IoT Botnet Forensics: A Comprehensive Digital Forensic Case Study on Mirai Botnet Servers

Xiaolu Zhang, Oren Upton, Nicole Lang Beebe, Kim-Kwang Raymond Choo

Department of Information Systems and Cyber Security, **University of Texas at San Antonio**

UTSA The Cyber Center for Security & Analytics

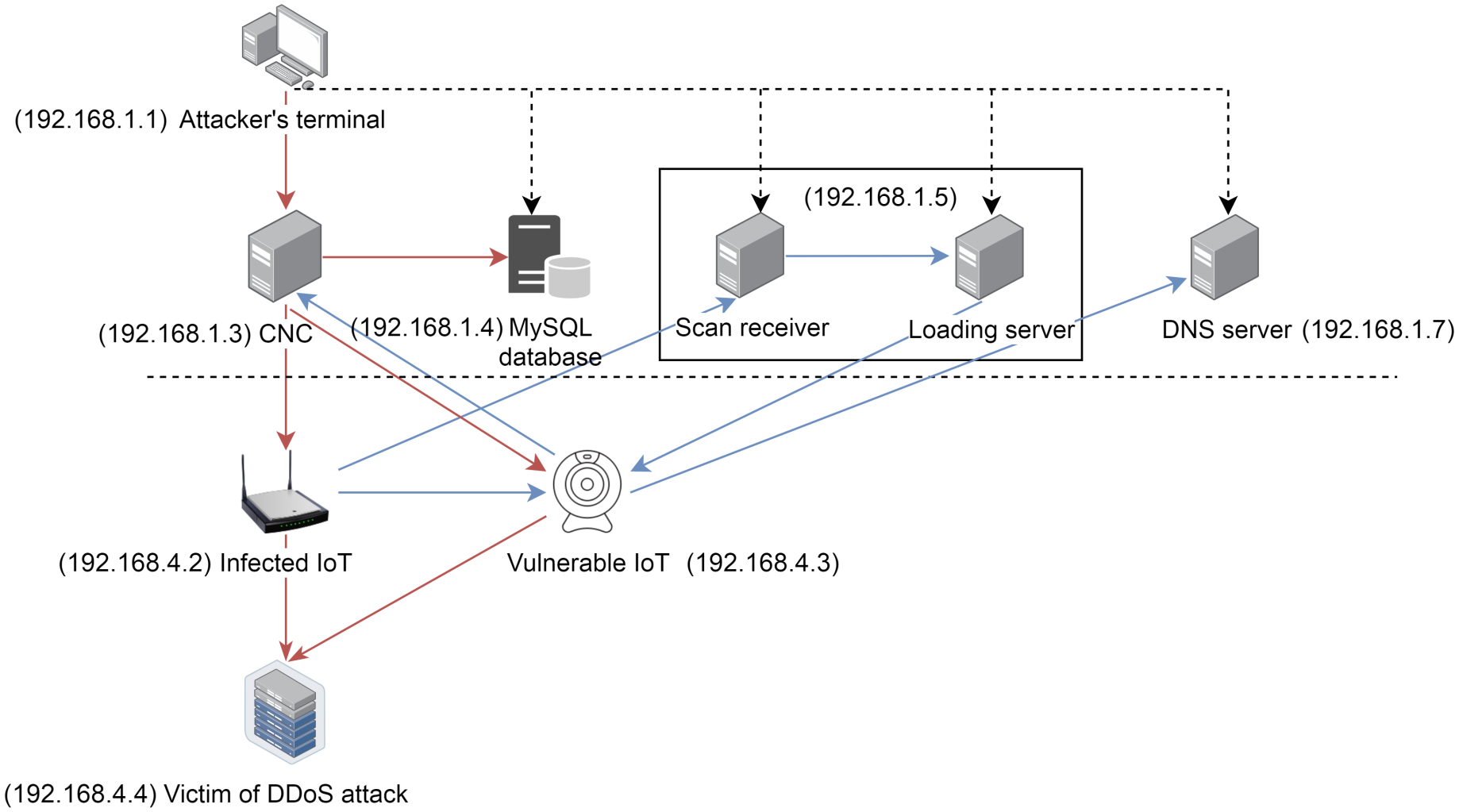
NSF

CACI EVER VIGILANT

Most important features of Mirai

- An IoT malware and a centralized Botnet
- Caused those most famous IoT DDoS attacks
- Open sourced and fast-growing

Mirai botnet structure



Motivation

- Many investigations/research of Mirai to date have focused on a traditional malware analysis of the executable code found on infected IoT devices.
- As Mirai is open sourced, there has been increasing abuse of Mirai's source code. Someone lacking the expertise to write an IoT botnet can easily build their own Mirai botnet for a DDoS attack. In this case, a forensic investigator might be involved in a case where the control server of a Mirai botnet is captured.

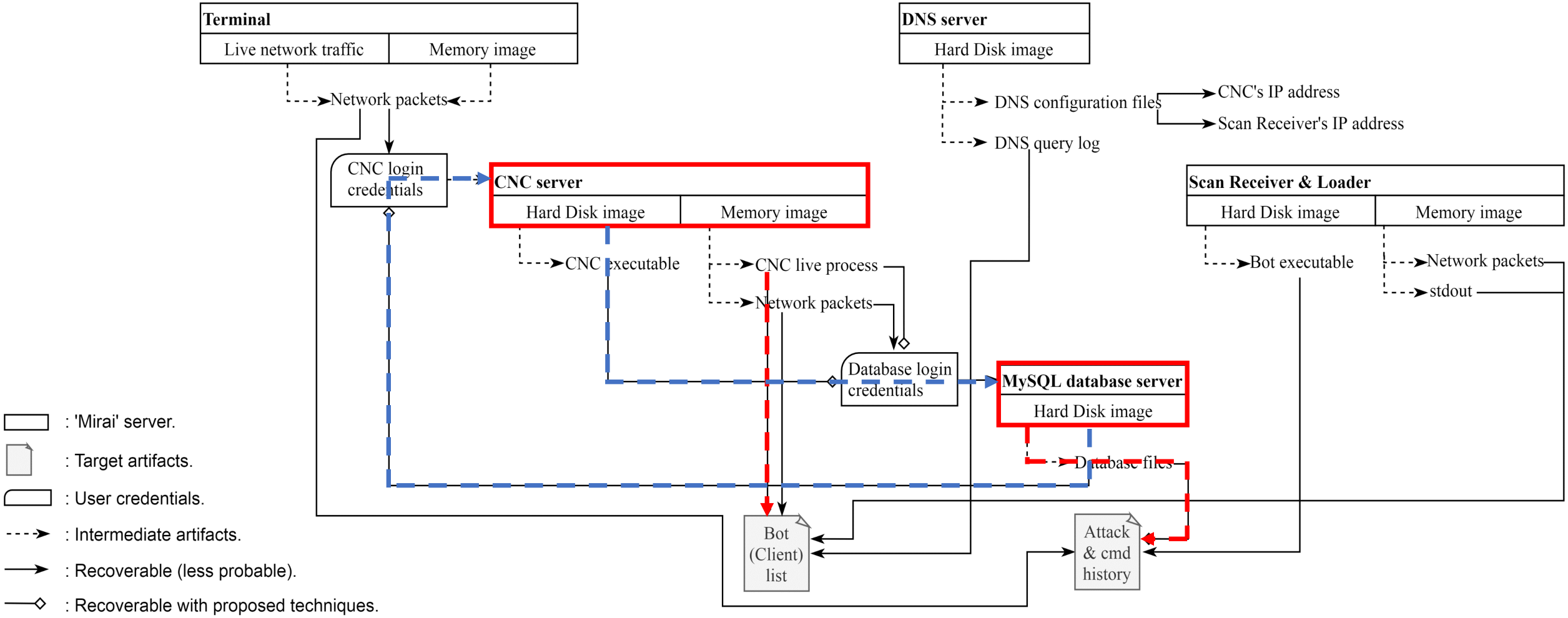
Research questions

- What forensic approaches could work on the botnet servers?
- What evidence is retrievable from the servers?
- Where is the evidence located?
- What investigative information could be obtained from the evidence?

Methodology

- Build our own local Mirai botnet.
- Acquire data from the file system, RAM, and network traffic for each physical server.
- Apply manual analysis on the data source acquired in the preceding step. Specifically targeting:
  1. The historical record of the achieved attacks
  2. The victim/target of the DDoS attack
  3. The information about the infected bots
- Incorporate the findings into a Road Map.

The road map for Mirai botnet server forensics



Key findings on CNC server

- CNC source code (Unlikely) → User credentials of the Database
- CNC executable (Disk image) → User credentials of the Database
- CNC live process (Memory dump) → Bot list

CNC source code

→ User credentials of the Database

/Mirai-Source-Code/mirai/cnc/main.go

```
const DatabaseAddr string = "192.168.1.4"
const DatabaseUser string = "db-login-usrname"
const DatabasePass string = "db-login-passwd"
const DatabaseTable string = "mirai"
```

CNC executable

→ User credentials of the Database

- By reverse engineering CNC executable (written in **Go Lang**), we proposed how to recover the Database server's user credentials.

```
LEA     RAX,[DAT_00666568]                          = 31h    1
MOV     qword ptr [RSP]=>local_90,RAX=>DAT_00666568  "192.168.1.4"
MOV     qword ptr [RSP + local_88],0xb              char[11]
LEA     RAX,[DAT_006678ab]                          "db-login-usrname"
MOV     qword ptr [RSP + local_80],RAX=>DAT_006678ab = 64h    d
MOV     qword ptr [RSP + local_78],0x10             char[16]
LEA     RAX,[DAT_00667442]                          "db-login-passwd"
MOV     qword ptr [RSP + local_70],RAX=>DAT_00667442 = 64h    d
MOV     qword ptr [RSP + local_68],0xf              char[15]
LEA     RAX,[DAT_0066525e]                          "mirai"
MOV     qword ptr [RSP + local_60],RAX=>DAT_0066525e = 6Dh    m
MOV     qword ptr [RSP + local_58],0x5              char[5]
CALL    main.NewDatabase                            NewDatabase(
                                                    DAT_00666568
                                                    DAT_006678ab
                                                    DAT_00667442
                                                    DAT_0066525e)
```

CNC live process

→ Bot list

- CNC server retains a queue of live Bots in RAM only.
- By tracking the “Bot” data structure in RAM the IP address of the live bots could be recovered.

Key findings on Database server

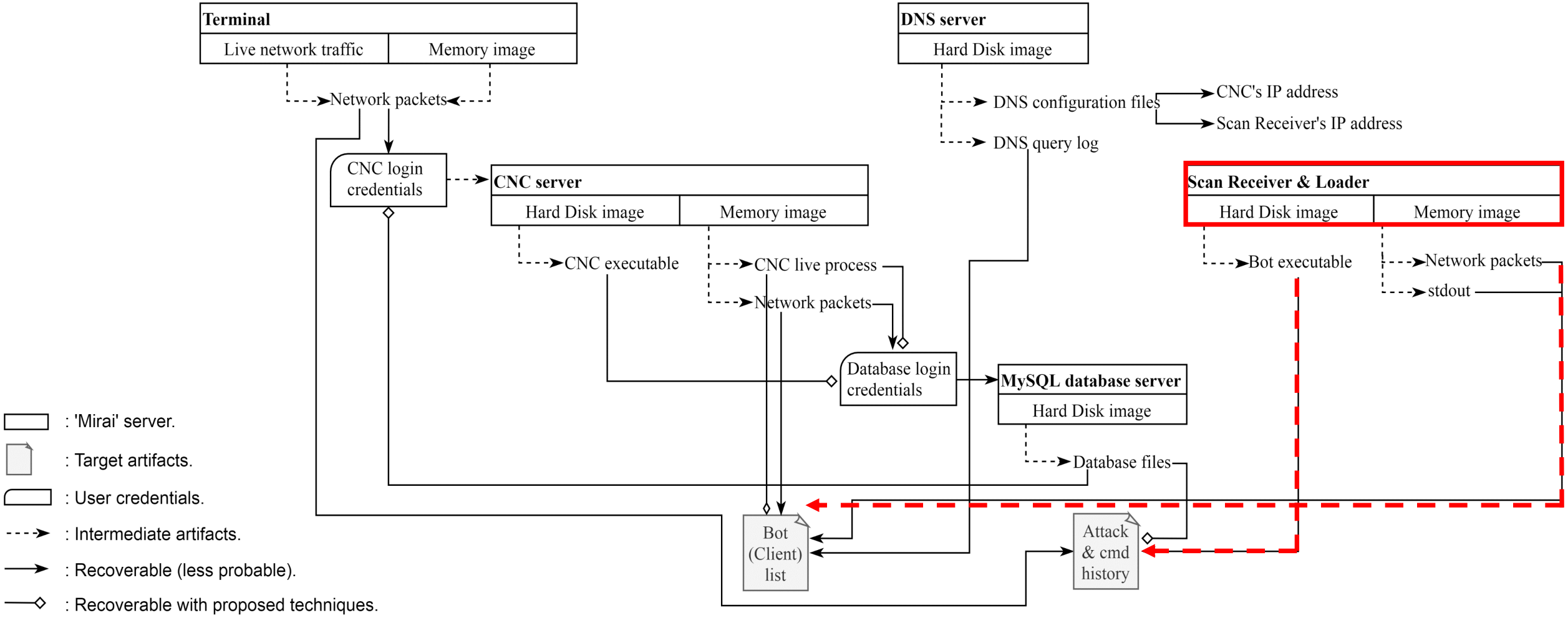
→ CNC User credentials and Command history

- If the database server is captured, CNC is accessible remotely.

id	user_id	time_sent	duration	command	max_bots
1	1	1563551713	192	ack 192.168.4.4 192	-1
2	1	1563569141	180	syn 192.168.4.4 180	-1
3	1	1563570397	10	ack 192.168.4.4 10	-1
4	1	1563576388	20	http 192.168.4.4 20	-1
5	1	1563589661	10	http 192.168.4.4 10	-1
6	1	1563590219	5	http 192.168.4.4 5	-1
7	1	1563590424	11	udp 192.168.4.4 11	-1
8	1	1563590864	100	http 192.168.4.4 100	-1
9	1	1563592046	64	stomp 192.168.4.4 64	-1
10	1	1563604305	99	udp 192.168.4.4 99	-1
11	1	1563617689	99	http 192.168.4.4 99	-1
12	1	1563618389	99	http 192.168.4.4/24 99	-1
13	1	1563618501	99	http 192.168.4.4/24 99	-1
14	1	1563632476	100	http 192.168.4.4/24 100	-1
15	1	1563643742	99	udp 192.168.4.4/24 99	-1
16	1	1563682051	100	udp 192.168.4.4/24 100	-1
17	1	1564710267	1000	udp 192.168.4.4/24 1000	-1
18	1	1565641307	100	udp 192.168.4.4/24 100	-1

id	username	password	duration_limit	cooldown	wrc	last_paid	max_bots	admin	intvl	api_key
1	mirai-user	mirai-pass	0	0	0	0	-1	1	30	

The road map for Mirai botnet server forensics



Key findings on Scan Receiver & Loader

→ Bot list

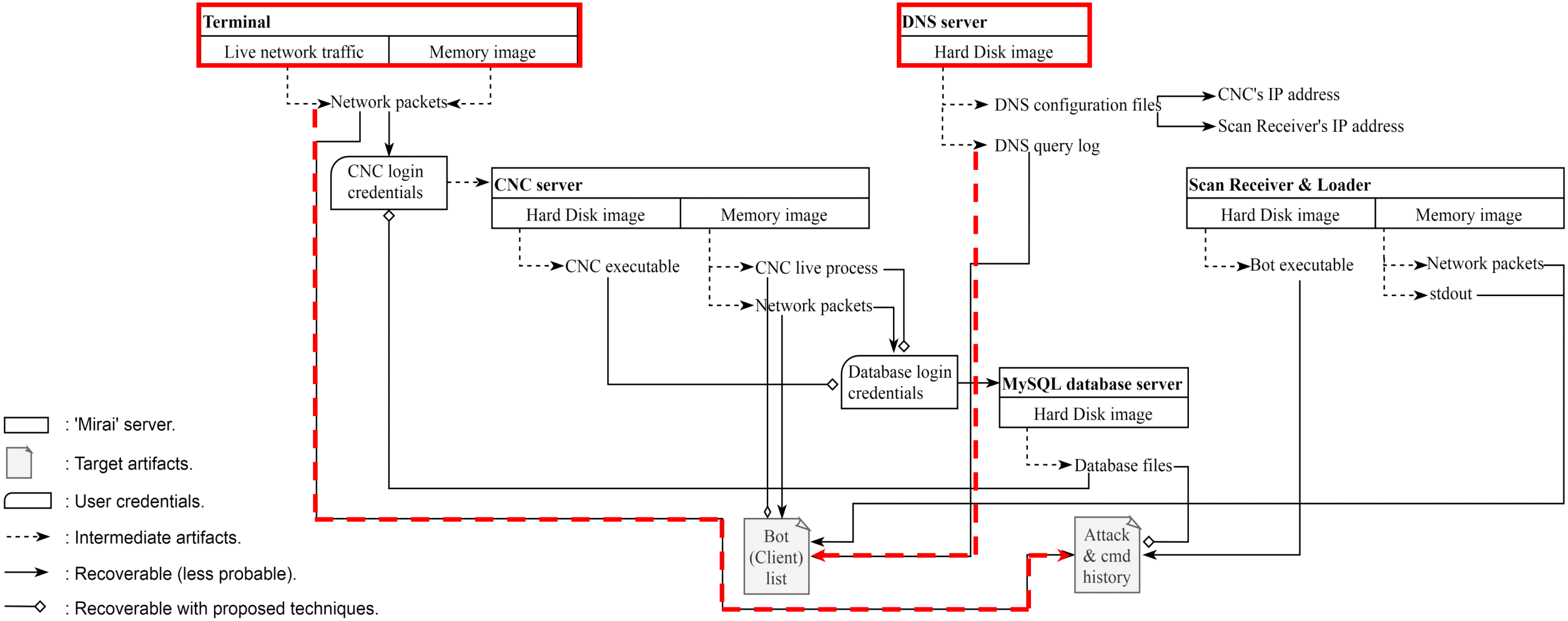
- The standard output stream (or ‘stdout’) is where the Loader acquires the information of a vulnerable IoT device reported by a bot

```
64 65 62 75 67 23 20 2E  2F 73 63 61 6E 4C 69 73  debug# ./scanLis
74 65 6E 20 0A 31 39 32  2E 31 36 38 2E 34 2E 33  ten  192.168.4.3
3A 32 33 20 72 6F 6F 74  3A 70 61 73 73 0A 31 39  :23 root:pass 19
32 2E 31 36 38 2E 34 2E  33 3A 32 33 20 72 6F 6F  2.168.4.3:23 roo
74 3A 70 61 73 73 0A 0A  00 00 00 00 00 00 00 00  t:pass
```

→ Bot Executable

- a Loader must store bot executables for infecting the vulnerable IoT devices

The road map for Mirai botnet server forensics



Key findings on DNS server

→ CNC server and the Scan Receiver's IP address

→ Client (bot) list

A Bot who queried the CNC's IP address.

```
Aug 16 16:55:22 cnc named[515]: client @0x7ff9140c72a0 192.168.4.2#42576
(www.mirai.com): query: www.mirai.com IN A + (192.168.1.7)
Aug 16 16:55:50 cnc named[515]: client @0x7ff9140c72a0 192.168.4.2#55160
(report.mirai.com): query: report.mirai.com IN A + (192.168.1.7)
```

CNC's domain name

Scan Receiver's domain name

Summary

- This research intends to fill the gap where no existing study has performed a digital forensic analysis on IoT botnet servers.
- This research provides findings tactically useful to forensic investigators, not only from the perspective of what data could be obtained but also important information about which device they should target for acquisition and investigation to obtain the most investigatively useful information.

Thank you

Xiaolu Zhang	xiaolu.zhang@utsa.edu
Oren Upton	oren.upton@utsa.edu
Nicole Lang Beebe	nicole.beebe@utsa.edu
Kim-Kwang Raymond Choo	raymond.choo@fulbrightmail.org

UTSA The Cyber Center for Security & Analytics

NSF

CACI EVER VIGILANT