Analysis Report

SECURITY REPORT

2023

APR

Command&Control
サーバを利用する不正行為分析

CyberFortress

このレポートは、株式会社サイバーフォートレスの独自の調査にて収集されたデータに基づき作成しています。

Analysis Report.

Command & Control サーバを利用する不正行為分析

01. Command & Control? C2?

マルウェア分析レポートを読んでいると、「外部C2サーバまたはC&Cで追加のファイルやスクリプトをダウンロードしてマルウェアをダウンロードする」という文章がよく見られる。しかし、「なぜC2またはC&Cがこれほどまでに使用されるのか」という疑問があり、また、C2という用語がどのような意味を持つのかが曖昧な場合がある。

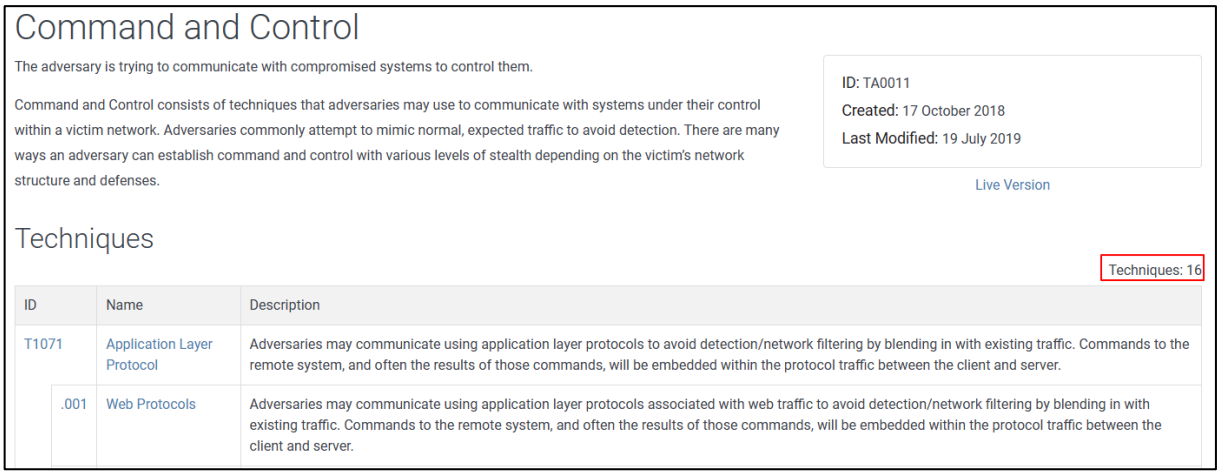
2021年下半期におけるマルウェアの流布先は、前年同期比で38%増加し、製造、健康/医学、教育/塾のサイトが主な経由地であることが分かる。また、IoTマルウェアのMoziはまだ流行しており、Emotetの持続的な検出、情報漏洩マルウェアの持続的な流布、およびMicrosoftサポート診断ツールの脆弱性であるFolina(CVE-2022-30190)を利用したマルウェアも流行している。これらの情報から、マルウェアの流布先および経由地は持続的に増加しており、様々な業種のサイトが経由地として使用されていることが確認された。

マルウェアの流布先は、マルウェアを配布するためのサイトですが、経由地とは、マルウェアまたはスクリプトに含まれる、流布先に移動させるためのURLを指す。我々が知っているCommand＆Controlアーバーとは、この経由地と同じ用途を持つため、C&C、C2、経由地などと呼ばれることがある。しかし、これらの用語には混乱が生じることがある。そこで、今回は「Command＆Controlサーバを利用したマルウェア分析」に焦点を当て、C&Cサーバの定義から、既存の使用/悪用の方法、最近発生した動向までを調べてみる。

02. Command & Controlサーバとは

1) Command & Controlサーバの定義

サーバはコマンド及び制御、C2または、C&Cで呼ばれており、攻撃者が初期侵入に成功した機器との通信を維持するために使用するツール及び技術の集合とされる。一般的には被害者組織の機器と攻撃者が制御するプラットホーム間の一つ以上の隠密な通信チャンネルで構成され、感染した対象に対してコマンド送信、追加悪性ペイロードのダウンロード、奪取したデータを攻撃者に送信することに使用される。MITRE ATT&CKフレームワークv12基準では、16個の上位Command＆Control技術と23個の下位Command＆Control技術が記載されている。今回はCommand＆Controlサーバを略してC2サーバと表記する。



Command and Control

The adversary is trying to communicate with compromised systems to control them.

Command and Control consists of techniques that adversaries may use to communicate with systems under their control within a victim network. Adversaries commonly attempt to mimic normal, expected traffic to avoid detection. There are many ways an adversary can establish command and control with various levels of stealth depending on the victim's network structure and defenses.

ID: TA0011
Created: 17 October 2018
Last Modified: 19 July 2019

Live Version

Techniques

Techniques: 16

ID	Name	Description
T1071	Application Layer Protocol	Adversaries may communicate using application layer protocols to avoid detection/network filtering by blending in with existing traffic. Commands to the remote system, and often the results of those commands, will be embedded within the protocol traffic between the client and server.
.001	Web Protocols	Adversaries may communicate using application layer protocols associated with web traffic to avoid detection/network filtering by blending in with existing traffic. Commands to the remote system, and often the results of those commands, will be embedded within the protocol traffic between the client and server.

【▲ MITRE ATT&CK version 12.1から確認できるCommand and Control技術リスト】

C2サーバを話すときに登場する用語として、Zombie、Botnet、Beaconがある。

① Zombie

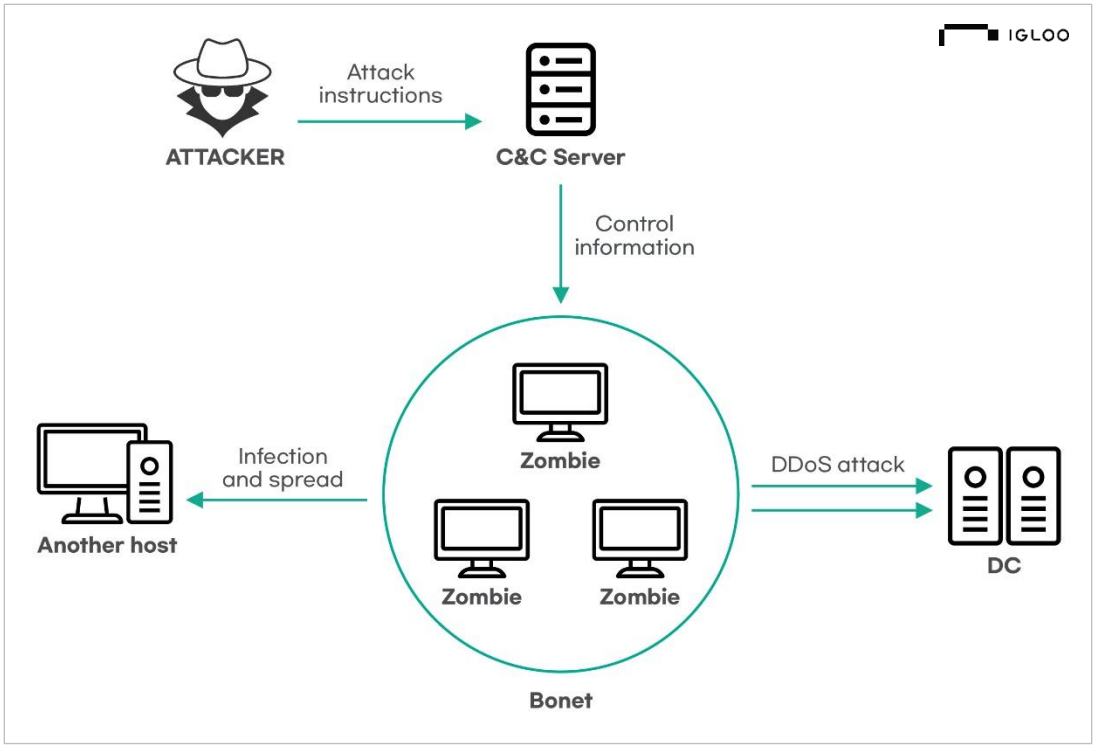
Zombieは、マルウェアに感染して、実際の所有者に知られることなく、もしくは同意を得ることなく、攻撃者がリモートで制御できるコンピューターや、その他の類似の機器を指す。一部のウイルス、トロイの木馬、および望ましくないプログラム(PUA)が機器に感染した後、情報を盗み出したり、特定の操作を行ったりする。しかし、多くのマルウェアは、主に攻撃者のC2サーバにアクセスするためのツールとして使われる。C2サーバにアクセスが成功すると、スパムメールのリレーサーバーから大規模な分散型サービス拒否攻撃(DDoS)を行うなど、多様な操作が実行できる。

② Botnet

Botnetは、不法な目的のためにZombie化された機器の集まりである。Botnetを使って仮想通貨の採掘を行うことから、ウェブサイトをオフラインにするための分散型サービス拒否攻撃(DDoS)まで、様々な攻撃が可能である。Botnetは、一般的に共有C2インフラを中心に統合される。また、攻撃者がAaaS(Attack as a Service)タイプで、他の犯罪者にBotnetのアクセス権限を売ることもある。

③ Beacon

Beaconは、感染した機器が、コマンドや追加のペイロードを受信するために、特定の時間に攻撃者のC2サーバにアクセスしようとするクライアントマルウェアのことを指す。検知を避けるために、Beaconは、わざと任意の間隔で信号を送信する場合や、C2サーバにアクセスするために一定期間待機する場合がある。



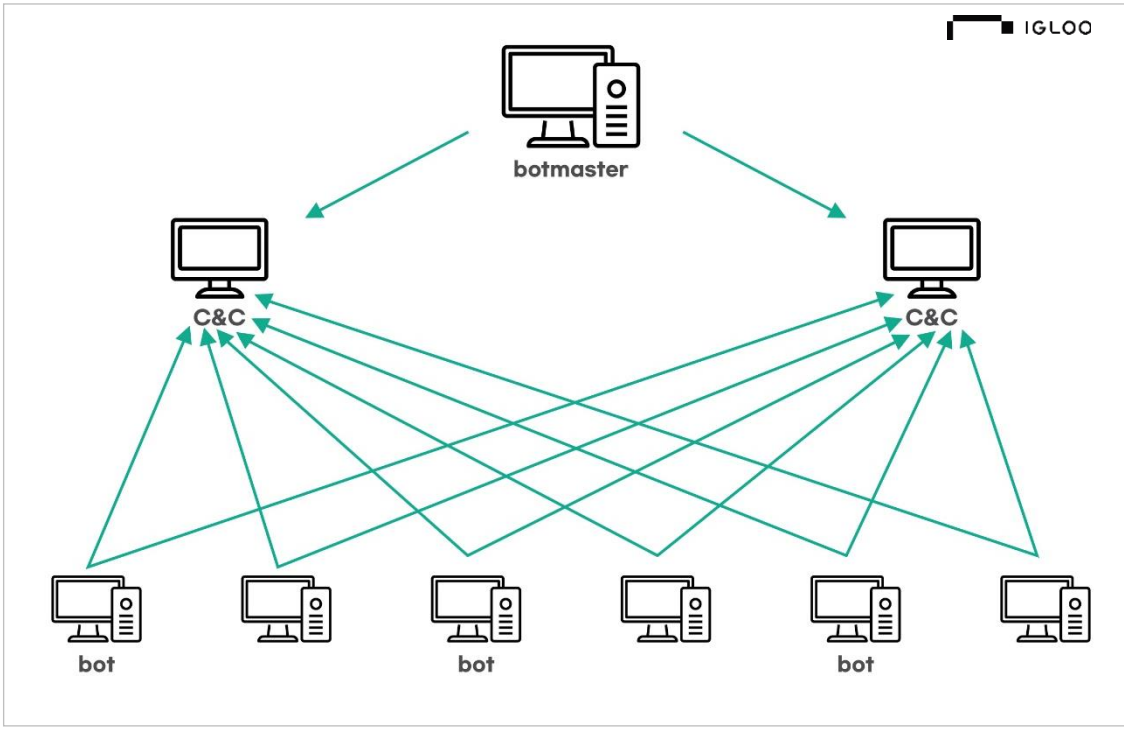
【▲ C2を利用して行われる攻撃の一般的な例
(参考: https://download.huawei.com/mdl/image/download?uuid=6a3009dbd6f64c4c9b0e469431442693)】

2) C2モデル

①中央集中型

中央集中型モデルは、一般的なクライアント↔サーバ関係と非常に似た動作をしている。感染したPCから動作するクライアントマルウェアは、最初に実行されると、C2サーバにアクセスして実行有無を知らせる。通常、感染したPCが属するインフラは、様々なセキュリティ機器やロードバランサーなどの複雑なネットワーク構成を持っている。攻撃者は、合法的なウェブサイトに侵入して、所有者の同意なしにC2サーバをホスティングしたり、Public Cloudサービスやコンテンツ送信ネットワーク(CDN)を利用してC2活動をホスティングまたはマスキングすることがある。

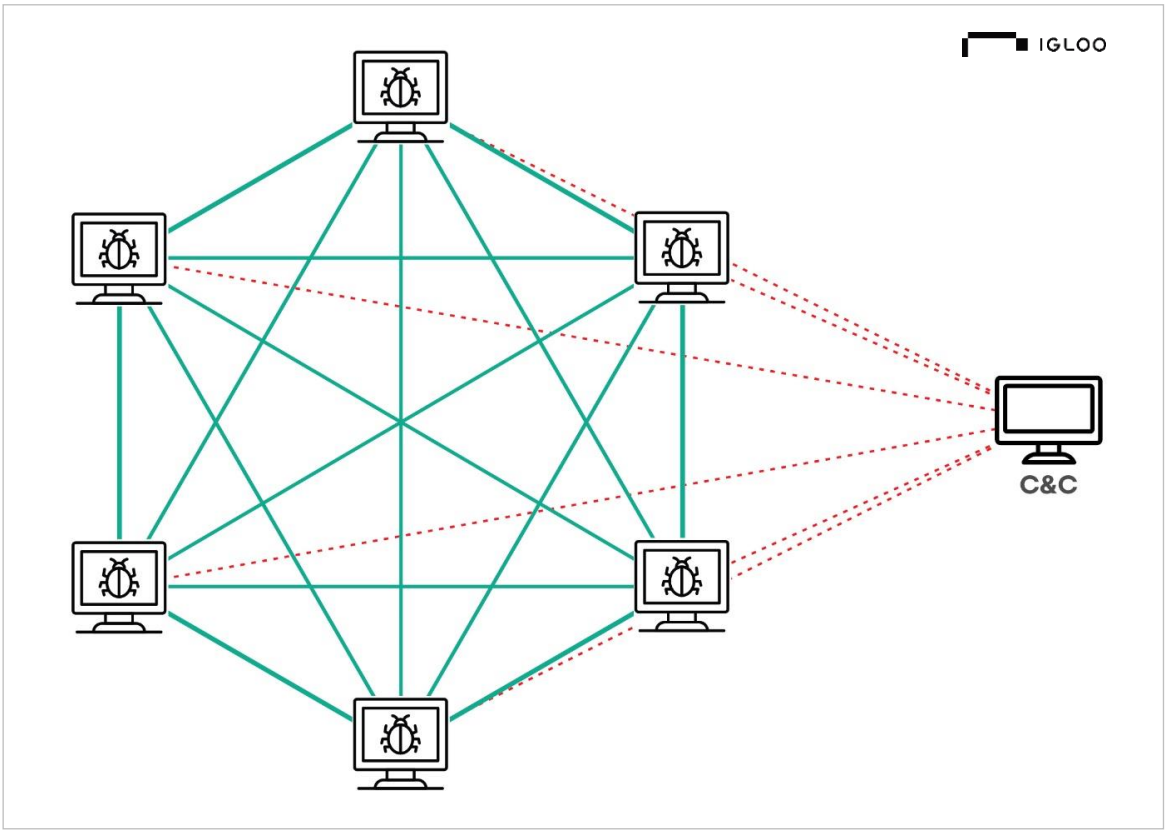
C2で使用されるサーバは、通常、キャンペーンに関連するものであり、早期に発見されると、ドメインやサーバは数時間以内に非活性化されることがある。最新のマルウェアは、複数のC2サーバを使用し、片方に接続できない場合は、他のC2サーバに接続するようにハードコーディングされる場合がある。早期発見を回避するために、C2を難読化する方法を使用することもある。さらに、GPS座標やInstagramのコメントなど、画像に含まれる情報からC2サーバリストを取得するマルウェアも発見されている。



【▲ 中央集中型方法の構成図
(参考：https://www.usenix.org/legacy/event/hotbots07/tech/full_papers/wang/wang_html/figure1.png)】

② P2P

P2Pモデルでは、Botnetの構成員が互いにメッセージを中継する分散された方法で、コマンドや制御コマンドが送信される。一部のBotはまだサーバとして機能することができますが、駐留型またはマスターノードは存在しないため、これにより中央集中型モデルよりも非活性化がより困難になる。ただし、攻撃者が全体のBotnetにコマンドを送信することがより困難になる。また、P2Pネットワークは、基本的にC2チャンネルが中断された場合に代替メカニズムとして使用されることがある。



【▲ P2P方式の構成図
(参考：https://media.kasperskycontenthub.com/wp-content/uploads/sites/103/2020/04/09145041/DDG-P2P-300x223.png)】

③ その他

感染した機器に指示を下すためには、多様な技術が観察されており、一部の攻撃者は、一般的にSNSプラットフォームが遮断されないことを悪用して、C2サーバとして広く利用されるTwitterのDirect Messageを利用するなど、様々な手法を用いている。Twittorというプロジェクトは、完全なC2サーバとして悪用されることを目的として、TwitterのDirect Messageのみを使用している。また、攻撃者は、Gmail、IRCチャットルーム、Pinterestなどを利用して、感染した機器に指示を下すためのツールとして活用されることが確認されている。

03. 攻撃者がC2を作る方法

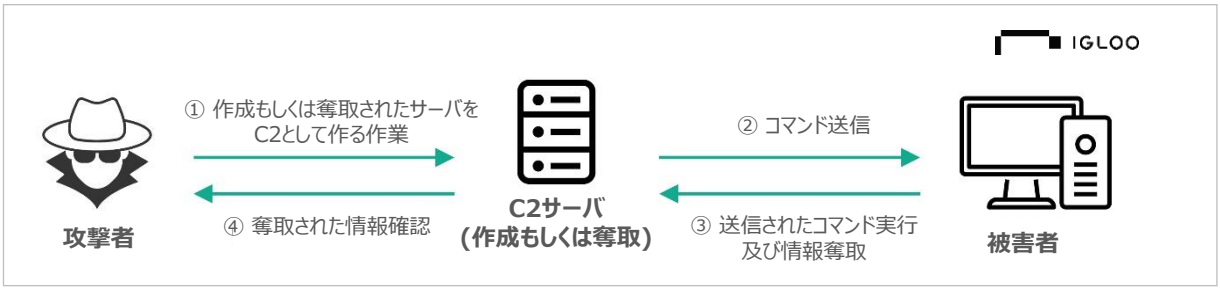
今までの調査により、C2の定義、モデル、使用している理由について明らかになる。攻撃者がC2を利用することで、マルウェアがインストールされた機器を利用した側面移動、情報奪取、Zombie化によるBotnet登録など多数の攻撃に悪用できることが確認された。

しかし、攻撃者がC2を作るためには、サーバやコマンドを送信する対象が必要である。多様なC2モデルを使用しても、攻撃者は少なくとも1つ以上の直接使えるC2を確保する必要がある。また、C2の存在を最大限に気づかれないようにする必要がある。このため、攻撃者はC2を直接作成する方法から、正常なサービスを悪用する方法まで、様々な手法を使っている。

1) 攻撃者がC2生成または、正常サーバ奪取

攻撃者がC2を直接作ることは一番簡単な方法である。物理または仮想的なPCを使用して、C2として機能するサーバを直接作成し、コマンドや制御に必要な情報を事前に保存することができる。攻撃者が直接作成したため、管理は容易であるが、大規模な攻撃キャンペーンの場合は、正体が発覚する可能性がある。

この問題を解決するための簡単な方法として、正常にサービスしているサーバを奪取する方法がある。一般的に、攻撃者は脆弱性のある正常にサービスを提供しているサーバを狙い、そのサーバをC2として使用する。使用中にC2を追跡するセキュリティ専門家などがサーバの非活性化を必要とする場合、攻撃者は簡単にサーバ内に隠されたファイルを削除することができる。一般的にアクセスする場合は正常なサイトに見えるため、そのサーバがC2として使用されていることに気づくことは難しい。

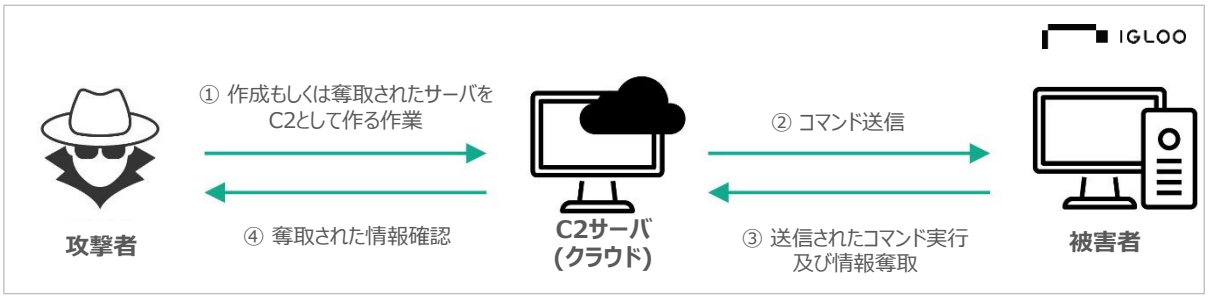


【▲ 攻撃者がC2生成または、正常サーバを奪取する方法の攻撃構成図】

2) 正常サービスにC2生成(クラウドコンピューティング)

攻撃者がC2を生成したり、正常なサーバを奪取して使用する方法に加えて、外部からサービス中のクラウドコンピューティングサーバにC2サーバを作成して使用する方法も存在する。例えば、2017年にはPOS(Point-of-Sales)マルウェアであるAlinaPOSやJackPOSの変種が、AWSから無料で提供されるEC2 t2.microインスタンスを悪用して数千個のC2を作成・使用した事例があった。

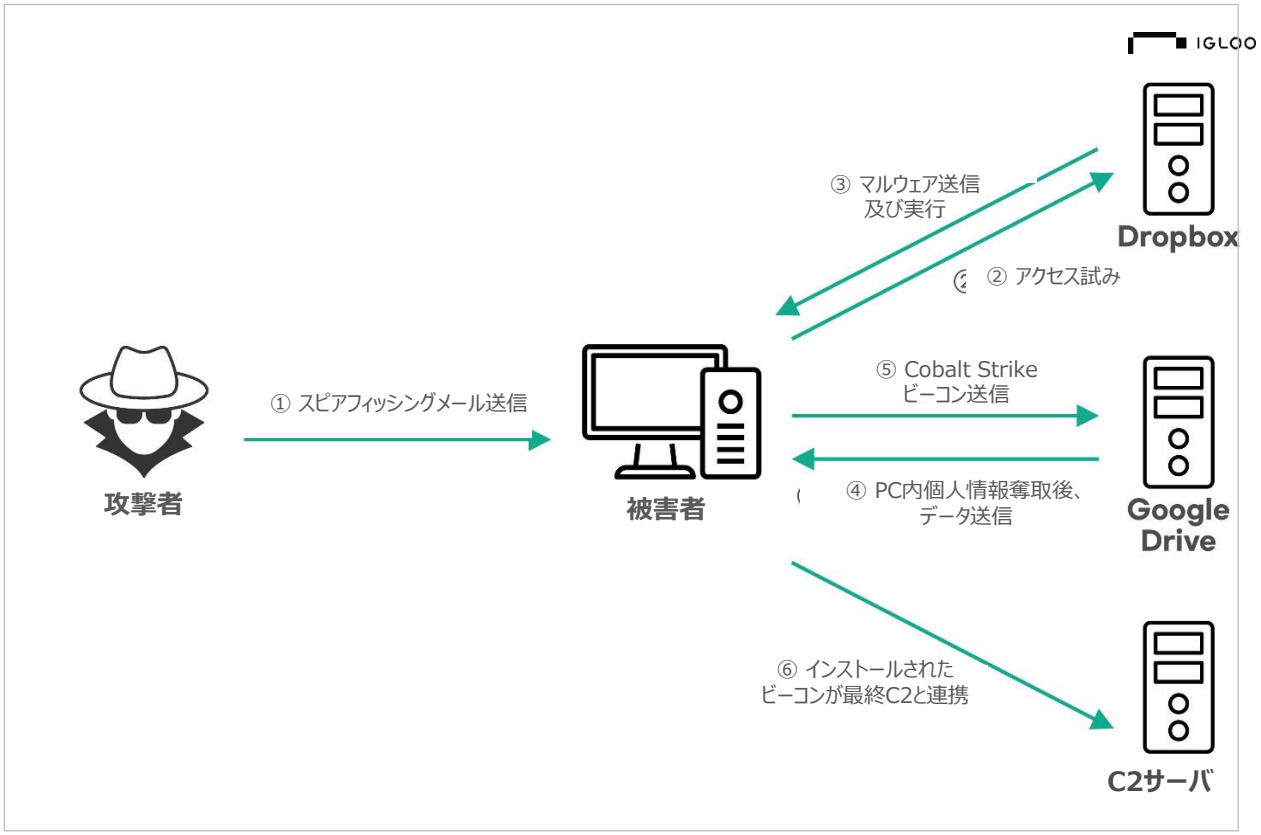
クラウドコンピューティングサービスでは、通常、アカウント作成やお支払い情報の入力によって無料のインスタンスが提供されますが、攻撃者はこのプロモーションを悪用して奪取されたメールアカウントやお支払い情報を利用してC2サーバを作成した。ただし、一定以上のトラフィックが発生すると使用料を支払わなければならず、報告などの理由でサービス管理者によってインスタンスが非活性化される可能性があるため、C2サーバを持続的に運用することは困難である。



【▲ クラウドコンピューティングを利用して正常サービスにC2を作成する方法の攻撃構成図】

3) クラウド保存先をC2として利用

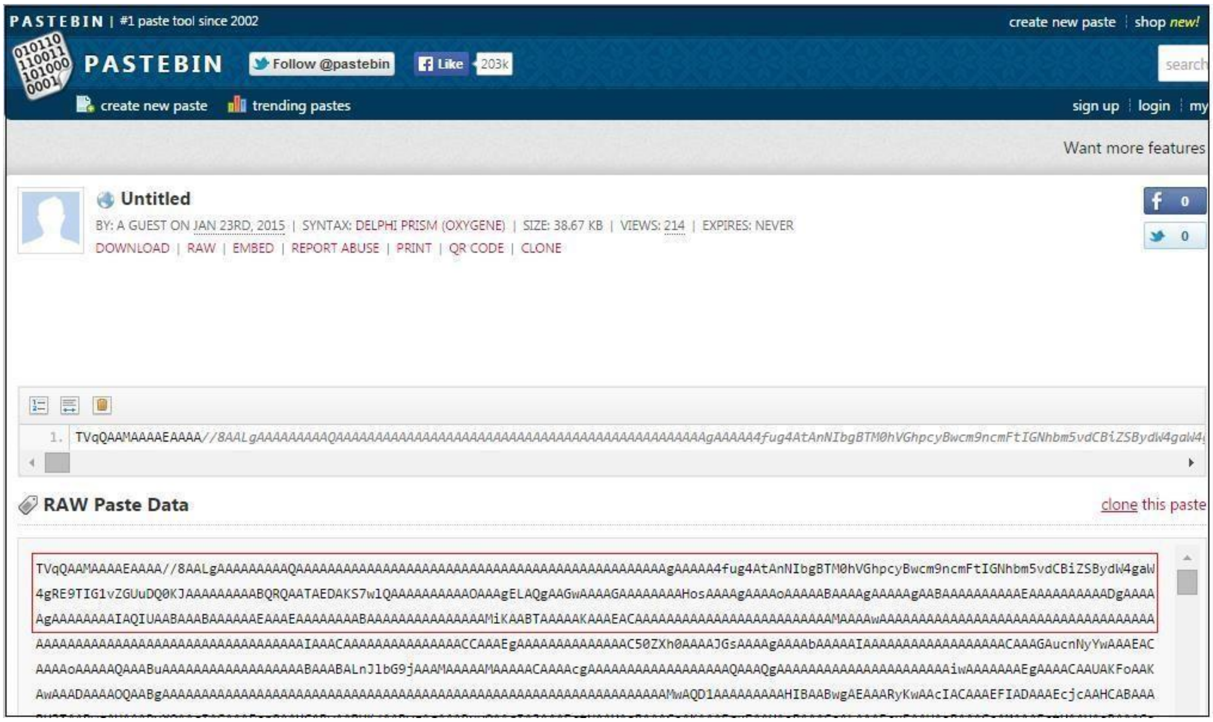
ロシア推定の攻撃グループであるAPT29(Cozy Bear, Nobelium)の場合、2022年5月~6月まで海外大使館を対象に攻撃するキャンペーンを展開する際、クラウド保存先であるDropbox, Google Driveを活用して攻撃に悪用した。クラウド保存先の使用メリットである手軽な会員登録手続きを悪用して奪取されたメールアカウント/お支払い情報だけでアカウントの作成ができ、データ保存に大きな容量は要らないことを利用している。しかし、クラウドサーバを使用することと同じく内部モニタリング、報告などの利用でファイルダウンロードができなかったりアカウントが中止される可能性がある。



【▲ クラウド保存先を利用してC2を作成する方法の攻撃構成図】

4) テキスト共有サービスをC2として利用

攻撃者が作成したC2サーバが悪性に検知されたり遮断されることが増えるため、攻撃者はファイルを送信する方法からマルウェアのダウンロードパスや悪性スクリプトをテキストで送信する方法に注目している。攻撃者がよく使う方法としては、テキスト共有サイトに攻撃者が望むテキストを保存した後、最初にインストールされるDownloaderやDropperがアップロードされたテキストを読み込んで2次C2にアクセスし、スクリプトコードをダウンロードしたり、非常に小さい単位の実行ファイルを直接ダウンロードするなどの方法を悪用している。

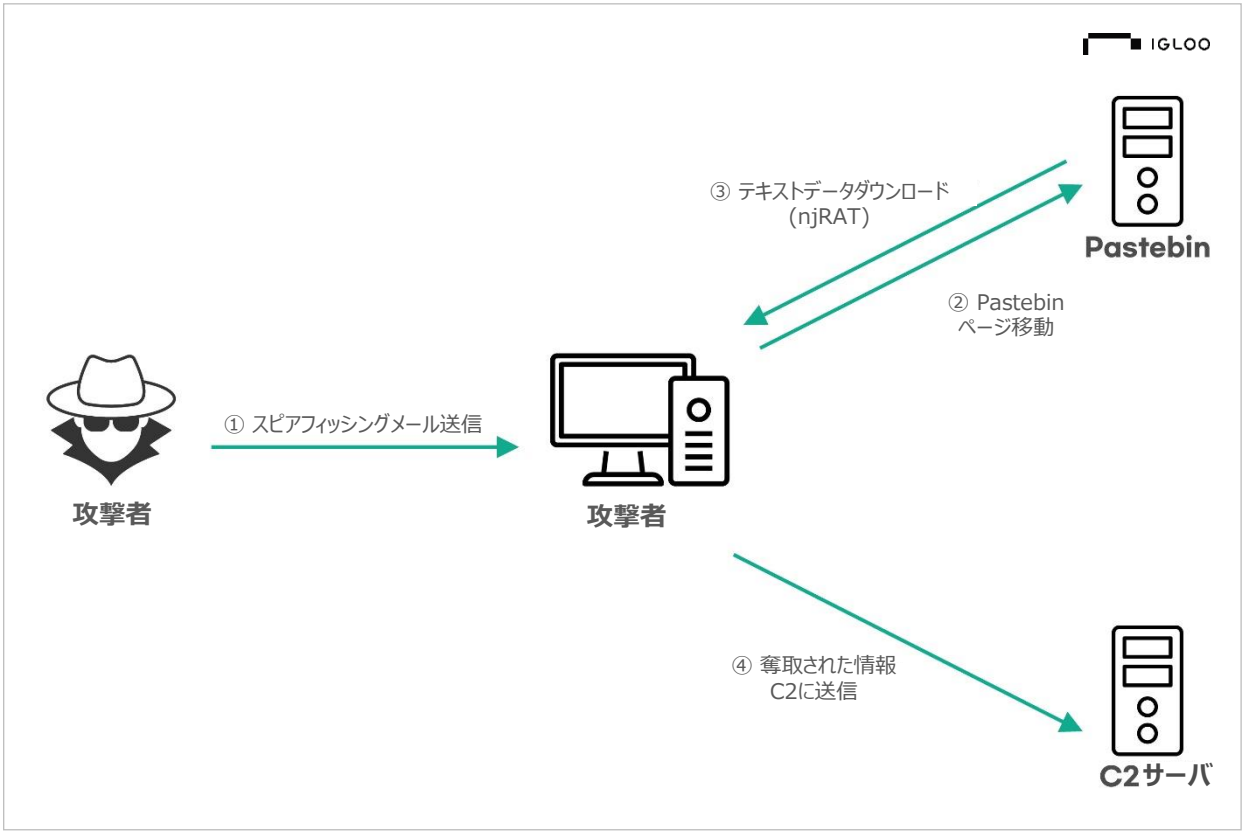


【▲ テキスト共有サービスであるPastebinにアップロードされたマルウェアのデータ】

確認するサンプルは、活発に活動しているnjRATの変種で、テキスト共有サービスであるPastebinを1次C2として利用している。攻撃に使用されるテキストをPastebinにアップロードし、当該のサンプルは実行ファイルデータをbase64でエンコードして保存する。最初にアクセスする際に、当該のテキストを読み込んでnjRATを実行することが確認された。

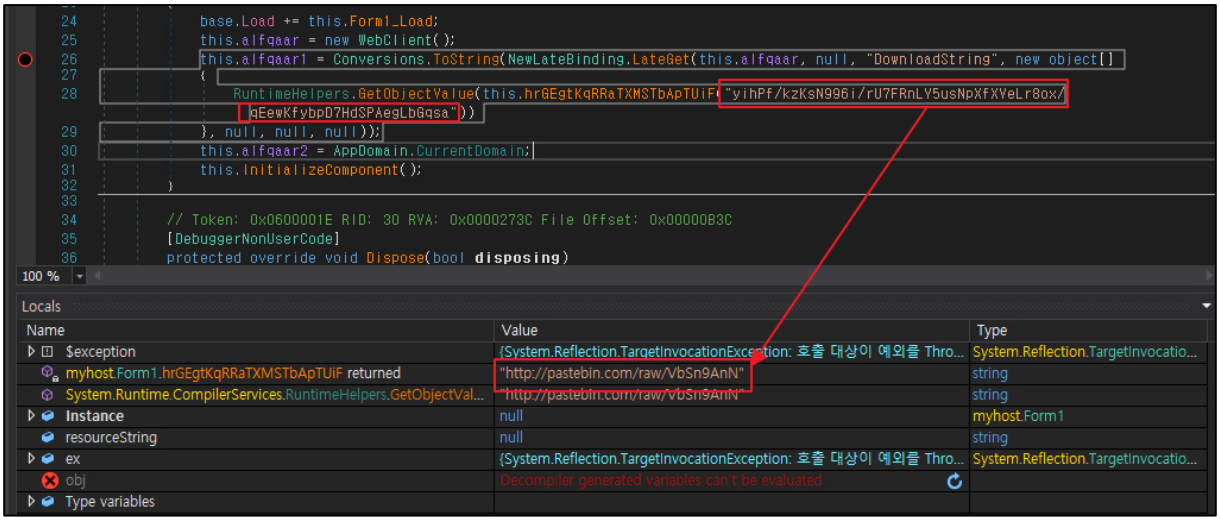
区分	DATA
ファイル名	Myhost.exe
HASH	9a8b2bed82fe51713400e3b8db8af5ca

【▲ njRATサンプル情報】



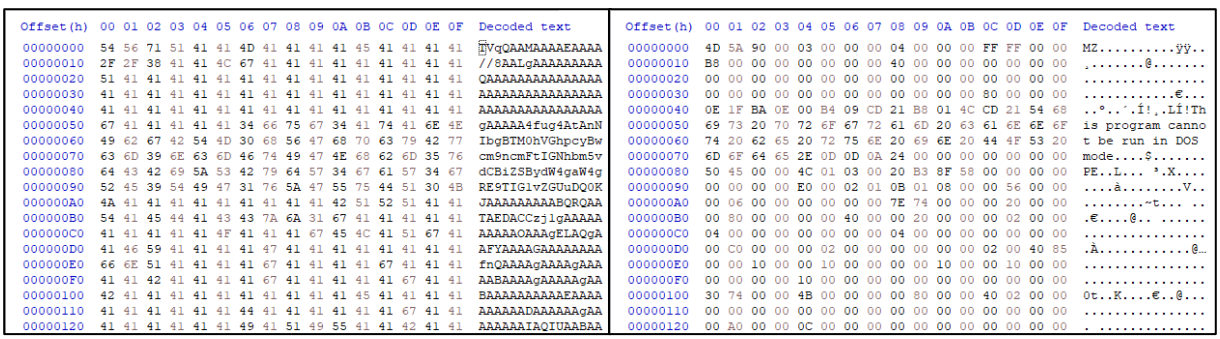
【▲ テキスト共有サービスをC2として悪用する方法の攻撃構成図】

最初に実行時アクセスするPastebinアドレスを暗号化していて復号化以降にPastebinにアクセスしてテキストデータを要請する。



【▲ 実行ファイル内部に存在する暗号化されたC2アドレス】

現在、ページは存在しないが送信されるデータはbase64でエンコードされたデータを受信されることが確認できてデコードしてみるとPE構造の実行ファイルだった。



【▲ Pastebinにアップロードされたbase64エンコードデータをデコードした結果】

最終に実行されるファイルはnjRATに確認できて、キーロギングを含めて情報奪取を目標とするインフォスティーラーであった。2次C2はhxxp://ddrrtt[.]ddns[.]net:1177に情報を送信することを確認した。

```
public kl()
{
    this.lastKey = Keys.None;
    this.Logs = "";
    this.vn = "[kl]";
}

// Token: 0x06000024 RID: 36
[DllImport("user32.dll")]
private static extern int ToUnicodeEx(uint a, uint b, byte[] c, [MarshalAs(UnmanagedType.LPWStr)] [Out] StringBuilder d, int e, uint f, IntPtr g);

// Token: 0x06000025 RID: 37
[DllImport("user32.dll")]
private static extern bool GetKeyboardState(byte[] a);

// Token: 0x06000026 RID: 38
[DllImport("user32.dll")]
private static extern uint MapVirtualKey(uint a, uint b);

// Token: 0x06000027 RID: 39
[DllImport("user32.dll", CharSet = CharSet.Ansi, ExactSpelling = true, SetLastError = true)]
private static extern int GetWindowThreadProcessId(IntPtr a, ref int b);

// Token: 0x06000028 RID: 40
[DllImport("user32", CharSet = CharSet.Ansi, ExactSpelling = true, SetLastError = true)]
private static extern int GetKeyboardLayout(int a);

// Token: 0x06000029 RID: 41
[DllImport("user32", CharSet = CharSet.Ansi, ExactSpelling = true, SetLastError = true)]
private static extern short GetAsyncKeyState(int a);
```

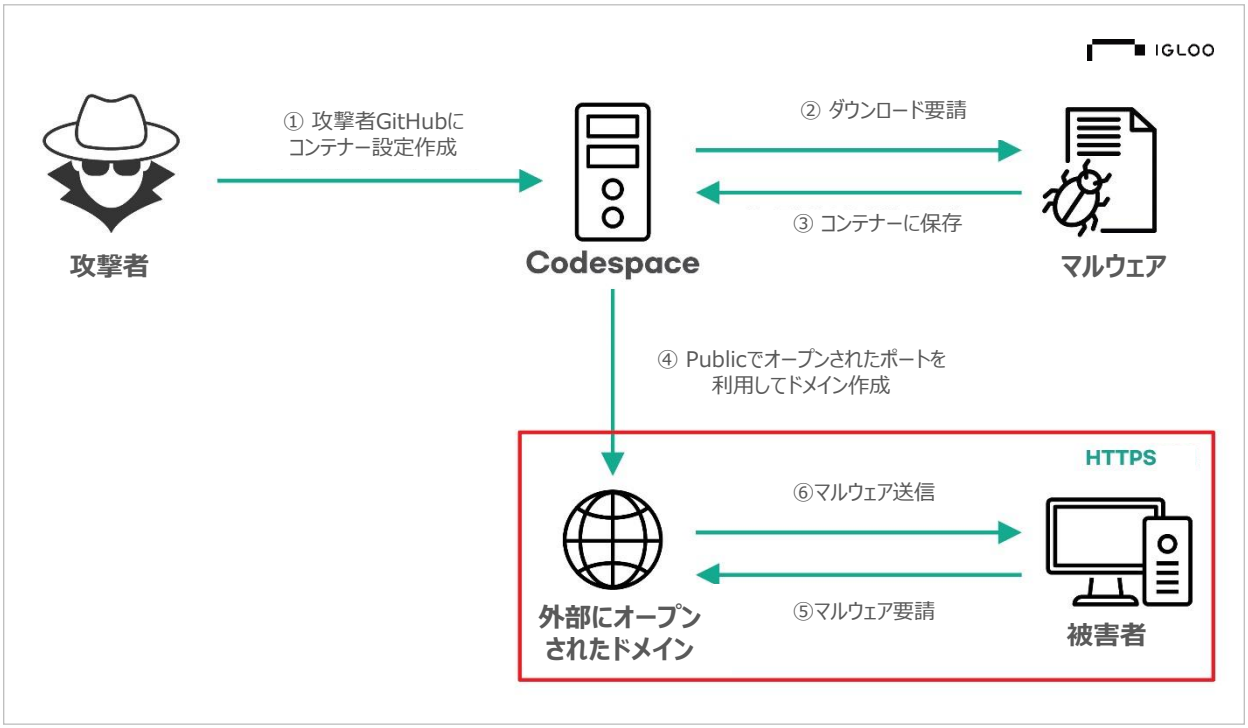
【▲ njRATから使用されるキーロギング関連のAPIリスト】

5) GitHub Codespacesを悪用してC2として利用

GitHub Codespacesは、クラウドベースの開発環境であり、開発者がウェブブラウザから直接コードを作成、編集、実行できるようになっている。JavaScript、Python、Rubyプロジェクトに必要な全てのツールや依存関係が含まれるコンテナー基盤の仮想マシン(VM)を作成する。これにより、開発者は、ローカルで環境設定に時間がかかるIDEを設定せずに、ウェブからリアルタイムで作業することができる。

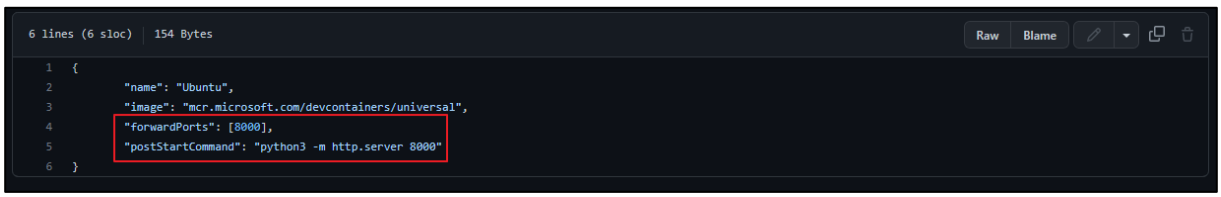
GitHub Codespacesの機能の中で、特定のポートを外部にオープンして、Private/Publicに共有することができる。Privateの場合、組織メンバーのみがアクセス可能であり、Cookieを使用して認証する。Publicに設定する場合、一般ユーザーも認証なしで公開されたポートにアクセスできる。この方法が悪用されると、攻撃者はスクリプトやマルウェアを攻撃に使用するために簡単に流布させたり、C2として使用することができる。特に、GitHubの場合、他のクラウドサービス(AWS、GCPなど)が要求するクレジットカード番号のようなお支払い方法の設定が不要であるため、攻撃者がより簡単に悪用できる。

2023年1月にTrend Microが発見したGitHub Codespacesの脆弱性を調べ、PoCを使用して実際に悪用する方法について調査しよう。

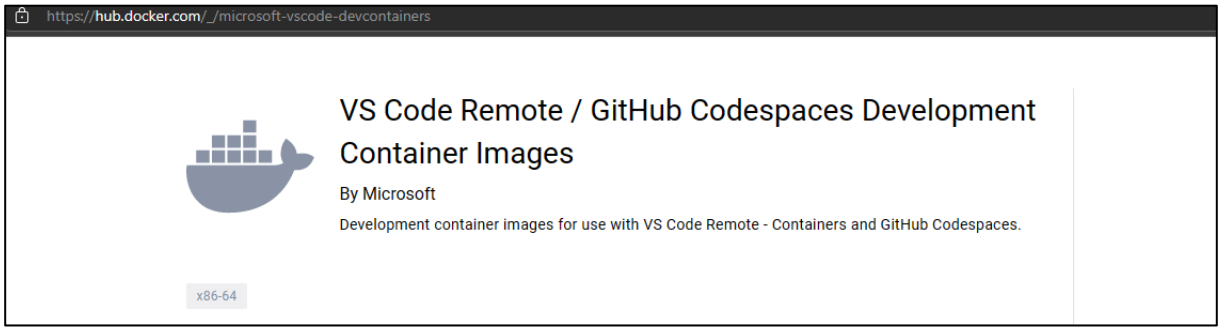


【▲ GitHub CodespacesをC2として悪用する方法の攻撃構成図】

PoCテストに使用するコンテナーイメージは、Microsoftが作成したdevcontainers Dockerイメージで、VS Code Remoteを利用してGitHub Codespacesを使用するイメージである。攻撃者は、devcontainersイメージを使用してコンテナーを作成し、forwardPortsプロパティを使用して8000番ポートをフォワーディングに設定し、また、postStartCommandプロパティを使用して、コンテナーの起動に成功するたびにPythonベースのHTTPサーバを実行するコードをGitHubリポジトリの中に「.devcontainer/devcontainer.json」として保存する。



【▲ コンテナー作成のためのdevcontainers.jsonファイルの内容】



【▲ Docker Hubから確認できるMicrosoftから作成したdevcontainer】

攻撃者は、GitHub CLIにアクセストークンを使用して、codespaceを配布するためにアクセスする。以前に作成したdevcontainer.jsonを使用してcodespaceインスタンスを作成し、ここで設定された8000番ポートをPublicにオープンしてフォワーディングし、HTTPサーバをオープンして外部からもアクセスできるようにする。また、攻撃者はcodespace内部に不正ファイルを保存し、作成されたリンクはPoC基準で100秒間保持され、最大30日間保持される。

```
CODESPACE=$(gh codespace create -R adititli/adititli -m basicLinux32gb)
TORUN=$(cat do-stuff.sh)
echo "[+] Codespace Name: $CODESPACE"
echo "$TORUN" | gh codespace ssh -c $CODESPACE
echo "[+] Updating port visibility to public..."
gh codespace ports visibility $2:public -c $CODESPACE
if [ $? -eq 0 ]; then echo "[+] Here's your opendir -- https://$CODESPACE-$2.preview.app.github.dev/"; fi
echo "[+] Sleeping for 100 seconds..." && sleep 100
echo "[+] Deleting all codespaces..." && gh codespace delete --all
```

【▲ 脆弱性PoCコードの一部】

04. 最後に

今まで、マルウェアがC2サーバを利用して攻撃者が望む方向に使用する方法について調べてみた。攻撃者がC2サーバを作成/奪取するよりも、正常なサービスを悪用して検知を迂回する方法を使用していることから、C2サーバの使用は情報奪取を目的にしている場合には必須だと考えられる。また、情報奪取以外にも、マルウェアの流布の流れで中間経由地の役割及びサーバを使い捨てる戦略を使って検知をより難しくしているようである。

一般的に、インバウンド通信に気を付けるときは多いですが、アウトバウンド通信に対する監視を怠ることがあるようである。普通、マルウェアに感染すると、C2が内部にアクセスするインバウンド通信ではなく、感染したPCが外部のC2に通信するリバースコネクション（アウトバウンド通信）が多いため、アウトバウンドに対する監視/対応をもっと強化する必要があると考えられる。

アウトバウンドDNS要求を、組織内部で使用するDNSサーバに制限して、DNSトンネリング攻撃を防ぐことができたり、プロキシを使用してアウトバウンド通信をチェックする方法もある。プロキシを使用してチェックする場合は、C2と暗号化通信を行う可能性があるため、SSL/TLSトラフィック追加チェック設定に対する協議が必要であり、DNSフィルタリングサービスを利用して、怪しいもしくは新たに登録されたドメインのC2要求を遮断するなどの方法で対応ができると思われる。

05. 参考資料

1) https://www.varonis.com/blog/what-is-c2
2) https://themerkle.com/thousands-of-amazon-aws-instances-host-cc-servers-for-pos-malware/
3) https://www.bleepingcomputer.com/news/security/russian-svr-hackers-use-google-drive-dropbox-to-evade-detection/
4) https://unit42.paloaltonetworks.com/njrat-pastebin-command-and-control/
5) https://krcert.or.kr/kr/bbs/view.do?searchCnd=1&bbsId=B0000127&searchWrd=%EC%9D%80%EB%8B%89&menuNo=205021&pageIndex=1&categoryCode=&nttId=66905
6) https://www.trendmicro.com/en_us/research/23/a/abusing-github-codespaces-for-malware-delivery.html