

断続的に活動するMooBotまたはFlodrix - NICTER Blog

 blog.nictcr.jp/2025/11/moobot_2025_3g

CSRI解析チーム

November 28, 2025

2019年に現れた Mirai 系 マルウェアである MooBot は、**The cow says moo**という使われない文字列や、二重起動防止用のファイル名 **.moopid** といった文字列を持つという特徴がありましたが、2021年には、その代わりにランダムな文字列を作成するためのシードとして **w5q6he3dbrsgmclkiu4to18npavj702f** という文字列を使用するという特徴を持つようになり、NICTにおいても、たびたびこの文字列を含むマルウェアの活動を観測しています。

ボットネットの相関

直近では、2025年9月10日からマルウェアの更新が再開され、DoS 攻撃活動を確認しました。過去にも何度か MooBot を感染させるエクスプロイトを検知していますが、2024年の3月と10月には RapperBot と共通のネットワークリソースが利用されたことが分かっています。

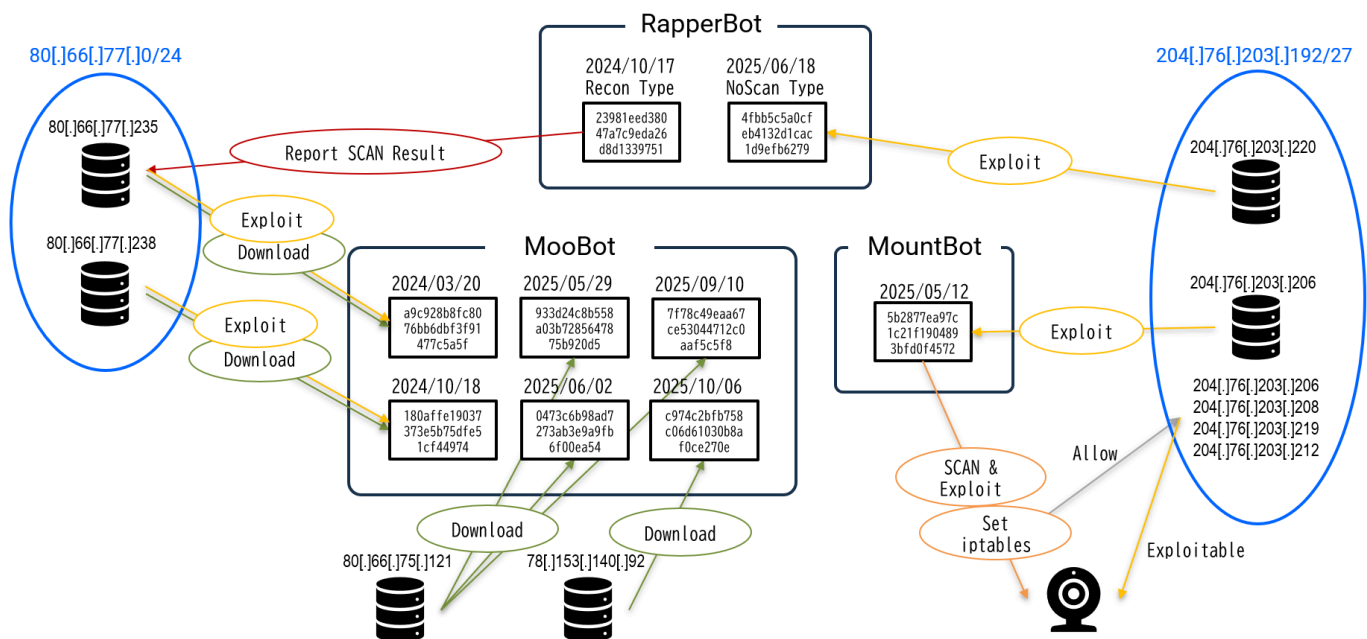


図1. ボットネット相関図 (ハッシュ値はMD5)

また、2025年6月2日のバージョンに関しては、NICTで観測している DVR の脆弱性を狙ったエクスプロイトとは別の経路 (CVE-2025-3248) による、別のターゲット機器 (Langflowサーバ) への感染が報告されていて、Flodrix ボットネットと呼ばれています。

初期化しないメモリデータによる難読化

2025年5月の検体には、Command & Control (C2)サーバへの接続に、複雑な仕組みが使われていました。その接続フローは以下のようなものです。

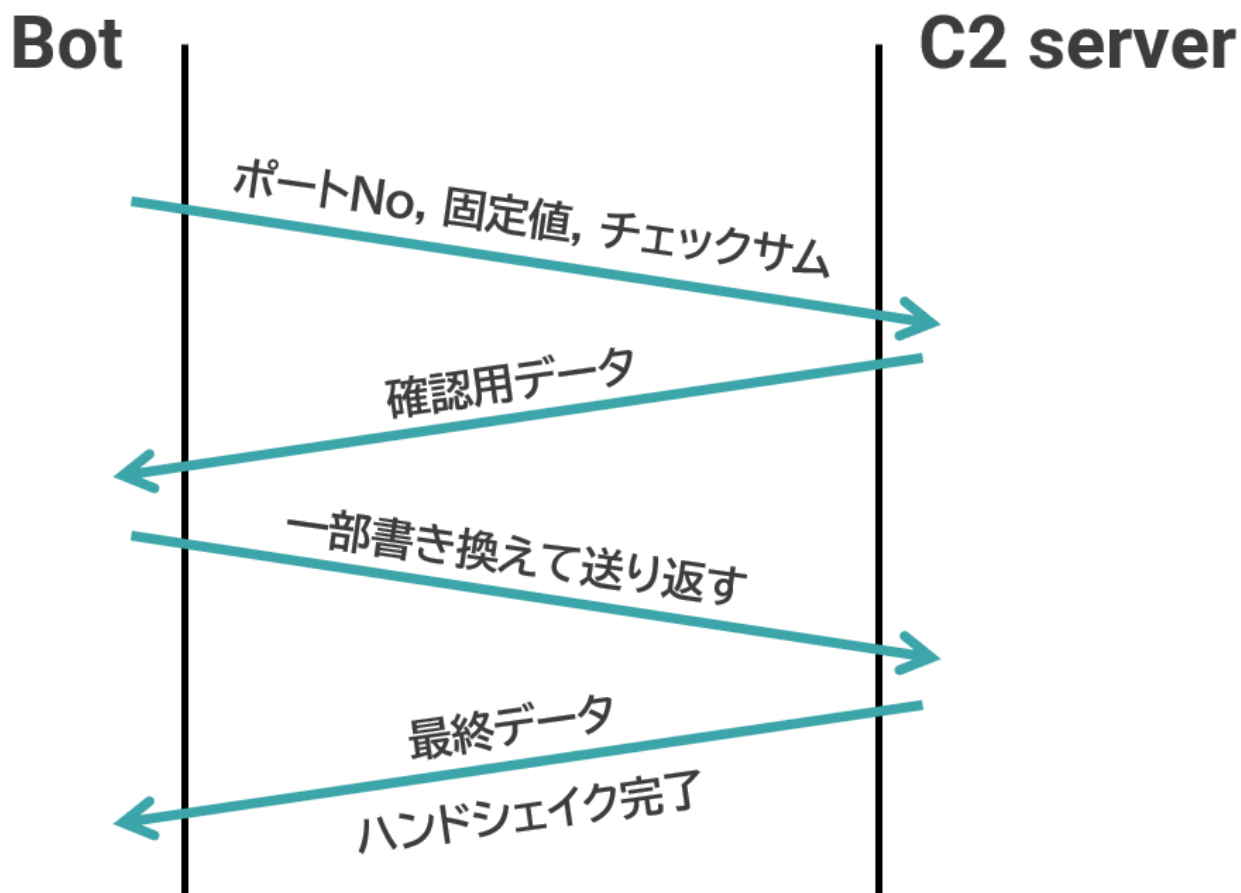


図2. C2 接続フロー

特殊な手法が使われているのが最初のチェックサムで、送信データはほとんどが固定データですが、この C2 接続関数内では、送信データを格納するメモリ領域の一部が初期化されずに使用されています（下図赤枠）。通常このようなコードは該当のメモリ領域の値が不定になり、いわゆる初期化もれによるバグとなりますが、この検体の場合、攻撃者が制御可能な値になります。

```

local_84._0_2_ =
    (ushort)((sStack_4c.sin_port & 0xff) << 8) | sStack_4c.sin_port >> 8;
local_84[2] = 0x20;
local_84[3] = 0x3a;
local_84[0xe] = 1;
local_84[0xf] = 0;
local_84[0x10] = 0;
local_84[0x11] = 0;
local_84[0x18] = 0;
local_84[0x19] = 0;
local_84[4] = 0x42;
local_84[5] = 0xb0;
local_84[6] = 0;
local_84[7] = 0;
local_84._8_4_ = uVar5;
local_84._20_4_ = uVar5;
local_84._28_4_ = uVar5;

```

Source port

Checksum

図3. 初回送信データ

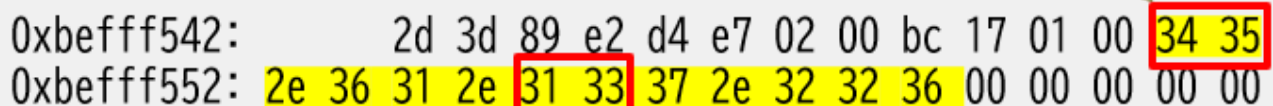
該当するメモリ領域は、上記の C2 接続関数より前に、下図の C2 サーバの IP アドレスを取得する関数で先に使用されます。ここで、3つ目の IP アドレスをこのメモリ領域に格納したときに、第1オクテットおよび第3オクテットの各先頭2文字が、赤枠の初期化されないメモリアドレスにちょうどあてはまるようになっています。

```

void getC2ip(void)
{
    char acStack_78 [32];
    char acStack_58 [32];
    char acStack_38 [32];

    unit_fillbytes(acStack_38, '\0', 0x20);
    unit_fillbytes(acStack_58, '\0', 0x20);
    unit_fillbytes(acStack_78, '\0', 0x20);
    unit_strncpy(acStack_38, "P^P\\EGPSVG(##/", 0xe);
    string_decode(acStack_38, 0xd);
    c2ipset[0].ip = (ip_addr_t)strip2ip(acStack_38);
    c2ipnum = c2ipnum + 1;
    c2ipset[0].port = 0xb3d5;
    unit_strncpy(acStack_58, "SV^\\C@HLTF( #", 0xd);
    string_decode(acStack_58, 0xc);
    c2ipset[1].ip = (ip_addr_t)strip2ip(acStack_58);
    c2ipnum = c2ipnum + 1;
    c2ipset[1].port = 0xb3d5;
    unit_strncpy(acStack_78, "V[HDCXOQUP4 $", 0xd);
    string_decode(acStack_78, 0xc);
    c2ipset[2].ip = (ip_addr_t)strip2ip(acStack_78);
    c2ipnum = c2ipnum + 1;
    c2ipset[2].port = 0xb3d5;
    return;
}

```



0xbefff542:	2d	3d	89	e2	d4	e7	02	00	bc	17	01	00	34	35
0xbefff552:	2e	36	31	2e	31	33	37	2e	32	32	36	00	00	00

45[.]61[.]137[.]226

図4. 初期化もれ領域が設定されている場所

このメモリ領域は各関数の内部でローカル変数として使用するスタック領域であるため、本来は赤枠の部分も初期化する必要がありますが、初期化しないことで、別の関数で設定されたデータをそのまま利用しています。

4バイトのチェックサムは、このデコンパイラが考慮できない過去のスタック領域のデータを含めて計算しなければいけないため、コードをみただけではチェックサムを正しく計算できません。

この複雑な仕組みは、その後すぐに無効化され、使われなくなりました。ただし、複雑なコードは残したまま、ハンドシェイクの状態遷移を一部スキップして接続完了させる形で更新されています。

RapperBotから感染機器を奪ったことをレポート

このボットは競合マルウェアを対象とした killer 関数を持ち、例えば、一般的にマルウェアが広く実施する、自身の実行ファイルを削除するという特徴を利用して、`/proc/{pid}/exe`に `(deleted)` が含まれる場合、そのプロセスを強制停止させるようになっています。

いくつもある killer 処理の中でも最も注目される機能は、5月から6月の検体に含まれていた、特定の IP アドレスと TCP 接続を確立しているプロセスを検出した場合、そのプロセスを強制停止した後に、専用の ID 番号 5 でレポートサーバに機器情報を送信するというものです。

45.89.63.25 194.226.121.51 192.145.28.71 188.92.28.62 138.124.16.126
--

図5. ID=5でレポートする接続IPアドレス5件

このIPアドレスは、すべて5月当時まだ活動していた RapperBot の C2サーバであり、レポート ID 番号 5 は明らかに RapperBot から感染機器を奪ったことを意味しています。しかし、一部ネットワークリソースを共有しているケースがあったこともあり、この MooBot を運用していたグループが Ethan Foltz と敵対していたのか、あるいは同一グループ内で botnet を切り替えたことを検出するための機能だったのか、動機は不明です。

TeamSpeak3攻撃ベクタ

C2の監視をした結果（2025年9月10日～10月6日）、このボットネットの主な攻撃先はロシアであり、特に TeamSpeak3 というボイスチャットプロトコル専用の攻撃ベクタが多いのが特徴的です。

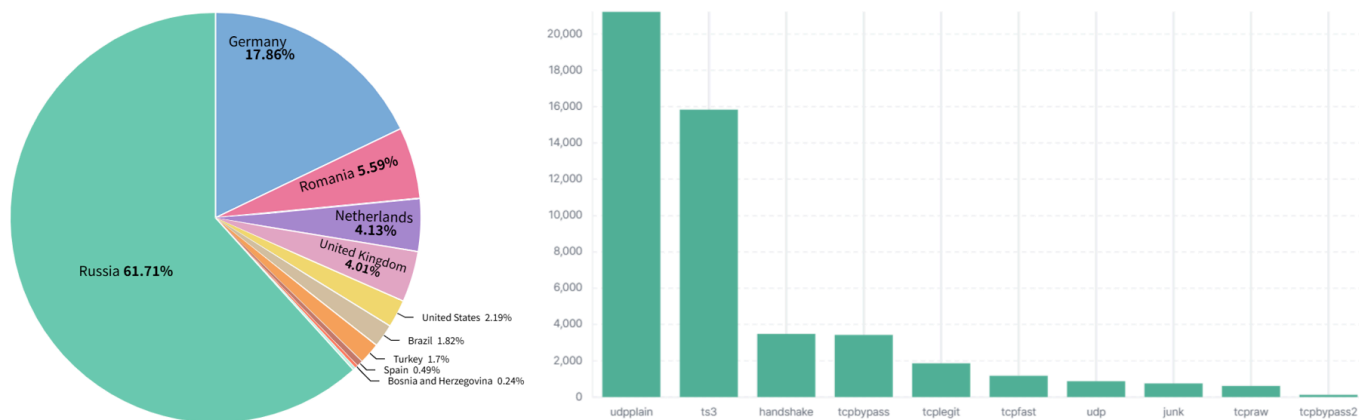


図6. DoS攻撃先の国別割合と攻撃ベクタ(秒)の分布

0020	00 c8 4e ab 30 39 00 2a 79 c5 54 53 33 49 4e 49	..N.09.* y.TS3INI
0030	54 31 00 65 00 00 88 13 dc 26 29 00 67 14 00 f6	T1.e.... .&).g...
0040	32 f6 32 19 00 01 00 00 00 00 00 00	2.2....

図7. TeamSpeak3攻撃ベクタのUDPパケット(C2エミュレートによる)

感染規模

ダウンロードサーバのログデータが公開状態になっていた時期があり、ログによると9月時点の MooBot の感染機器は 3160 台で、国別、および、感染機器の DVR などの名称を示すと推定される引数別の割合は以下の通りです。

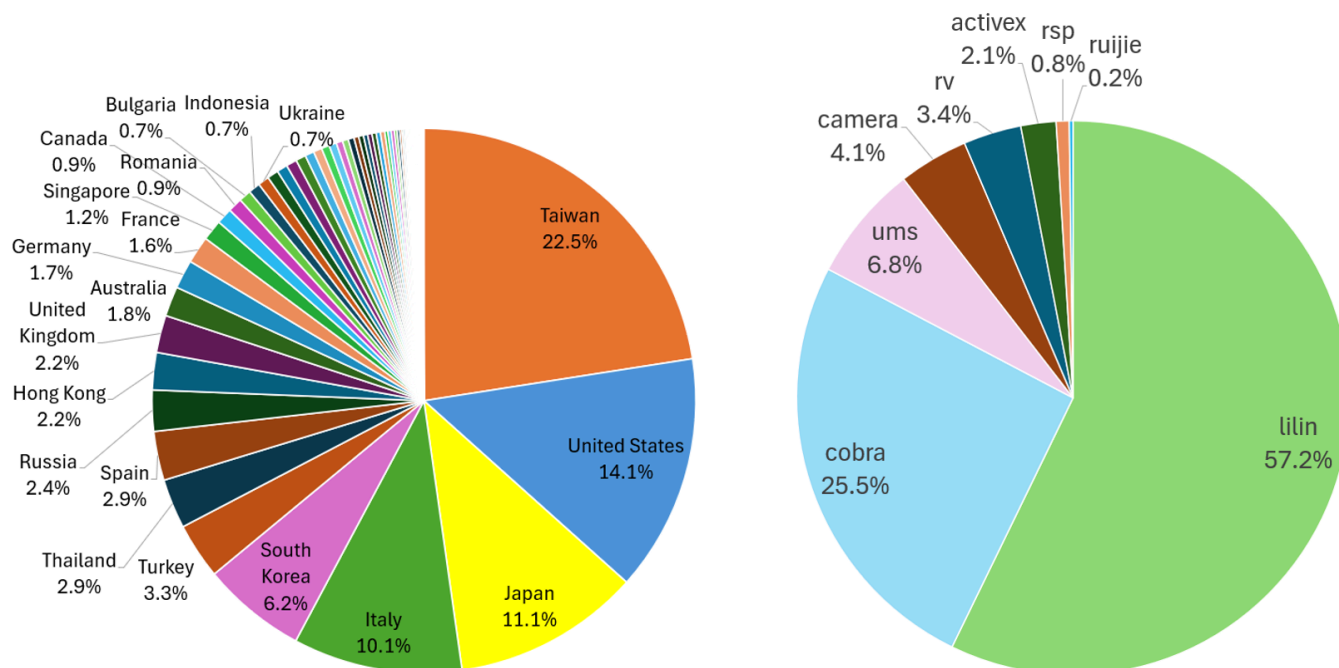


図8. 感染デバイスの統計

このログを出力するダウンロードサーバは、機器によって引数とダウンロードポートを使い分けるという手法で稼働していました。それ以前の固定ポート25565、80を使用する方法とは異なるため、この感染規模は全体の一部であると考えられます。6月にTrend Micro社やCensys社から報告されている、Langflowサーバにおいて使用された引数は無く、DVR やルータをターゲットとした活動で使用されたダウンロード方式のようです。

```
# Порт -> аргумент для запуска прошивки
self.port_to_arg = {
    1: 'funny',
    27015: 'activex',
    27016: 'camera',
    27017: 'cobra',
    27018: 'lilin',
    27019: 'lilin',
    27020: 'rv',
    27021: 'ruijie',
    27022: 'ums',
    27024: 'usarouter',
    27026: 'xfinity',
    27027: 'epon',
    27028: 'rapid',
    #27029: 'wcp',
    27030: 'rsp',
    27031: 'xpon'
}
```

図9. ダウンロードポートと引数の変換テーブル

また、変換テーブルの手前にロシア語のコメント文がありますが、このように MooBot ではロシア語圏で開発が行われているように見える痕跡が多く見つかります。

活動の終了

このボットネットは、9月に活動を再開してから約1ヶ月の間に10回以上のマルウェア更新を行い、仕様が定まらないまま、最終的に同じ C2 サーバを使いながらも異なるポートを使用する別のマルウェアに切り替わり、10月7日には活動を終了しました。この別のマルウェアは、MountBot と同様に `/proc/self` を `/proc/1` にマウントして自身のプロセスを隠蔽する手法を取り入れています。DoS 攻撃コマンドの処理を実装していないなど、まだ作り始めたばかりに見える未完成な状態のものでした。

短期間で活動を終了した理由は分かりませんが、9月から10月にかけて観測した DoS 攻撃の時間帯は、明らかに特定の地域の活動時間帯に偏っており、ロシア語圏で開発されている可能性があることも考慮すると、DDoS サービスを運用していたというよりは、開発、もしくは実験的な活動だったのではないかと推測しています。

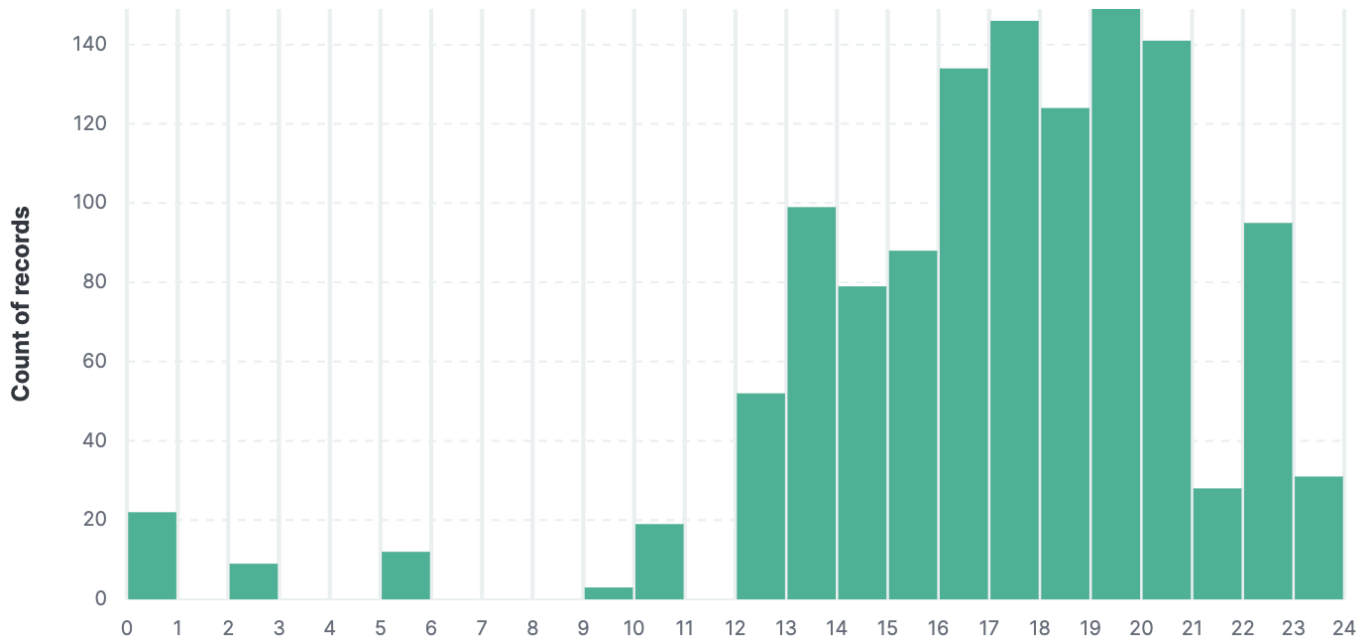


図10. DoS攻撃の時間帯 (UTC)

IoC

マルウェア

SHA256

更新月 ハッシュ値

5月 9dd852b9ddff7e1c2fbaa8e75e2b4905c100de1e83ce999157ece4e797484585

6月 a6cf8124e9b4558aacc7ddfa24b440454b904b937929be203ed088b1040d1b36

9月 b7083abcf87e6192904a08629e2f9e038b712a66132578df76ca9175d80e2841

10月 b622a7b848a9846e2af4f8f985777bb8d095b8eb56d57a502f97d446cfe82e9b

MD5

更新月 ハッシュ値

5月 933d24c8b558a03b7285647875b920d5

6月 0473c6b98ad7273ab3e9a9fb6f00ea54

9月 7f78c49eaa67ce53044712c0aaf5c5f8

更新月 ハッシュ値

10月 [c974c2bfb758c06d61030b8af0ce270e](#)

ダウンロードサーバ

更新月 IP

5月 [80\[.\]66\[.\]75\[.\]121:25565](#)

6月 同上

9月 [80\[.\]66\[.\]75\[.\]121:80](#)

10月 [78\[.\]153\[.\]140\[.\]92:80](#)

C2サーバ

更新月 IP / ドメイン

5月 [188\[.\]166\[.\]68\[.\]21:54707](#) [206\[.\]71\[.\]149\[.\]179:54707](#)
[45\[.\]61\[.\]137\[.\]226:54707](#)

6月 同上

9月 [193\[.\]168\[.\]196\[.\]38:60177](#) [5\[.\]144\[.\]176\[.\]160:60177](#)
[85\[.\]209\[.\]134\[.\]80:60177](#)

10月 [blueblackside\[.\]com:60177](#)

1. <https://blog.netlab.360.com/the-botnet-cluster-on-185-244-25-0-24-en/>
2. <https://www.fortinet.com/blog/threat-research/mirai-based-botnet-moobot-targets-hikvision-vulnerability>
3. https://www.trendmicro.com/en_us/research/25/f/langflow-vulnerability-flodrix-botnet.html
4. <https://censys.com/blog/poking-the-flodrix-botnet>

[MooBot](#) [Flodrix](#)

[← 前へ](#)