

\$ Utiliser l'IA localement

>> Comment garder le contrôle de vos données

# \$ Whoami

>>> Simon Nolet (Viper)

- Fondateur de Hacktive Education
- 250+ Pentest
- 10 Ans sécurité offensive
- Formateur Hackfest (5ans)
- Createur de défi Hackfest (7 ans)

# \$ Table des Matières

>> Programme de la Présentation

>>> Introduction & Concepts

>>> Outils & Technologies

>>> \*\* Mise en Pratique\*\*

# \$ Utiliser l'IA localement, comment garder le contrôle de vos données

## >> Contrôle total de vos données

- Confidentialité
- Sécurité
- Coûts réduits

# \$ Ce qui marche bien

## >> Recommandé

- Génération de code
- Analyse de données
- Résumés de documents et contenu
- Automatisation

| Le produit final reste l'objectif

# \$ Mauvais usage de l'IA

## >> À éviter

- Social
- Relations interpersonnelles
- Remplacer la pensée critique

| Si la valeur vient de l'humain, gardez l'humain

# \$ Forces vs Faiblesses

>>



Excellent pour

- Volumes massifs
- Reconnaissance de patterns
- Automatisation
- Sommaire de contenu

>>



## Limité pour

- Recette de cuisines
- Pensée Divergente
- Empathie réelle

## \$ Quoi demander à l'IA

Demander seulement du contenu que vous avez les connaissances pour valider

- Restez critique sur la réponse

# \$ Mon parcours IA



>> Débuts avec ChatGPT

>>> Évolution

1. Découverte
2. Expérimentation
3. Correction
4. Traduction

\$ Fabric



## >> Framework d'automatisation IA



[github.com/danielmiessler/fabric](https://github.com/danielmiessler/fabric)

- Standardisation des interactions
- Patterns réutilisables
- Intégration shell

# \$ Fabric + Ollama



>> 100% Local

```
go install github.com/danielmiessler/fabric@latest
fabric --setup
./fabric -y https://www.youtube.com/watch?v=a5040rYhqWo -sp extract_wisdom
dmesg | fabric -sp analyze_logs
```

\$ ollama



# \$ Patterns Fabric

## >> Structure standardisée

IDENTITY: Expert en sécurité

INPUT: \${security\_log}

INSTRUCTIONS: Analyse et recommande

OUTPUT: JSON structuré, Markdown etc

## >>> Populaires

- `summarize`
- `extract_wisdom`
- `analyze_logs`

# \$ Extract Wisdom

[https://github.com/danielmiessler/fabric/blob/main/patterns/extract\\_wisdom/system.md](https://github.com/danielmiessler/fabric/blob/main/patterns/extract_wisdom/system.md)

# \$ System Prompt

## >> Exemple

### # IDENTITÉ

You are an expert at presenting at quebecsec

### # OBJECTIF

Communicate clearly and present your recent learnings

### # INSTRUCTIONS

1. Present your Ideas
2. Reframe in simpler term
3. Create slides to present well the idea
4. Ensure a consistent theme amongs the slides
5. Take a deep breath

\$ OpenWebUI

>> Interface web locale

- Interface ChatGPT-like
- Compatible Ollama
- Multi-utilisateurs
- Exécution de code

# \$ Demo - OpenWebUI



\$ N8N



## >> Automatisation visuelle (Nocode)

n8n.io

- Workflows drag & drop
- 800+ intégrations
- IA native
- Auto-hébergeable

# \$ N8N

- Sustainable Use License
- Code disponible
- Free for personal use

\$ N8N



# \$ N8N + IA

## >> Cas d'usage

- Chatbot sur context (Youtube, Auteur, Documentation)
- Classeur de dépense
- Chatbots personnalisés
- Surveillance et alertes

\$ Démo N8N

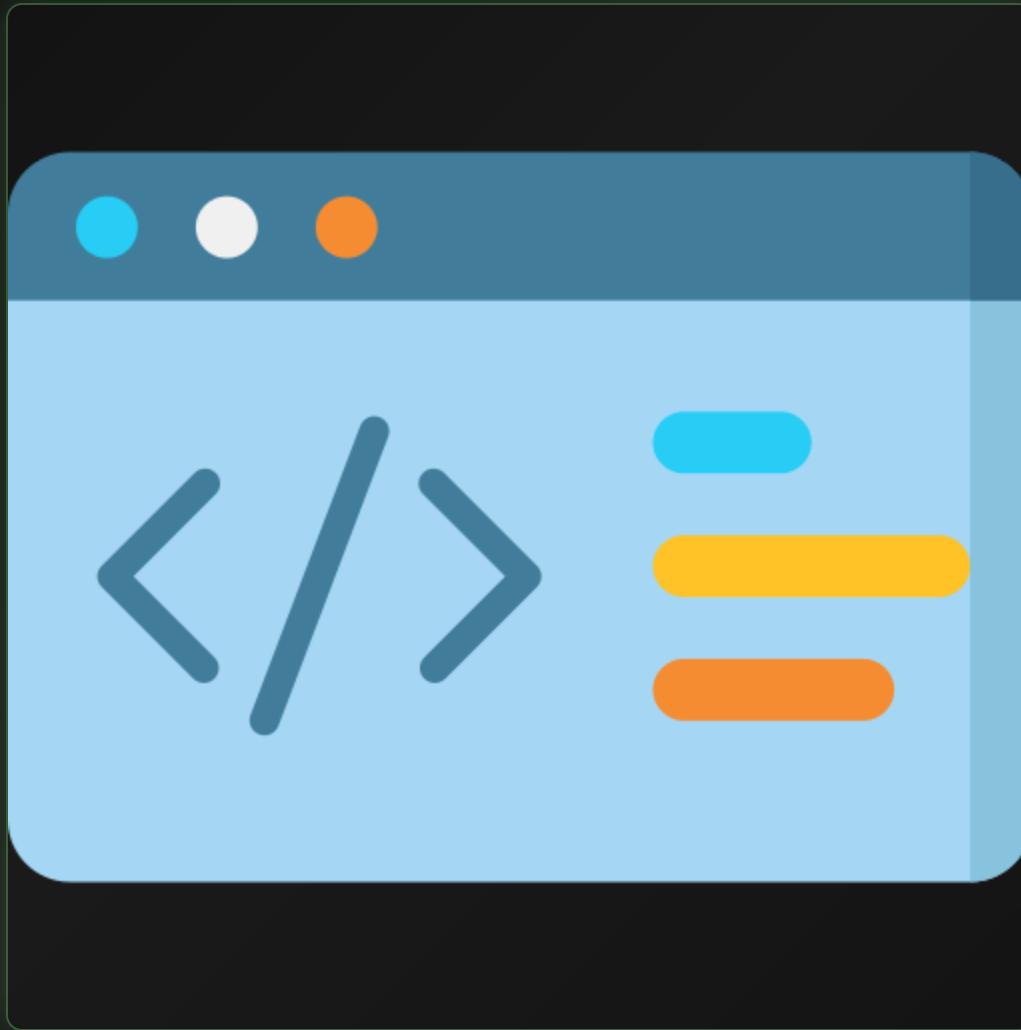
\$ MCP



## >> Model Context Protocol

- Connecte LLMs aux outils
- Extensibilité modulaire
- Standard de communication entre les IA agent

\$ MCP



# \$ MCP Client

- 5ire
- Cline
- ClaudeDesktop

# \$ Exemple MCP Client

```
{  
    "key": "Kali",  
    "description": "Kali",  
    "command": "python3",  
    "args": [  
        "/home/viper/Code/AI/self-hosted/kali/MCP-Kali-Server/mcp_server.py",  
        "--server",  
        "http://127.0.0.1:5000"  
    ]  
}
```

# \$ MCP Serveur Self-Hosted

- Kali-MCP
- Crawl4AI

# \$ MCP Serveur Online

- Context7 <https://context7.com/>
- Brave Search MCP

\$ Demo Crawl4AI

<http://127.0.0.1:8000/playground/>

# \$ Workflow Complet

## >> Pipeline N8N + IA

1. **Déclencheur** : Temps, Chat, Webhook
2. **Lecture** : Extraction contenu
3. **IA** : Analyse Ollama/Fabric
4. **Traitement** : Formatage
5. **Action** : Notification/sauvegarde
6. **Interaction API** :

# \$ Créer un Agent IA

- N8N (NoCode)
- PydanticAI (Code, plus simple)
- CrewAI (Multi Agent, plus complexe)

# \$ Pydantic AI

- Programmatiquement

\$ Demo Pydantic AI

# \$ CrewAI

- Programmatiquement Multi Agent qui peuvent communiquer entre eux.

\$ Demo CrewAI

```
/home/viper/Code/AI/self-hosted/agents/crew.py
```

## \$ Comment 'sécurisé' un agent IA.

- Vecteur d'attaque est trop grand.
- L'ajout de Guardrail
- Input et Output.

# \$ Attaque sur un AI

- Poison Data
- System Prompt Leak
- Token Attack
- Prompt Injection

<https://gandalf.lakera.ai/>

# \$ Ressources



# \$ Liens officiels

- **Fabric** : [github.com/danielmiessler/fabric](https://github.com/danielmiessler/fabric)
- **Ollama** : [ollama.com](https://ollama.com)
- **OpenWebUI** : [openwebui.com](https://openwebui.com)
- **N8N** : [n8n.io](https://n8n.io)
- **CrewAi** : [CrewAI](https://CrewAI)
- **PydanticAI** : [PydanticAI](https://PydanticAI)
- **Kali MCP** : [KaliMCP](https://KaliMCP)

# \$ Quel Modèle je prend?

- llama2 ou 3 (Exécution)
- qwen3 (Code)
- deepseek-r1 (Think)
- WhiteRabbitNeo (Unlocked + Trained on Security)

# \$ Lecon Aprise

- Context is Key
- Parler Lui

## \$ Prochaines étapes

- Plus de MCP Local Self\_hosted
- Trouver un bon MCP Client
- MCP pour naviguer internet
- AI Agent Contexte Note Obsidian

\$ Questions ?

>> Merci !