# 诱骗 IoT 恶意软件跟踪 C&C 服务器

## 工作来源

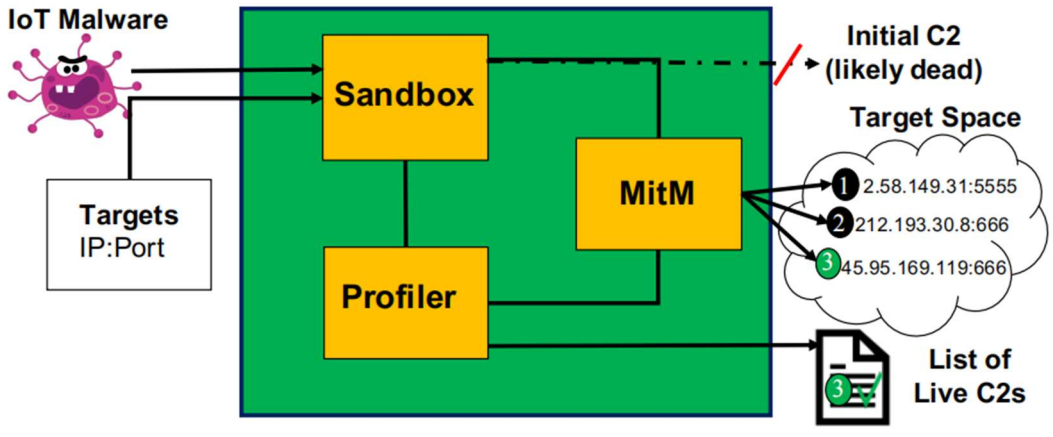
ASIA CCS 2024

## 工作背景

在分析 IoT 僵尸网络时，识别 C&C 服务器至关重要。C&C 服务器的 IP 地址一直都是商业威胁情报的重要组成部分，由于 C&C 服务器通信协议日渐复杂并且活跃周期较短，时效性和准确性也非常重要。如果可以自动化识别 IoT 恶意软件使用的 C&C 服务器，能够提供极有价值的威胁情报。

## 工作设计

首先在沙盒中执行 IoT 恶意软件触发 C&C 请求，紧接着 Profiler 在所有流量中剥离出 C&C 流量，再通过 MitM 将流量重定向至给定的 IP:端口空间范围，最后通过分析通信确定目标是否为 C&C 服务器。




### 激活 C&C 流量

要按照样本文件的架构使用对应架构的环境，使用 QEMU 和 RiotMan 来模拟所需环境。

| Type | Breakdown | Lines of Code (LOC) |
|---|---|---|
| Programming Language | Shell | 636 |
| | Python | 2,897 |
| C2Miner Module | Sandbox | 1,239 |
| | MitM/Probing | 553 |
| | Profiler | 1,231 |
| | Other | 510 |

**C&C 流量剥离**

算法如下所示。首先过滤掉不相关的协议（ICMP、DHCP、ARP、NTP 等），接着计算与对端的连接次数/请求解析次数大不大。针对域名要过滤掉信誉高的域名，针对 IP:端口计算连接频率与对端端口号出现的 IP 地址数。

**Input:** Packets
**Output:** Scores ▷ for IP:ports

$TargetStats \leftarrow \{\}$ ▷ A hashtable tracking the number of connections to each target (ip:port or DNS).
$Ports \leftarrow \{\}$ ▷ A hashtable tracking the number of times a destination port is seen.
$Scores \leftarrow [\,]$ ▷ A list of targets with their C2 likelihood score.
**for each** $pkt \in Packets$ **do**
  **if** $Approved(pkt) == TRUE$ **then**
    $target \leftarrow Get\_Target(pkt)$
    $Update\_Target(target, TargetStats)$
    $Update\_Ports(target, Ports)$
**for each** $target \in TargetStats$ **do**
  **if** $is\_DNS(target)$ **and not** $White\_list(target)$ **then**
    $Scores[target] \leftarrow Calc\_DNS\_Score(target)$
  **if** $is\_IP(target)$ **then**
    $Scores[target] \leftarrow Calc\_IP\_Score(target, Ports)$
$Sort\_Desc(Scores)$
**return** $Scores$

**重定向探测**

通过中间人模块来重定向流量，出于伦理考量需要做映射和限制。

**确定 C&C 服务器**

在传输层使用两种方式：

（1）SYN-DATA 感知。在 SYN 设置低于阈值且存在数据交换时，即可认为是

C&C 服务器。即便握手也不代表在应用层面连接成功，但失败后一直重试也可能是 C&C 服务器。

（2）指纹识别。将网络流建模成对话的模式：确定流的开始与结束，保证流符合 TCP 标准规范，再提取数据包属性。算法如下所示，属性可以是数据包大小、字节熵值或者 Payload 中的字符串等。通过这种方式，可以将每个流都转换成字符串。转换成字符串后，比较字符串间的相似度（最长公共序列，LCS）再进行聚类（K-Means）。

$$
\begin{aligned}
Flow &\longrightarrow handshake\ FlowBody && (1)\\
FlowBody &\longrightarrow DataPacket\ FlowBody && (2)\\
FlowBody &\longrightarrow ControlPacket\ FlowBody && (3)\\
FlowBody &\longrightarrow \epsilon && (4)\\
DataPacket &\longrightarrow Sender"\_"Attributes && (5)\\
ControlPacket &\longrightarrow Sender"\_"Flags && (6)\\
Sender &\longrightarrow client|server && (7)\\
Flags &\longrightarrow Flag\ Flags|Flag && (8)\\
Flag &\longrightarrow ACK|SYN|RST|FIN && (9)\\
Flag &\longrightarrow \epsilon && (10)\\
Attributes &\longrightarrow attr* && (11)
\end{aligned}
$$

---

H . CLIENT_11 . SERVER_4 . CLIENT_1

H . CLIENT_2 . SERVER_13 . SERVER_FIN_ACK . CLIENT_FIN_ACK

第一个流中，完成握手后客户端发送 11 字节数据，随后服务器回复 4 字节数据，客户端响应 1 字节数据结束。第二个流中，完成握手后客户端发送 2 字节数据，随后服务器回复 13 字节数据，紧接着服务器与客户端互相确认 FIN 流结束。

## 工作准备

C2Miner 使用 Python 和 Shell 编写，总计约 3500 行代码。

除了向 C&C 服务器发送的流量之外，全部都被过滤掉了。

从 MalwareBazaar 与 VirusTotal 中收集 1447 个 MIPS 架构的样本，平均每天可收集到四个新样本。

| Dataset | Description | Section |
|---|---|---|
| DAll | 1,447 binaries collected in total | Base |
| Ground1 | 241 malware binaries used as ground truth | §5.3 |
| Ground2 | 1,083 binaries and their IP:port C2 address cross verified by VirusTotal | §5.3 |
| Ground3 | 202 binaries of Ground1 with a live C2 used as ground truth | §5.4 |
| Trace-1 | 317MB traffic of 80 binaries from Ground3 redirected to 34 C2 servers and 39 benign servers | §5.4 |
| DFinger | 202 traffic fingerprints in our formal grammar of the 202 binaries from Ground3 | §5.5 |
| Trace-2 | 230MB traffic of 49 binaries from Ground3 redirected to “talk” to 32 C2 servers in January 2022 | §5.6 |

分析期间，样本向 15 万个“IP:端口”组合发出了 300 万次请求。一共有 202 个样本文件与 C&C 服务器建立了连接，生成了 230MB 流量（其中，C&C 流量只占 0.06%）。

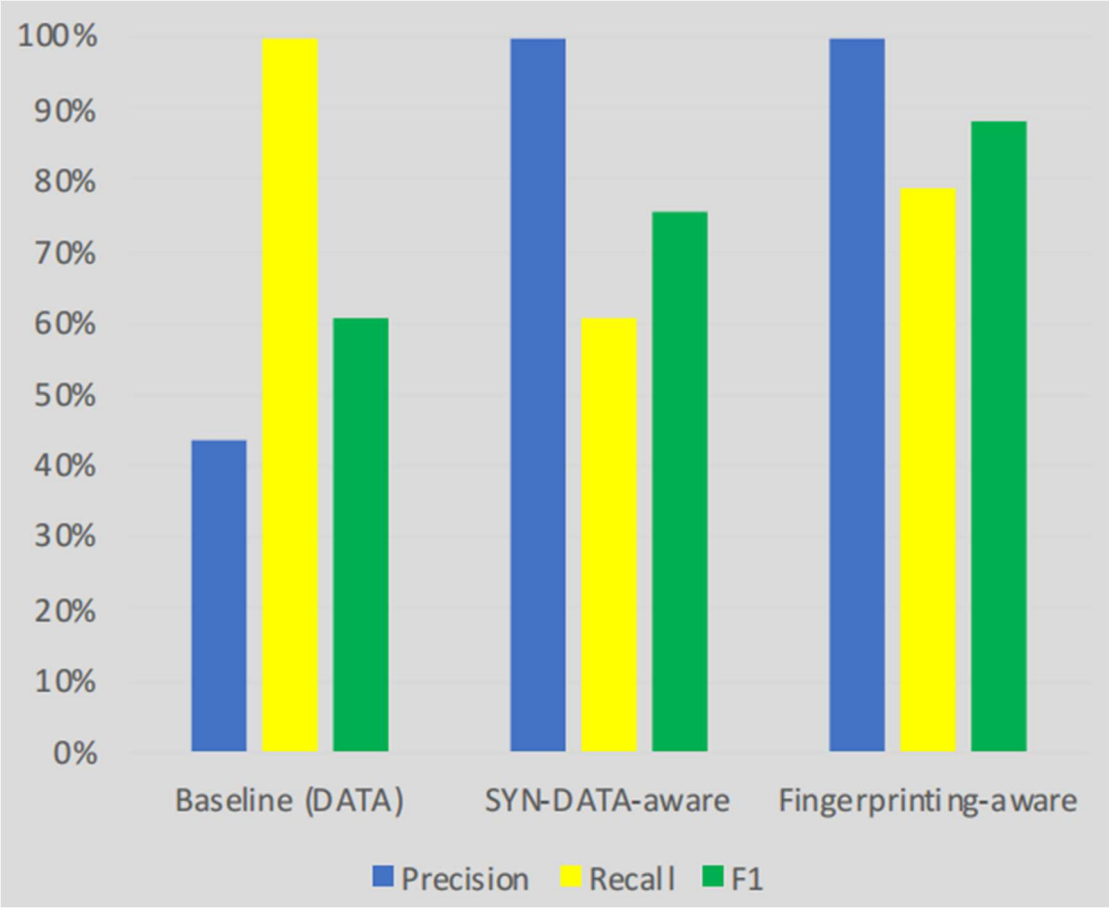
| Malware | Communication | Details |
|---|---|---|
| Gafgyt | Custom | PONG command is communicated via IRC, others are text commands. |
| Mirai | Custom | All C2 commands are custom binary based. |
| Lightaidra | IRC | All C2 commands are wrapped inside IRC PRIVMSG (private) messages. |
| Remaiten | IRC | Similar to Lightaidra but commands are different. |
| Lizkebab | Custom | Similar to Gafgyt but commands are different. |
| LuaBot | Encrypted Payload | Uses MatrixSSL lib for encryption. |
| Tsunami | IRC | All C2 commands are wrapped inside IRC NOTICE messages. |
| BASHLIFE | Custom | Similar to Gafgyt but commands are different. |

这些样本覆盖了 11 个恶意软件家族，但 MIPS 样本的杀软标签很不准确。

| Family | Not Packed | UPX | Modified UPX | Total |
|---|---|---|---|---|
| Mirai | 449 (59.63%) | 273 (36.25%) | 31 (4.12%) | 753 |
| Gafgyt | 227 (82.55%) | 22 (8.00%) | 26 (9.45%) | 275 |
| Xored | 224 (98.25%) | 4 (1.75%) | 0 (0.00%) | 228 |
| P2P | 0 (0.00%) | 70 (100.00%) | 0 (0.00%) | 70 |
| Bash | 3 (100.00%) | 0 (0.00%) | 0 (0.00%) | 3 |
| Dakkatoni | 0 (0.00%) | 28 (100.00%) | 0 (0.00%) | 28 |
| Tsunami | 9 (75.00%) | 0 (0.00%) | 3 (25.00%) | 12 |
| Lightaidra | 53 (100.00%) | 0 (0.00%) | 0 (0.00%) | 53 |
| Daddyl33t | 10 (100.00%) | 0 (0.00%) | 0 (0.00%) | 10 |
| VPNFilter | 2 (100.00%) | 0 (0.00%) | 0 (0.00%) | 2 |
| Hajime | 13 (100.00%) | 0 (0.00%) | 0 (0.00%) | 13 |
| Total | 990 (68.42%) | 397 (27.44%) | 60 (4.15%) | 1,447 |

## 工作评估

对比之下，最为稳定的就是指纹识别的方法。




聚类也可以很好地表征恶意软件家族的行为，但并不完美。大多数 Mirai 的样本模式为 CLIENT_4.CLIENT_1.CLIENT_2.SERVER_2，大多数 Gafgyt 的样本模式为 SERVER_4.SERVER_1.SERVER_4.SERVER_1，这二者就覆盖率 68% 的 C&C 通信。

| | Mirai | Gafgyt | Daddyl33t | Xored | Light/ra | Hajime | Tsunami |
|---|---|---|---|---|---|---|---|
| #1 | 25% | **68%** | 20% | 34% | 0% | 50% | 0% |
| #2 | **46%** | 0% | 40% | **66%** | 14% | 0% | 0% |
| #3 | 18% | 26% | 40% | 0% | **86%** | 0% | 0% |
| #4 | 11% | 6% | 0% | 0% | 0% | 50% | **100%** |

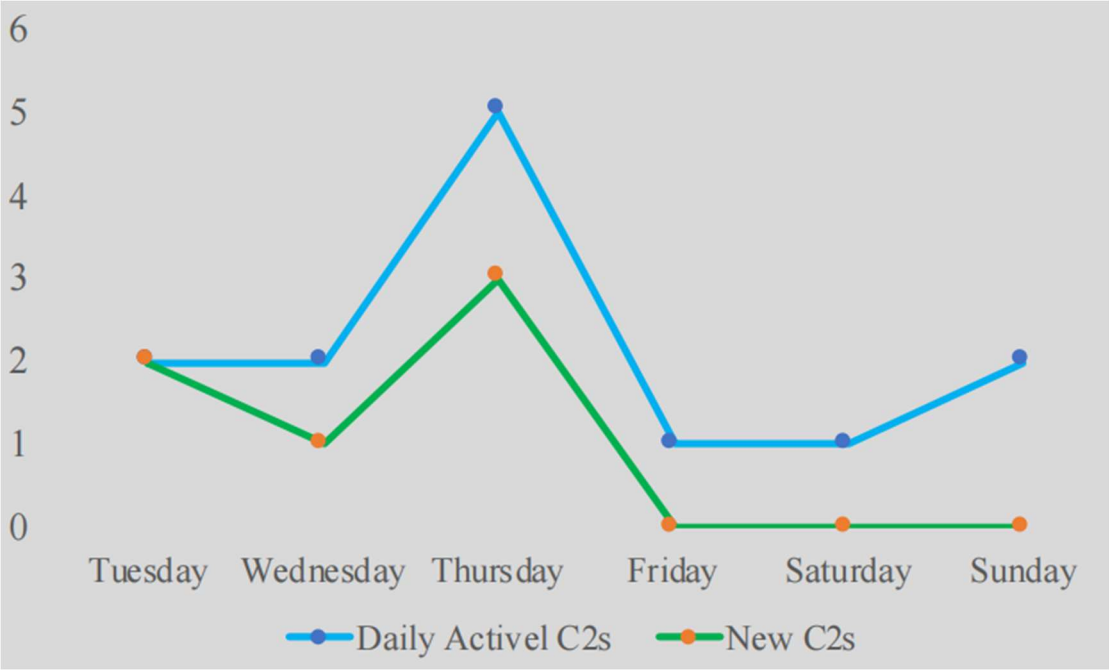
该方法对 84% 的样本文件都有效。

## 工作思考

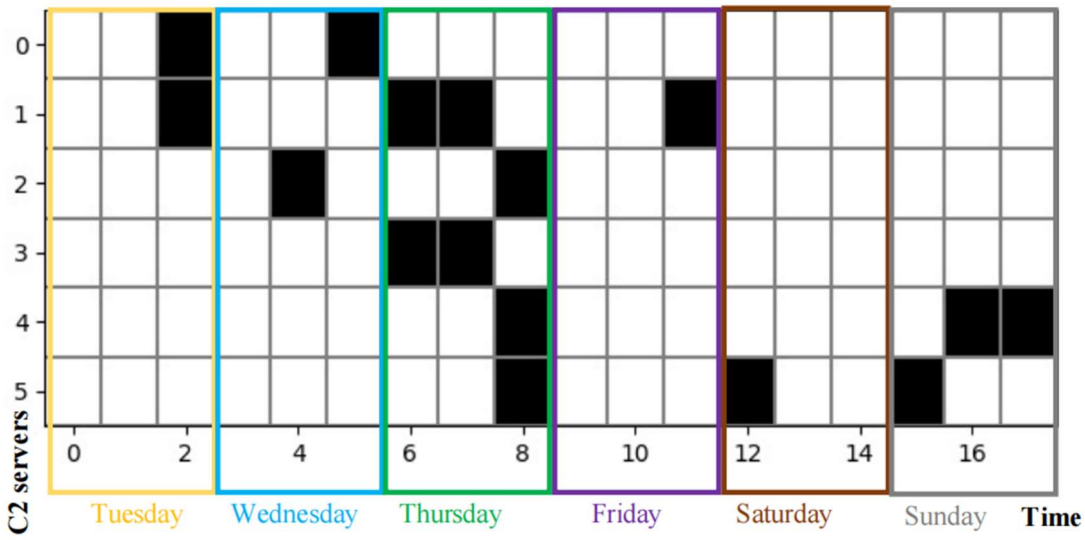
根据既定模式在大网展开探测，C2Miner 发现了好几个活跃的 C&C 服务器。

| Parameter | Values |
|---|---|
| Subnets | 136.144.41/24, 195.133.40/24, 2.58.149/24, 212.193.30/24, 107.173.176/24, 45.95.169/24 |
| Ports | 1312, 666, 1791, 9506, 606, 6738, 5555, 1014, 3074, 6969, 42516, 81 |
| Sample(s) | Gafgyt 46501d723f368c22e5401f7c95d928ab |
| Sample(s) | Mirai 800af659256f0232a27f955a4430aed0 |
| Live C2 Servers | 2._._.34:5555, 212._._.91:666, 45._._.119:666, 136._._.240:666, 212._._.123:5555, 107._._.144:42516 |

尽管每天识别出一个活跃 C&C 服务器，但受制于各方条件限制已经相当好了。




即使是正确的 C&C 服务器，也未必一定会正确响应。


C2 servers
0
1
2
3
4
5
0
2
4
6
8
10
12
14
16
Tuesday
Wednesday
Thursday
Friday
Saturday
Sunday
Time