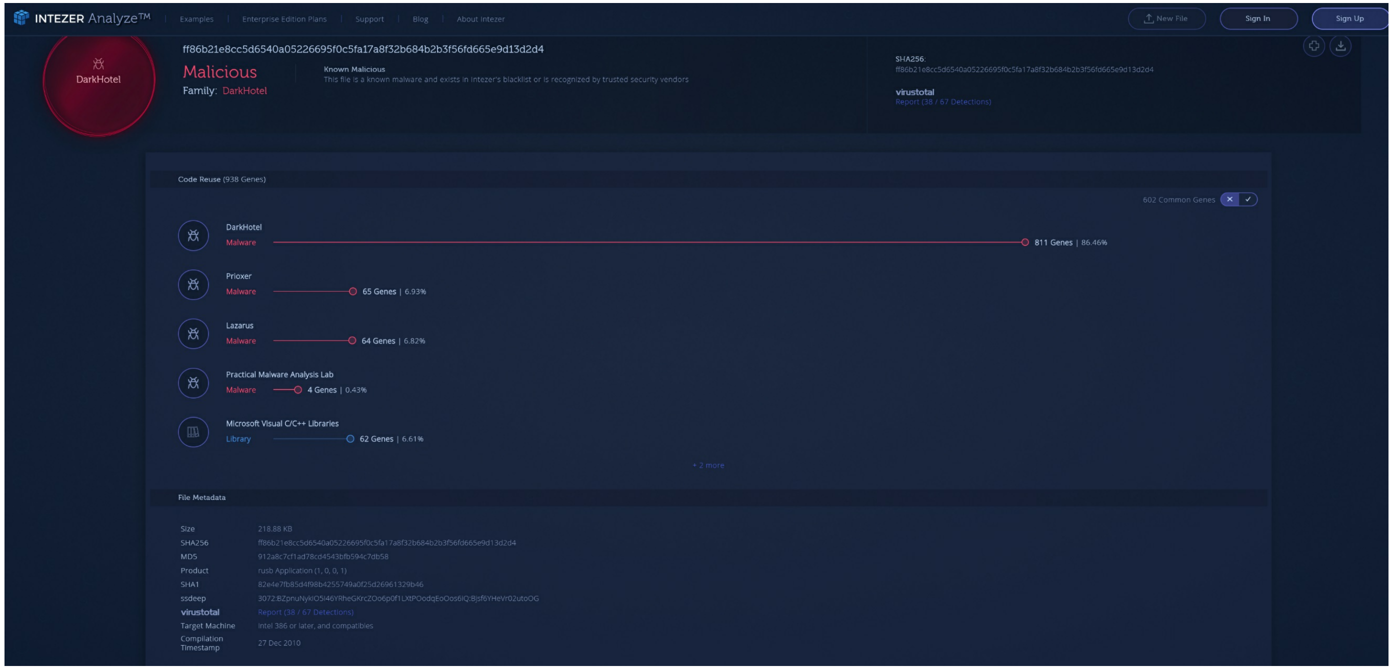
```
loc_402D92:
add     esp, 158h
retn
sub_402CA0 endp
```

Figure 10: A code block reused in the malware families Brambul/SierraBravo and KorDllBot.

4. Operation Dark Hotel

In 2014, Kaspersky reported a more than seven-year campaign against Asian hotels, in which the adversaries used an arsenal of tools to break into the computers of hotel visitors. Zero days and control servers were used, along with the malware family Tapaoux, or DarkHotel, according to the report.

While we examined the DPRK samples, we noticed a hit with the Dark Hotel samples in our collections. By going through the code, we noticed several pieces of code overlap and reuse, for example, with samples from Operation Troy.



DarkHotel

ff86b21e8cc5d6540a05226695f0c5fa17a8f32b684b2b3f56fd665e9d13d2d4

Malicious

Family: DarkHotel

Known Malicious
This file is a known malware and exists in Intezer's blacklist or is recognized by trusted security vendors

SHA256:
ff86b21e8cc5d6540a05226695f0c5fa17a8f32b684b2b3f56fd665e9d13d2d4

virustotal
Report (38 / 67 Detections)

Code Reuse (938 Genes)

602 Common Genes ✕ ✓

DarkHotel
Malware — 811 Genes | 86.46%

Prioxer
Malware — 65 Genes | 6.93%

Lazarus
Malware — 64 Genes | 6.82%

Practical Malware Analysis Lab
Malware — 4 Genes | 0.43%

Microsoft Visual C/C++ Libraries
Library — 62 Genes | 6.61%

+ 2 more

File Metadata

Size	218.88 KB
SHA256	ff86b21e8cc5d6540a05226695f0c5fa17a8f32b684b2b3f56fd665e9d13d2d4
MD5	912a8c7cf1ad78cd4543bfb594c7db58
Product	rusb Application (1, 0, 0, 1)
SHA1	82e4e7fb85d4f98b4255749a0f25d26961329b46
ssdeep	3072:BZpnuNykIO5I46YRheGKrcZOo6p0f1LXtPOodqEoOos6iQ:Bjsf6YHeVr02utoOG
virustotal	Report (38 / 67 Detections)
Target Machine	Intel 386 or later, and compatibles
Compilation Timestamp	27 Dec 2010

Figure 11: Code overlap in a Dark Hotel sample.

Identifying a group

By applying what we learned from our comparisons and code-block identifications, we uncovered possible new links between malware families and the groups using them.

With the different pieces of malware we have analyzed, we can illustrate the code reuse and sharing between the groups known to be affiliated with North Korea.

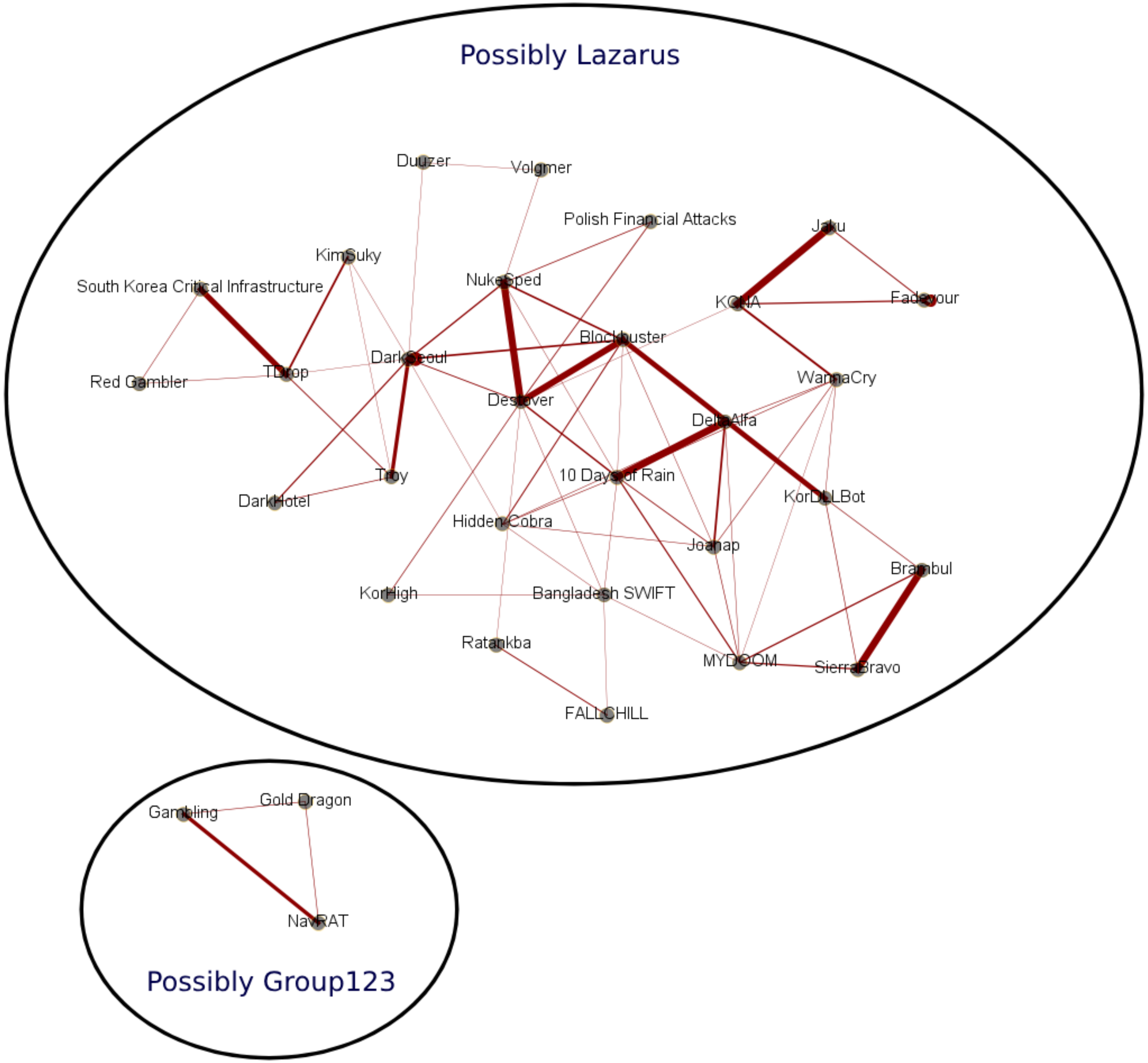


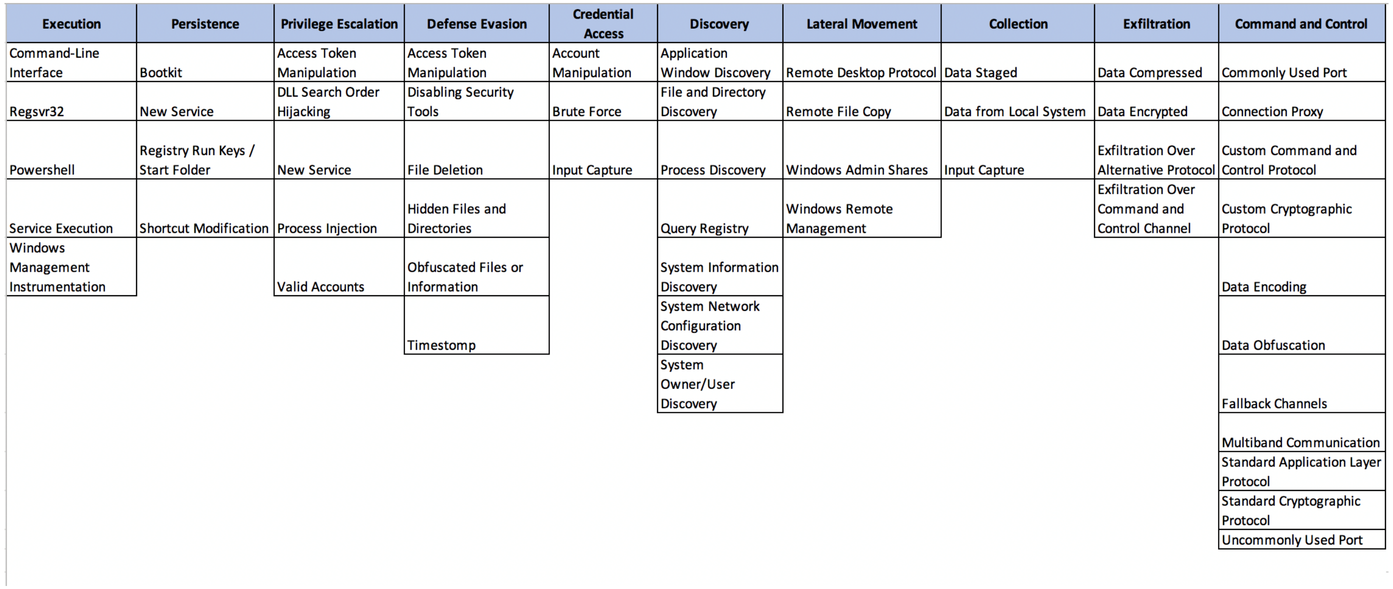
Figure 12: Groups and families linked by code reuse.

The malware attributed to the group Lazarus has code connections that link many of the malware families spotted over the years. Lazarus is a collective name for many DPRK cyber operations, and we clearly see links between malware families used in different campaigns.

The malware (NavRAT, gambling, and Gold Dragon) possibly created by Group 123 are connected to each other but are separate from those used by Lazarus. Although these are different units focusing on different areas, there seems to be a parallel structure in which they collaborate during certain campaigns.

MITRE ATT&CK

From our research of these malware samples, we can identify the following techniques used by the malware families:



Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Exfiltration	Command and Control
Command-Line Interface	Bootkit	Access Token Manipulation	Access Token Manipulation	Account Manipulation	Application Window Discovery	Remote Desktop Protocol	Data Staged	Data Compressed	Commonly Used Port
Regsvr32	New Service	DLL Search Order Hijacking	Disabling Security Tools	Brute Force	File and Directory Discovery	Remote File Copy	Data from Local System	Data Encrypted	Connection Proxy
Powershell	Registry Run Keys / Start Folder	New Service	File Deletion	Input Capture	Process Discovery	Windows Admin Shares	Input Capture	Exfiltration Over Alternative Protocol	Custom Command and Control Protocol
Service Execution	Shortcut Modification	Process Injection	Hidden Files and Directories		Query Registry	Windows Remote Management		Exfiltration Over Command and Control Channel	Custom Cryptographic Protocol
Windows Management Instrumentation		Valid Accounts	Obfuscated Files or Information		System Information Discovery				Data Encoding
			Timestomp		System Network Configuration Discovery				Data Obfuscation
					System Owner/User Discovery				Fallback Channels
									Multiband Communication
									Standard Application Layer Protocol
									Standard Cryptographic Protocol
									Uncommonly Used Port

When we zoom in on the Discovery category in the MITRE model, for example, we notice that the techniques are typical for first-stage dropper malware. The adversary drops these samples on victims’ machines and collects information on where they landed in the victims’ networks and which user/access rights they gained.

In 2018, we saw examples of campaigns in which attackers used PowerShell to download and execute these droppers. Once information has been sent to a control server, the adversary determines the next steps, which often include installing a remote access tool to enable lateral movement on the network and pursue the goals of the campaign.

Final words

Security vendors and researchers often use different names when speaking about the same malware, group, or attack. This habit makes it challenging to group all the malware and campaigns. By taking a scientific approach, such as looking for code reuse, we can categorize our findings. We believe our research will help the security community organize the current “mess” we face in relation to North Korean malware and campaigns.

We clearly saw a lot of code reuse over the many years of cyber campaigns we examined. This indicates the North Koreans have groups with different skills and tools that execute their focused parts of cyber operations while also working in parallel when large campaigns require a mix of skills and tools.

We found our months of research, data gathering, and analysis very satisfying. By combining our skills, data, and technology, we were able to draw connections and reveal links that we had not seen before. The cybersecurity industry would greatly benefit from more collaboration and sharing of information, and we hope that this effort between McAfee and Intezer will inspire the community to work together more often.

The authors thank Costin Raiu for providing them with samples they did not have in their collections.

Sources

Glenn Simpson, Gordon Fairclough, and Jay Solomon, “U.S. Probes Banks’ North Korea Ties.” Wall Street Journal, last updated September 8, 2005.

Christiaan Beek, “Attacks on SWIFT Banking system benefit from insider knowledge.” https://securingtomorrow.mcafee.com/mcafee-labs/attacks-swift-banking-system-benefit-insider-knowledge/

Atif Mushtaq, “DDOS Madness Continued…” https://www.fireeye.com/blog/threat-research/2009/07/ddos-madness-climax.html

Ryan Sherstobitoff and Jessica Saavedra-Morales, “Gold Dragon Widens Olympics Malware Attacks, Gains Permanent Presence on Victims’ Systems.” https://securingtomorrow.mcafee.com/mcafee-labs/gold-dragon-widens-olympics-malware-attacks-gains-permanent-presence-on-victims-systems/

Alex Drozhzhin, “Darkhotel: a spy campaign in luxury Asian hotels.” https://www.kaspersky.com/blog/darkhotel-apt/6613/

Warren Mercer, Paul Rascagneres, and Jungsoo An, “NavRAT Uses US-North Korea Summit As Decoy For Attacks In South Korea.” https://blog.talosintelligence.com/2018/05/navrat.html

Sergei Shevchenko and Adrian Nish**, “**Cyber Heist Attribution.**”** https://baesystemsai.blogspot.com/2016/05/cyber-heist-attribution.html

Mydoom code reuse report. https://analyze.intezer.com/#/analyses/113ba80f-1680-43d7-b287-cc62f3740fad

NavRAT code reuse report. https://analyze.intezer.com/#/analyses/4f19fd5a-a898-4fdf-96c9-d3a4aad817cb

SierraBravo code reuse report. https://analyze.intezer.com/#/analyses/8da8104e-56e4-49fd-ba24-82978bc1610c

Dark Hotel code reuse report. https://analyze.intezer.com/#/analyses/c034e0fe-7825-4f6d-b092-7c5ee693aff4

Kang Jang-ho, "A foreign currency earned with a virtual currency … What is the life of a North Korean hacker?" http://m.mtn.co.kr/news/news_view.php?mmn_idx=2018062517065863930#_enliple

Awesome work by the team responsible for the "Operation Blockbuster" report. https://www.operationblockbuster.com/resources/

‹ Previous Article Next Article ›

Categories: McAfee Labs
Tags: advanced persistent threats, Advanced Threat Research, cybersecurity, hacking

Leave a reply

Facebook Comments (0) Comments (0) G+ Comments

0 Comments

Sort by Oldest ⬍

Add a comment...

Facebook Comments Plugin

Leave a Comment

Comment

Name *

Email *

Please enter an answer in digits:

5 × 3 =

Post Comment

☒

Search

Business

Consumer

Executive Perspectives

McAfee Partners

McAfee Labs

Languages

English

Follow us

About

Subscribe

Contact & Media Requests

Privacy Policy

Legal