bio-isac

Creating Collaborative Threat
Intelligence for the Bioeconomy

Tardigrade:
APT Attack on the Bioeconomy

(Bulz.253748 Variant Overview: intserrs644.dll)

Callie Churchwell – Senior Digital Biosecurity Analyst

Charles Fracchia – VP of Data at Dotmatics & CEO/Founder BioBright

Ed Chung, MD – Digital Biosecurity Lead & CMO BioBright

Website: https://isac.bio
Contact: tips@isac.bio
PGP: EB2A A1AA 4AD9 4A0B BE07 E8A9 625B BFEE 2E95 C7FC

contributed by: BioBright
a dotmatics company

Bottom Line Up Front – Impact

- Extremely sophisticated malware actively spreading in the bioeconomy
- Metamorphic version of the SmokeLoader family
- Potentially the first identified malware with this level of sophistication targeting biomanufacturing facilities
- This is ongoing and this disclosure was accelerated in the public interest given the observed spread

Bottom Line Up Front – Motivations and Targets

Targets:

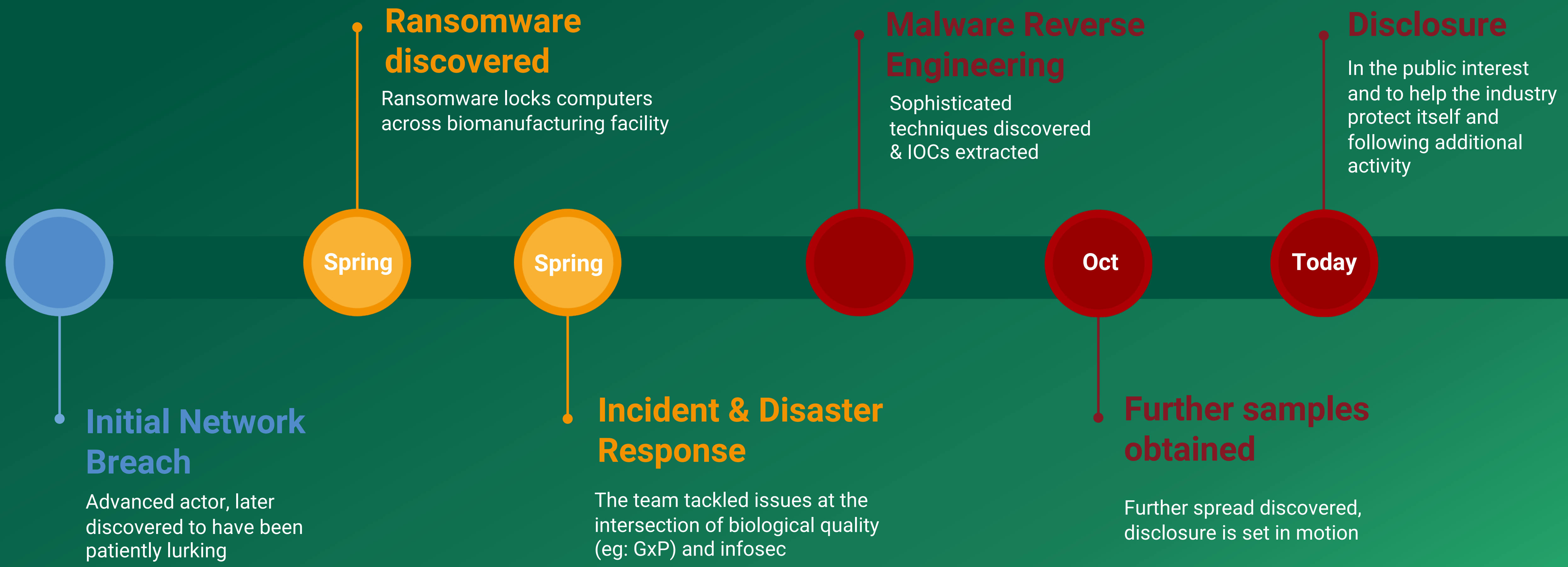
- Bioeconomy companies
- Biomanufacturing sector
- Possibly targeted based on public / news activity

Motivations (based on activity):

- Intellectual property theft
- Persistence
- Ransomware preparation

Tardigrade APT attack on Bioeconomy infrastructure

Timeline of Discovery



Initial Network Breach

Advanced actor, later discovered to have been patiently lurking

Spring

Ransomware discovered

Ransomware locks computers across biomanufacturing facility

Spring

Incident & Disaster Response

The team tackled issues at the intersection of biological quality (eg: GxP) and infosec

Malware Reverse Engineering

Sophisticated techniques discovered & IOCs extracted

Oct

Further samples obtained

Further spread discovered, disclosure is set in motion

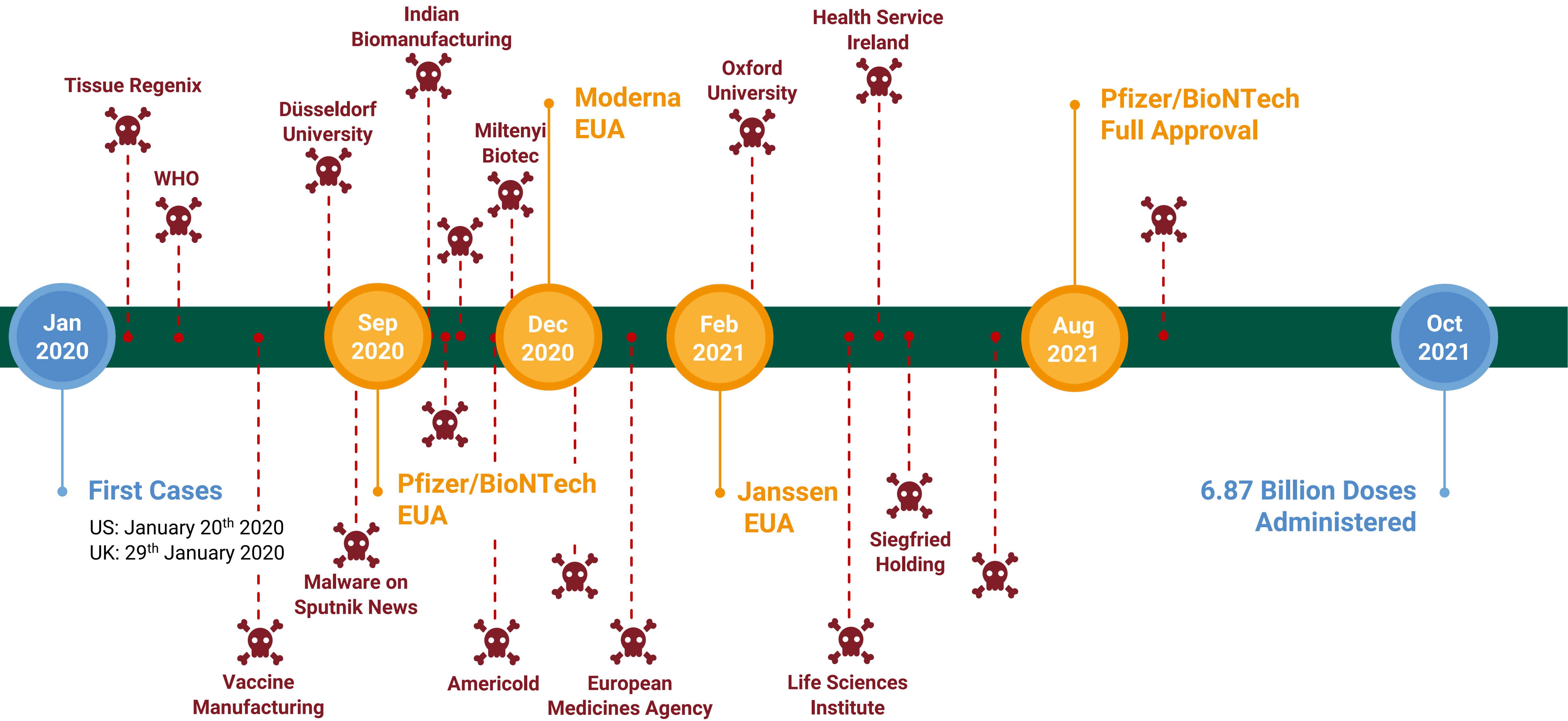
Today

Disclosure

In the public interest and to help the industry protect itself and following additional activity

contributed by: BioBright a dotmatics company

Where are we today? - Attacks



Jan 2020

First Cases

US: January 20th 2020
UK: 29th January 2020

Tissue Regenix

WHO

Vaccine Manufacturing

Düsseldorf University

Sep 2020

Malware on Sputnik News

Pfizer/BioNTech EUA

Indian Biomanufacturing

Americold

Miltenyi Biotec

Moderna EUA

Dec 2020

European Medicines Agency

Feb 2021

Oxford University

Janssen EUA

Life Sciences Institute

Health Service Ireland

Siegfried Holding

Aug 2021

Pfizer/BioNTech Full Approval

Oct 2021

6.87 Billion Doses Administered

SmokeLoader / Bulz / Dofoil background

- SmokeLoader - Smokey Bear family, Loader/Trojan
- Purpose: To inject a **more effective and destructive malware** into the machine
- Smokey Bear Family is constantly automizing techniques/tactics
- Smokey Bear Focus: multi-purpose tools that include keylogging, information theft, botnet support, and backdoor access
- Attack Delivery: Infected email software, plug-ins, adverts, infected networks, physical infections (USB).

Background Information – Tardigrade

Tagged Bulz.Method:253748 Ransomware Trojans

- First Variant: SmokeLoader
- Suspected Second Variant: Dofoil

Attack delivery:

- USB, Files, Network autonomously
- Primary: Phishing

Goal:

- The main role of this malware is still to download, manipulate files, send main.dll library if possible, deploy other modules and remain hidden.
- Espionage, tunnel creation, carry a bigger payload.
- Compatible with other APT made payloads so far: Conti, Ryuk, Cobalt Strike

Malware Architecture and Capabilities

Metamorphic

- While many malware systems are polymorphic, this system seems to be able to recompile the loader from memory without leaving a consistent signature.
- Recompiling occurs after a network connection in the wild that could be a call to a command and control (CnC) server to download and execute the complier
- Allows the system to change portions/all the functions based on CnC like a normal loader system but with a level of autonomy that is unexpected

Minimum Supported System for Functions Performed

Minimum supported client	**Windows 2000 Professional [desktop apps only]**
Minimum supported server	Windows 2000 Server [desktop apps only]
Target Platform	Windows
Header	winbase.h (include Windows.h)
Library	Advapi32.lib
DLL	Advapi32.dll

Malware Dynamics – MITRE ATT&CK

Resource Development

- Stage Capabilities

Initial Access

- External Remote Services
- Phishing
- Replication Through Removable Media
- Supply Chain Compromise
- Valid Accounts

Execution

- Command and Scripting Interpreter
- Inter-Process Communication
- Scheduled Task/Job
- User Execution

Persistence

- Boot or Logon Autostart Execution
- Boot or Logon Initialization Scripts
- Browser Extensions
- Create or Modify System Process
- Event Triggered Execution
- External Remote Services
- Scheduled Task/Job
- Valid Accounts

Privilege Escalation

- Abuse Elevation Control Mechanism
- Boot or Logon Autostart Execution
- Boot or Logon Initialization Scripts
- Create or Modify System Process
- Event Triggered Execution
- Exploitation for Privilege Escalation
- Process Injection
- Scheduled Task/Job
- Valid Accounts

Defense Evasion

- Abuse Elevation Control Mechanism
- Deobfuscate/Decode Files or Information
- File and Directory Permissions Modification
- Hide Artifacts
- Impair Defenses
- Indicator Removal on Host
- Modify Registry
- Obfuscated Files or Information
- Process Injection
- Signed Binary Proxy Execution
- Valid Accounts
- Virtualization/Sandbox Evasion

Credential Access

- Credentials from Password Stores
- Unsecured Credentials

Malware Dynamics – MITRE ATT&CK

Discovery
File and Directory Discovery
Virtualization/Sandbox Evasion

Lateral Movement
Exploitation of Remote Services
Remote Service Session Hijacking
Remote Services
Replication Through Removable Media

Collection
Data from Local System
Data from Network Shared Drive
Data Staged
Email Collection

Exfiltration
Exfiltration Over C2 Channel

Command and Control	
Application Layer Protocol	Web Protocols
Ingress Tool Transfer	
Web Service	One-Way Communication

Impact	
Account Access Removal	
Data Manipulation	Runtime Data Manipulation
	Stored Data Manipulation
	Transmitted Data Manipulation

Malware Dynamics – High Level

Autonomy

- Previous SmokeLoader versions were externally directed, dependent on CnC infrastructure
- This “Tardigrade” version is far more autonomous, able to decide on lateral movement based on internal logic
- Significant level of autonomous decision-making ability, possibly on random wait times.
- The ability to selectively identify files for modification.

Privilege Escalation

- Uses impersonate client technique to gain Admin control

Connectivity

- Replaces Main.dll and attempts to export original to varying IPs that do not correlate with a specific CnC
- Traffic is encrypted and uses a diversity of methods (no more to share at this time)
- One method of lateral spread uses network shares and creates folders in CnC connected servers with random names (eg: ProfMargaretPredovic)

Indicators of Compromise (IoCs)

Websites Reached

- Random Batch of Amazon Web Services (AWS):
- GoDaddy
- Akamai

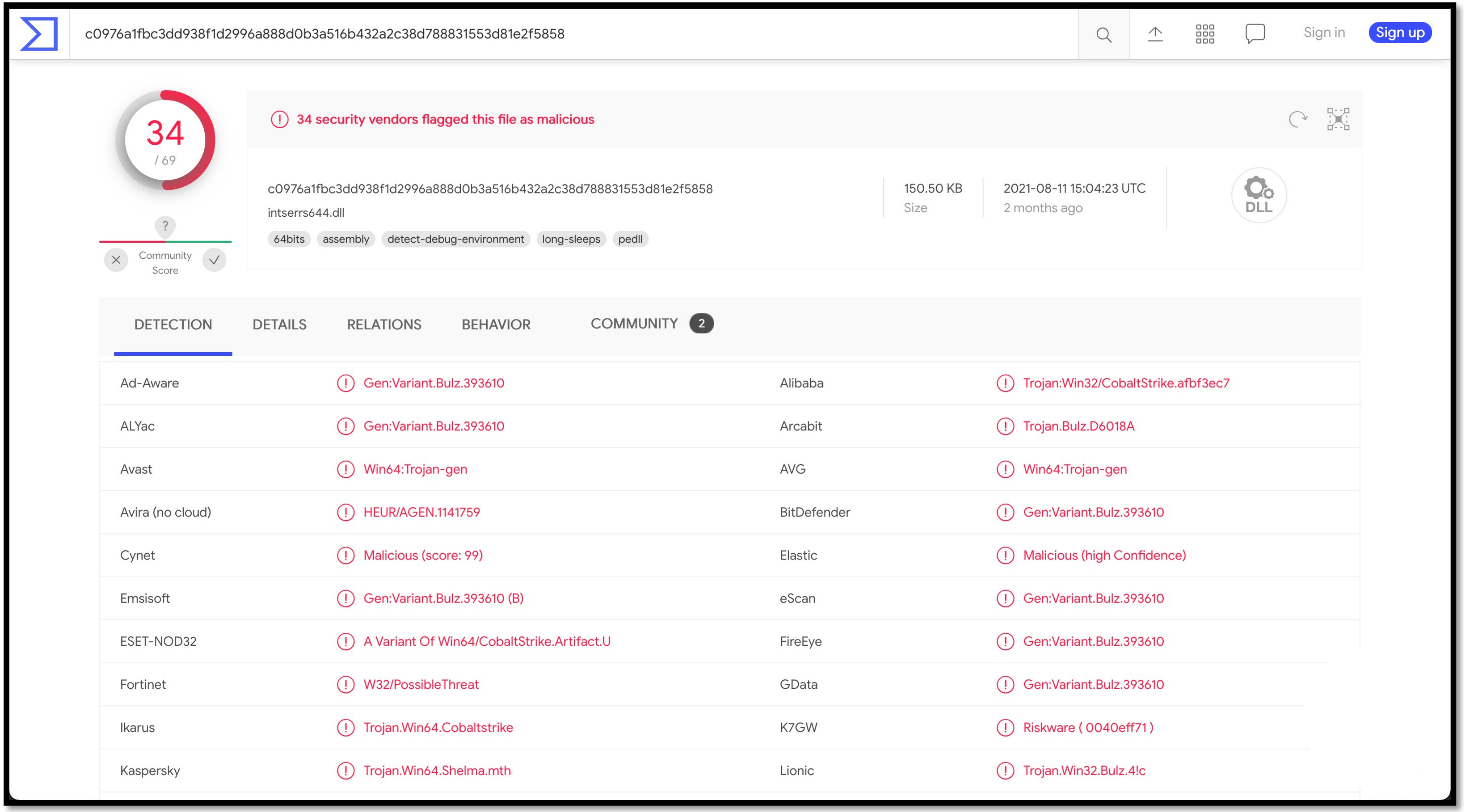
Exports

- DllGetClassObject
- DllMain
- DllRegisterServer
- DllUnregisterServer
- InitHelperDll
- StartW

"Out of band" behavior detection:

- Registry flushing, monitoring specific files

Detection status – 10/25/2021



c0976a1fbc3dd938f1d2996a888d0b3a516b432a2c38d788831553d81e2f5858

34 / 69

34 security vendors flagged this file as malicious

c0976a1fbc3dd938f1d2996a888d0b3a516b432a2c38d788831553d81e2f5858
intserrs644.dll

64bits assembly detect-debug-environment long-sleeps pedll

150.50 KB
Size

2021-08-11 15:04:23 UTC
2 months ago

DETECTION DETAILS RELATIONS BEHAVIOR COMMUNITY 2

Ad-Aware	Gen:Variant.Bulz.393610	Alibaba	Trojan:Win32/CobaltStrike.afbf3ec7
ALYac	Gen:Variant.Bulz.393610	Arcabit	Trojan.Bulz.D6018A
Avast	Win64:Trojan-gen	AVG	Win64:Trojan-gen
Avira (no cloud)	HEUR/AGEN.1141759	BitDefender	Gen:Variant.Bulz.393610
Cynet	Malicious (score: 99)	Elastic	Malicious (high Confidence)
Emsisoft	Gen:Variant.Bulz.393610 (B)	eScan	Gen:Variant.Bulz.393610
ESET-NOD32	A Variant Of Win64/CobaltStrike.Artifact.U	FireEye	Gen:Variant.Bulz.393610
Fortinet	W32/PossibleThreat	GData	Gen:Variant.Bulz.393610
Ikarus	Trojan.Win64.Cobaltstrike	K7GW	Riskware (0040eff71)
Kaspersky	Trojan.Win64.Shelma.mth	Lionic	Trojan.Win32.Bulz.4!c

VirusTotal: https://www.virustotal.com/gui/file/c0976a1fbc3dd938f1d2996a888d0b3a516b432a2c38d788831553d81e2f5858/detection

Static Analysis

Registry Actions

Registry Keys Opened

- HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Ext\Stats\{761497BB-D6F0-462C-B6EB-D4DAF1D92D43}\iexplore
- HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Ext\Stats\{DBC80044-A445-435B-BC74-9C25C1C588A9}\iexplore
- HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings
- HKEY_CURRENT_USER\SOFTWARE\Policies\Microsoft\Windows\CurrentVersion\Internet Settings
- HKEY_CURRENT_USER\Software
- HKEY_CURRENT_USER\Software\Microsoft\Internet Explorer\Download
- HKEY_CURRENT_USER\Software\Microsoft\Internet Explorer\Main
- HKEY_CURRENT_USER\Software\Microsoft\Internet Explorer\Main\FeatureControl
- HKEY_CURRENT_USER\Software\Microsoft\Internet Explorer\Main\FeatureControl\FEATURE_ALLOW_REVERSE_SOLIDUS_IN_USERINFO_KB932562
- HKEY_CURRENT_USER\Software\Microsoft\Internet Explorer\Main\FeatureControl\FEATURE_ALWAYS_USE_DNS_FOR_SPN_KB3022771

Registry Keys Deleted

- HKLM\SYSTEM\ControlSet001\Services\WmiApRpl\Performance\First Counter
- HKLM\SYSTEM\ControlSet001\Services\WmiApRpl\Performance\Last Counter
- HKLM\SYSTEM\ControlSet001\Services\WmiApRpl\Performance\First Help
- HKLM\SYSTEM\ControlSet001\Services\WmiApRpl\Performance\Last Help
- HKLM\SYSTEM\ControlSet001\Services\WmiApRpl\Performance\Object List

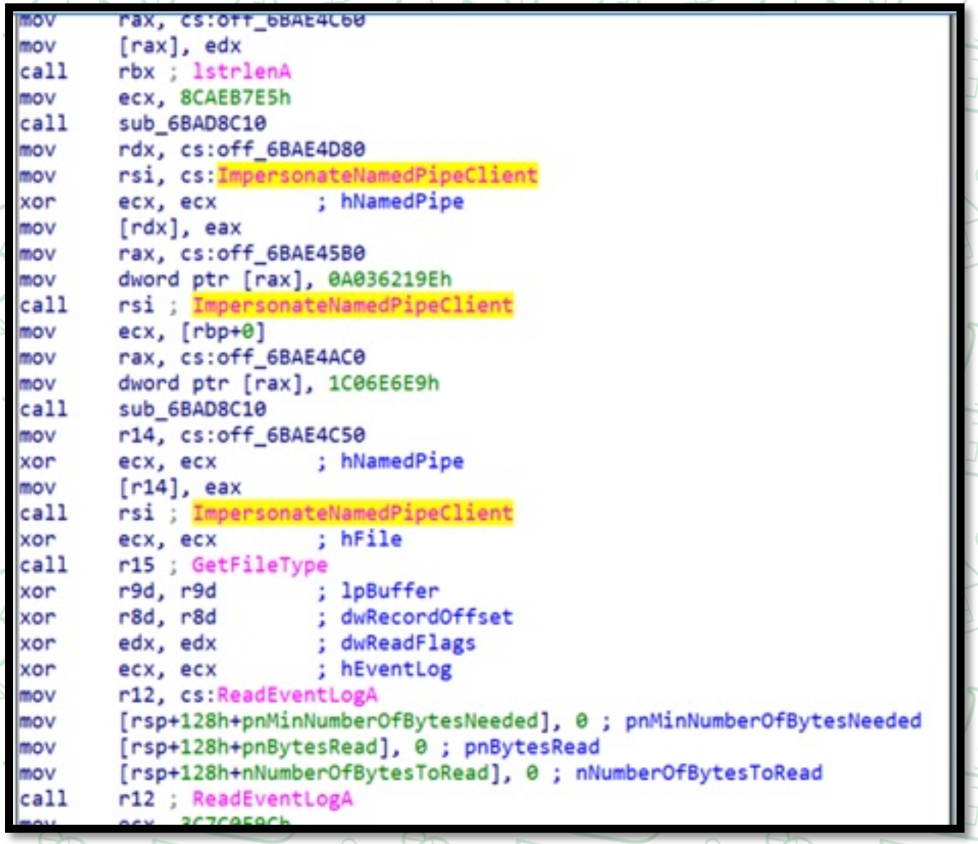
Process Termination Trees

```
2368 - %windir%\System32\svchost.exe -k WerSvcGroup

2912 - %windir%\system32\WerFault.exe -u -p 2728 -s 660

1028 - wmiadap.exe /F /T /R

2860 - %windir%\system32\DllHost.exe /Processid:{3EB3C877-1F16-487C-9050-104DBCD66683}

872 - %windir%\system32\wbem\wmiprvse.exe

2728 - %SANDBOX_DLL_LOADER_AMD64% %SAMPLEPATH% %WORKDIR% 483

2800 - cmd.exe /c echo kOJAdtQoDcMuogIZIl>"%TEMP%\DEM3504.tmp"&exit

2808 - %CONHOST% "110683108618492252-5067897321351265997-609353785-17193410651810180281534763768
```

Initial Actions

ImpersonateNamedPipeClient

- This function allows the server end of a named pipe to impersonate the client end.
- When this function is called, the named-pipe file system changes the thread of the calling process to start impersonating the security context of the last message read from the pipe.
- Only the server end of the pipe can call this function. The server can call the RevertToSelf function when the impersonation is complete

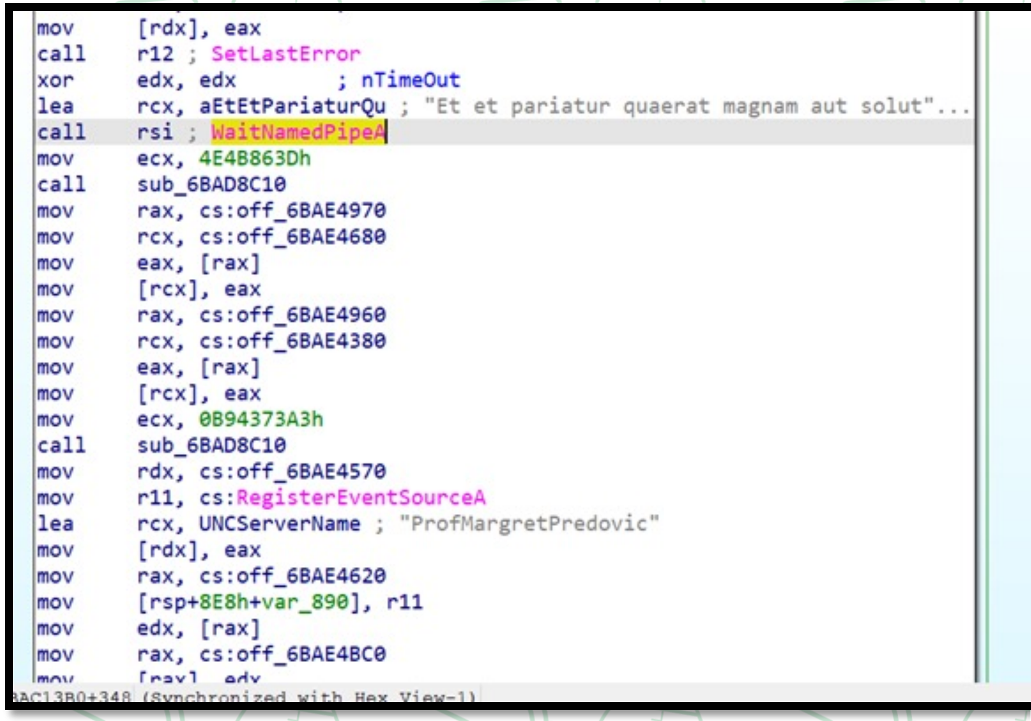


```
mov     rax, cs:off_6BAE4C60
mov     [rax], edx
call    rbx ; lstrlenA
mov     ecx, 8CAEB7E5h
call    sub_6BAD8C10
mov     rdx, cs:off_6BAE4D80
mov     rsi, cs:ImpersonateNamedPipeClient
xor     ecx, ecx        ; hNamedPipe
mov     [rdx], eax
mov     rax, cs:off_6BAE45B0
mov     dword ptr [rax], 0A036219Eh
call    rsi ; ImpersonateNamedPipeClient
mov     ecx, [rbp+0]
mov     rax, cs:off_6BAE4AC0
mov     dword ptr [rax], 1C06E6E9h
call    sub_6BAD8C10
mov     r14, cs:off_6BAE4C50
xor     ecx, ecx        ; hNamedPipe
mov     [r14], eax
call    rsi ; ImpersonateNamedPipeClient
xor     ecx, ecx        ; hFile
call    r15 ; GetFileType
xor     r9d, r9d        ; lpBuffer
xor     r8d, r8d        ; dwRecordOffset
xor     edx, edx        ; dwReadFlags
xor     ecx, ecx        ; hEventLog
mov     r12, cs:ReadEventLogA
mov     [rsp+128h+pnMinNumberOfBytesNeeded], 0 ; pnMinNumberOfBytesNeeded
mov     [rsp+128h+pnBytesRead], 0 ; pnBytesRead
mov     [rsp+128h+nNumberOfBytesToRead], 0 ; nNumberOfBytesToRead
call    r12 ; ReadEventLogA
```

Behavior in timing

WaitNamedPipeA

- Waits until either a time-out interval elapses or an instance of the specified named pipe is available for connection (that is, the pipe's server process has a pending ConnectNamedPipe operation on the pipe)
- Rax identifies that the malware is writing to a specific register

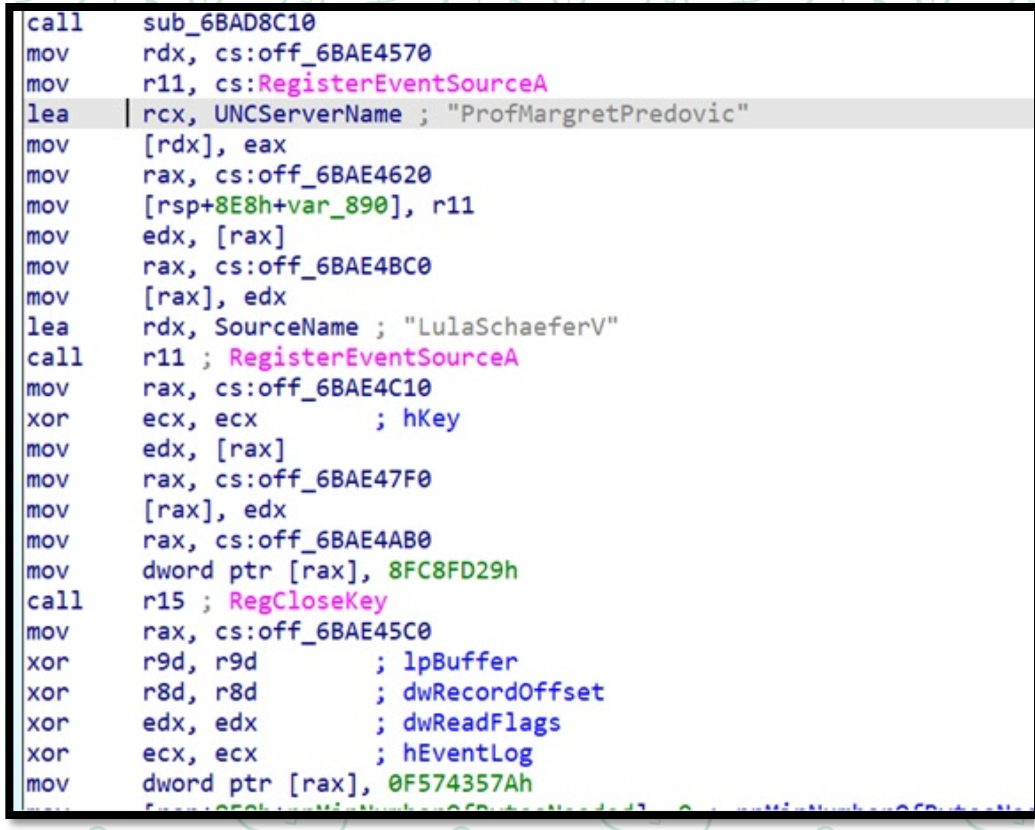


```
mov     [rdx], eax
call    r12 ; SetLastError
xor     edx, edx        ; nTimeOut
lea     rcx, aEtEtPariaturQu ; "Et et pariatur quaerat magnam aut solut"...
call    rsi ; WaitNamedPipeA
mov     ecx, 4E4B863Dh
call    sub_6BAD8C10
mov     rax, cs:off_6BAE4970
mov     rcx, cs:off_6BAE4680
mov     eax, [rax]
mov     [rcx], eax
mov     rax, cs:off_6BAE4960
mov     rcx, cs:off_6BAE4380
mov     eax, [rax]
mov     [rcx], eax
mov     ecx, 0B94373A3h
call    sub_6BAD8C10
mov     rdx, cs:off_6BAE4570
mov     r11, cs:RegisterEventSourceA
lea     rcx, UNCServerName ; "ProfMargretPredovic"
mov     [rdx], eax
mov     rax, cs:off_6BAE4620
mov     [rsp+8E8h+var_890], r11
mov     edx, [rax]
mov     rax, cs:off_6BAE4BC0
mov     [rax], edx
```

AC13B0+348 (Synchronized with Hex View-1)

Remote Server Behavior

- Retrieves a registered handle to the specified event log. The Universal Naming Convention (UNC) name of the remote server on which this operation is to be performed.
- The name of the event source whose handle is to be retrieved. The source name must be a subkey of a log under the **Eventlog** registry key.
- Note that the **Security** log is for system use only.

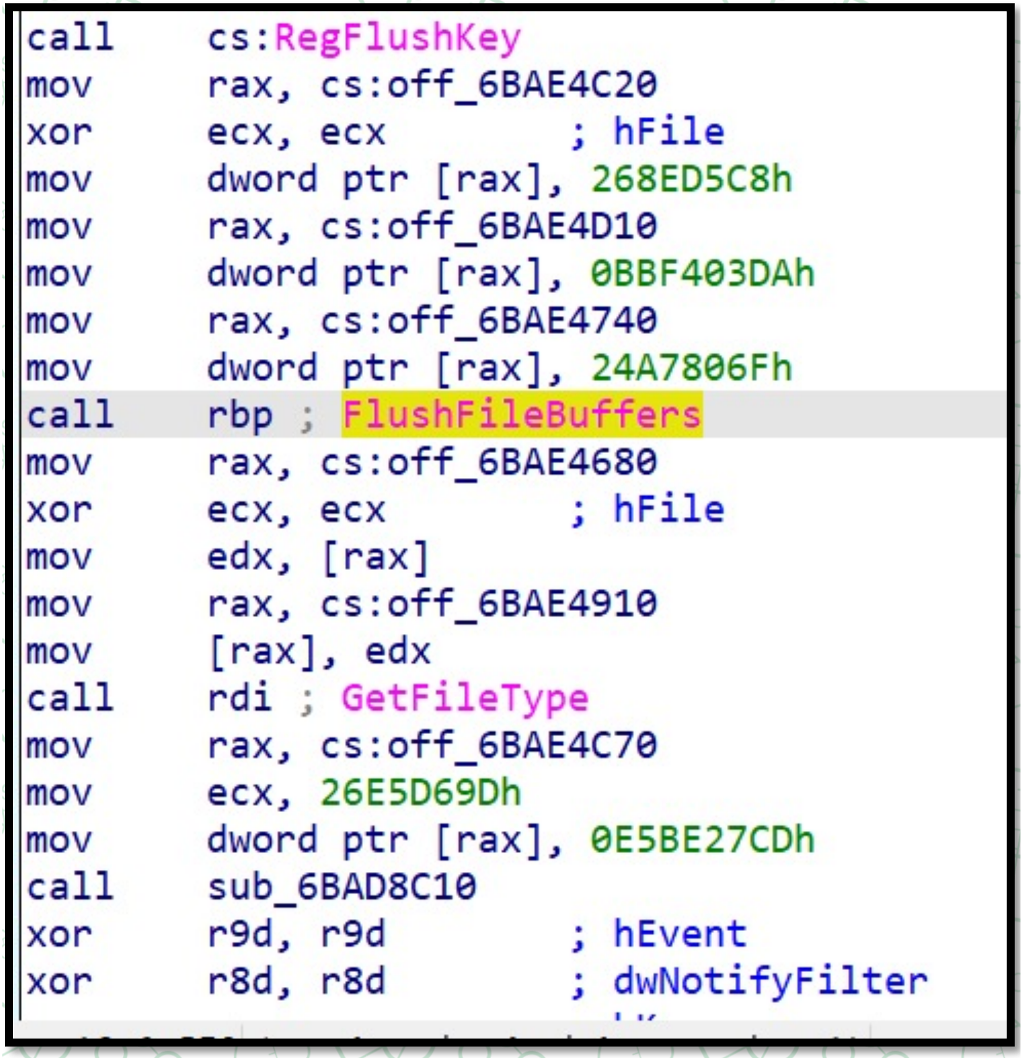


```
call    sub_6BAD8C10
mov     rdx, cs:off_6BAE4570
mov     r11, cs:RegisterEventSourceA
lea     rcx, UNCServerName ; "ProfMargretPredovic"
mov     [rdx], eax
mov     rax, cs:off_6BAE4620
mov     [rsp+8E8h+var_890], r11
mov     edx, [rax]
mov     rax, cs:off_6BAE4BC0
mov     [rax], edx
lea     rdx, SourceName ; "LulaSchaeferV"
call    r11 ; RegisterEventSourceA
mov     rax, cs:off_6BAE4C10
xor     ecx, ecx        ; hKey
mov     edx, [rax]
mov     rax, cs:off_6BAE47F0
mov     [rax], edx
mov     rax, cs:off_6BAE4AB0
mov     dword ptr [rax], 8FC8FD29h
call    r15 ; RegCloseKey
mov     rax, cs:off_6BAE45C0
xor     r9d, r9d        ; lpBuffer
xor     r8d, r8d        ; dwRecordOffset
xor     edx, edx        ; dwReadFlags
xor     ecx, ecx        ; hEventLog
mov     dword ptr [rax], 0F574357Ah
```

Flush File Function

FlushFileBuffers

- This function clears the buffers for the specified file and causes all buffered data to be written to the file

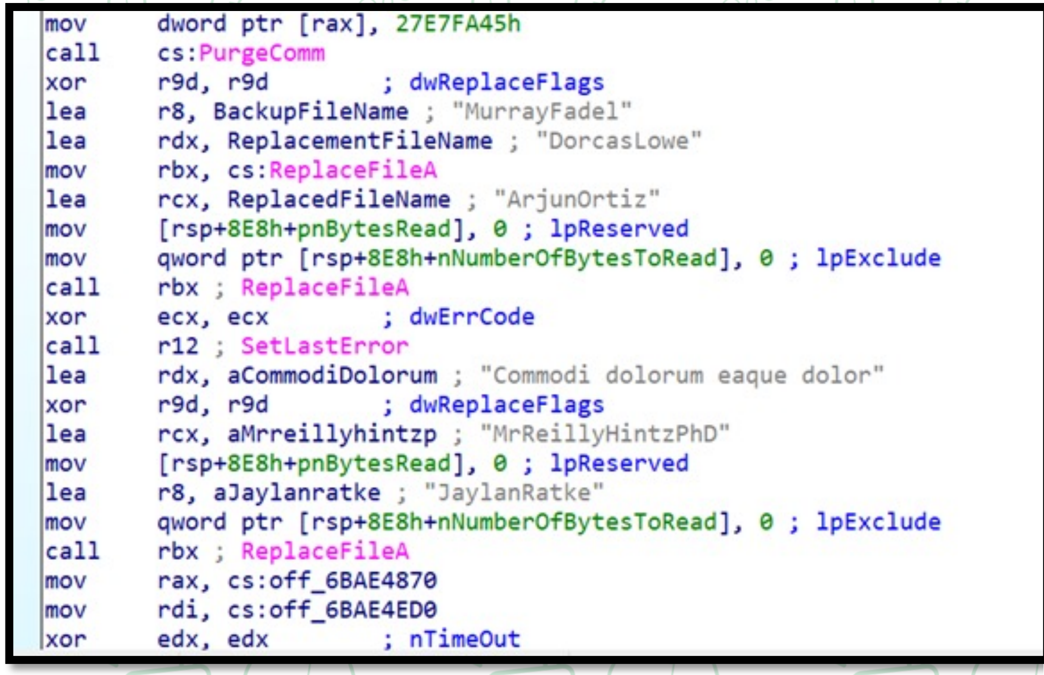


```
call    cs:RegFlushKey
mov     rax, cs:off_6BAE4C20
xor     ecx, ecx            ; hFile
mov     dword ptr [rax], 268ED5C8h
mov     rax, cs:off_6BAE4D10
mov     dword ptr [rax], 0BBF403DAh
mov     rax, cs:off_6BAE4740
mov     dword ptr [rax], 24A7806Fh
call    rbp ; FlushFileBuffers
mov     rax, cs:off_6BAE4680
xor     ecx, ecx            ; hFile
mov     edx, [rax]
mov     rax, cs:off_6BAE4910
mov     [rax], edx
call    rdi ; GetFileType
mov     rax, cs:off_6BAE4C70
mov     ecx, 26E5D69Dh
mov     dword ptr [rax], 0E5BE27CDh
call    sub_6BAD8C10
xor     r9d, r9d            ; hEvent
xor     r8d, r8d            ; dwNotifyFilter
```

Obfuscation and File Attribute Spoofing

ReplaceFile

- Replaces one file with another file, with the option of creating a backup copy of the original file. The replacement file assumes the name of the replaced file and its identity.
- This function combines several steps within a single function. An application can call **ReplaceFile** instead of calling separate functions to save the data to a new file, rename the original file using a temporary name, rename the new file to have the same name as the original file, and delete the original file.
- Another advantage is that **ReplaceFile** not only copies the new file data, but also preserves the following attributes of the original file:
  - Creation time
  - Short file name
  - Object identifier
  - DACLs
  - Security resource attributes
  - Encryption
  - Compression
  - Named streams not already in the replacement file
- For example, if the replacement file is encrypted, but the replaced file is not encrypted, the resulting file is not encrypted.

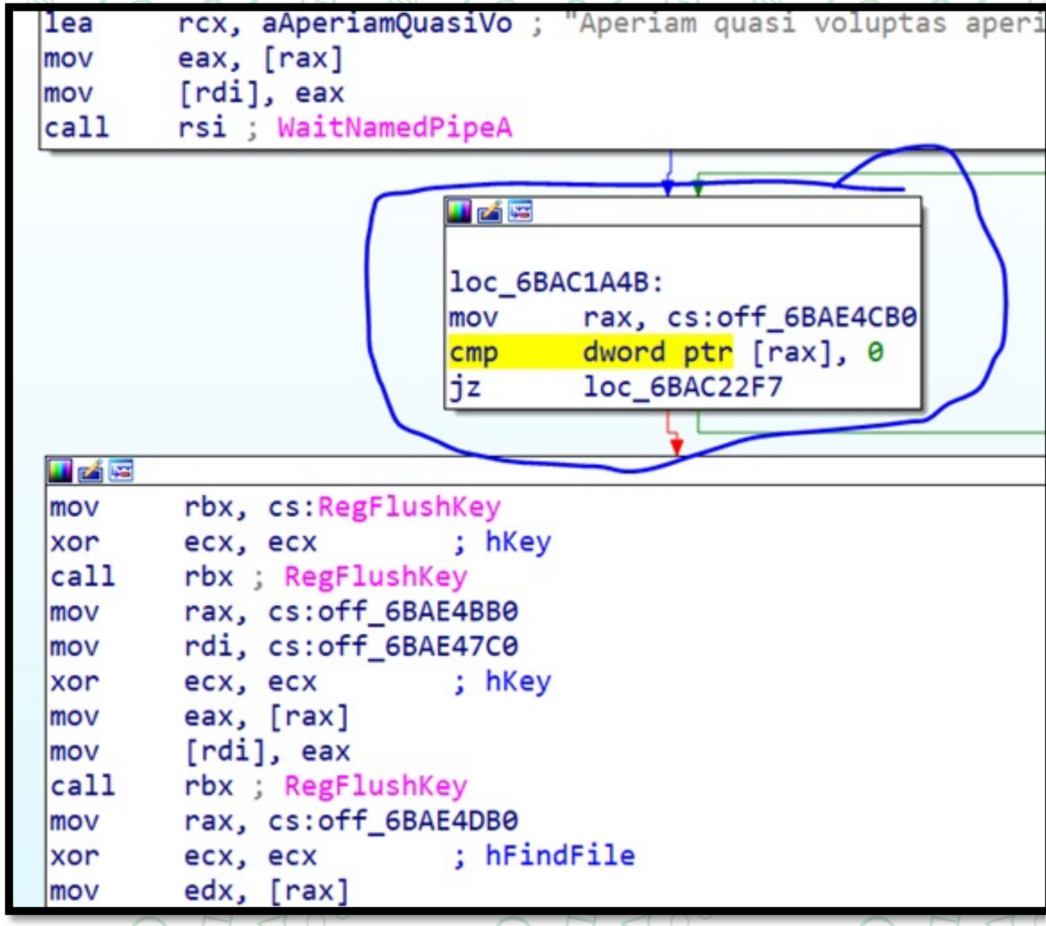


```
mov     dword ptr [rax], 27E7FA45h
call    cs:PurgeComm
xor     r9d, r9d        ; dwReplaceFlags
lea     r8, BackupFileName ; "MurrayFadel"
lea     rdx, ReplacementFileName ; "DorcasLowe"
mov     rbx, cs:ReplaceFileA
lea     rcx, ReplacedFileName ; "ArjunOrtiz"
mov     [rsp+8E8h+pnBytesRead], 0 ; lpReserved
mov     qword ptr [rsp+8E8h+nNumberOfBytesToRead], 0 ; lpExclude
call    rbx ; ReplaceFileA
xor     ecx, ecx        ; dwErrCode
call    r12 ; SetLastError
lea     rdx, aCommodiDolorum ; "Commodi dolorum eaque dolor"
xor     r9d, r9d        ; dwReplaceFlags
lea     rcx, aMrreillyhintzp ; "MrReillyHintzPhD"
mov     [rsp+8E8h+pnBytesRead], 0 ; lpReserved
lea     r8, aJaylanratke ; "JaylanRatke"
mov     qword ptr [rsp+8E8h+nNumberOfBytesToRead], 0 ; lpExclude
call    rbx ; ReplaceFileA
mov     rax, cs:off_6BAE4870
mov     rdi, cs:off_6BAE4ED0
xor     edx, edx        ; nTimeOut
```

Committing changes to the Registry

RegFlushKey

- Calling **RegFlushKey** is an expensive operation that significantly affects system-wide performance as it consumes disk bandwidth and blocks modifications to all keys by all processes in the registry hive that is being flushed until the flush operation completes.
- **RegFlushKey** should only be called explicitly when an application must guarantee that registry changes are persisted to disk immediately after modification.
- All modifications made to keys are visible to other processes without the need to flush them to disk



```
lea     rcx, aAperiamQuasiVo ; "Aperiam quasi voluptas aperi
mov     eax, [rax]
mov     [rdi], eax
call    rsi ; WaitNamedPipeA
```

```
loc_6BAC1A4B:
mov     rax, cs:off_6BAE4CB0
cmp     dword ptr [rax], 0
jz      loc_6BAC22F7
```

```
mov     rbx, cs:RegFlushKey
xor     ecx, ecx        ; hKey
call    rbx ; RegFlushKey
mov     rax, cs:off_6BAE4BB0
mov     rdi, cs:off_6BAE47C0
xor     ecx, ecx        ; hKey
mov     eax, [rax]
mov     [rdi], eax
call    rbx ; RegFlushKey
mov     rax, cs:off_6BAE4DB0
xor     ecx, ecx        ; hFindFile
mov     edx, [rax]
```

Recommendations

Recommendations – DO THIS TODAY

1. Review your biomanufacturing network segmentation
   - Run tests to verify proper segmentation between corporate, guest and operational networks
   - Most facilities use remote logins with shared passwords to operate key instrumentation. Enforcing segmentation is essential.
2. Work with biologists and automation specialists to create a “crown jewels” analysis for your company
   - Ask: “if this machine was inoperable overnight, what would be the impact?”
   - Ask: “how long would it take to re-certify (GxP) this instrument?”
3. Test and perform offline backups of key biological infrastructure
   - Ladder logic for biomanufacturing instrumentation
   - SCADA and Historian configurations
   - Batch record system
4. Inquire about lead times for key bio-infrastructure components
   - Chromatography systems
   - Endotoxin and microbial contamination systems

Recommendations - continued

Prevention is Key

- Use antivirus with behavioral analysis capabilities
- Phishing is a vector of attack
  - Train biomanufacturing facility staff to look out for targeted attacks
  - Review LinkedIn and other social media posts of employees for vaccine manufacturing posts to determine likely targets

Awareness

- The Bioeconomy and Biomanufacturing sectors are under concerted, sophisticated attack. You are a target.
- This malware is extremely difficult to detect due to metamorphic behavior. Vigilance on key personnel corporate computers is important.

Recommendations – continued (2)

Accelerate upgrade paths for key instruments

- Many machines in the sector use outdated operating systems. Segment them off aggressively and accelerate upgrade timelines

Acknowledgements

We would like to thank Alexander Petrovitch for his contributions to this report and help in analysis

bio-isac

Creating Collaborative Threat Intelligence for the Bioeconomy