Génération automatisée d'architectures vulnérables variables

Soutenance de thèse

Pierre-Victor BESSON

Équipe PIRAT\'); CentraleSupelec, IRISA

22/11/2024

Encadré par:
Valérie Viet Triem Tong, Gilles Guette, Guillaume Piolle, Erwan Abgrall.
Thèse financée par la Direction Générale de l'Armement (CREACH LABS).

Table des matières

Introduction

- Croissance des **cyber-attaques** → Importance de la **cyber-défense**.
- Mesures préventives et curatives.

Prévention

Connaître son ennemi!

Pentesters

Objectif

Compromettre une organisation de façon éthique pour renseigner sur ses vulnérabilités.

- Comment entraîner des pentesters?
  - Besoin de déployer des terrains d'entraînement ...
  - **Architectures vulnérables**.

- Deux composantes :
  - **Environnement isolé** adapté à l'entrainement.
  - Génération et description de **scénarios d'attaque.**

Exemple de Cyber Range (DIATEAM).

Contributions de cette thèse

Première contribution

Nouveau **formalisme de scénario d'attaque** adapté à la spécification et au déploiement d'architectures vulnérables à ces scénarios.

- Descriptions **haut** et **bas-niveau** d'un scénario.
- Automatise le déploiement de variations d'une unique description afin d'en améliorer leur **réutilisation**.
- Publié à FPS 2023.

Deuxième contribution

Applications **scientifiques** et **éducatives** de l'outil issu de ce travail: URSID.

- Deux expériences :
  - **CERBERE**, publié à Cyberhunt 2023.
  - **Casinolimit**, dans le cadre de BreizhCTF 2024.

Table des matières

Table des matières

État de l'art : Cyber Ranges

Name	Date	Goal	Open source?	Scenario variability	Description level
CyRIS	2016	Educational	Yes	A lot Manual	Low
SecGen	2017	Educational	Yes	A lot Manual	Low/Mid
VulnerVAN	2019	Red team training	No	A lot Manual	Mid
SOCBED	2021	Log gathering	Yes	None -	-
Kyoushi	2021	Log gathering	Yes	Some Manual	Low
VSDL	2022	Versatile	No	Some Manual	Low
Yamin *et al*	2022	Versatile	No	A lot Manual	Low/mid
NASimEmu	2023	AI training	Yes	Some Manual	Mid

État de l'art : Cyber Ranges

```
– guest_settings:
  – id: desktop
    ip_addr: 192.168.122.50
    basevm_host: host_1
    basevm_config_file: /home/cyuser/images/basevm_desktop.xml
    basevm_type: kvm
    tasks:
    – add_account:
      – account: daniel
        passwd: danielpass
        full_name: Daniel Radcliffe
      – account: root
        new_passwd: abcd1234
    – install_package:
      – package_manager: yum
        name: wireshark
        version: 1.8.10
      – package_manager: yum
        name: GeoIP
      – package_manager: yum
        name: nmap
```

Fig: Exemple de description bas-niveau (CyRIS) - Configuration d'une machine à mot de passe faible.

- **Nos objectifs :**
  - Open Source.
  - Génération automatisée de scénarios variables.
  - Scénarios décrit à Haut/Bas niveau.

Comment décrire un scénario d'attaque?

- **Killchains** : Actions de l'attaquant découpées en phases.
  - Pensée pour la défense active.
  - Besoin de plus de spécificité.

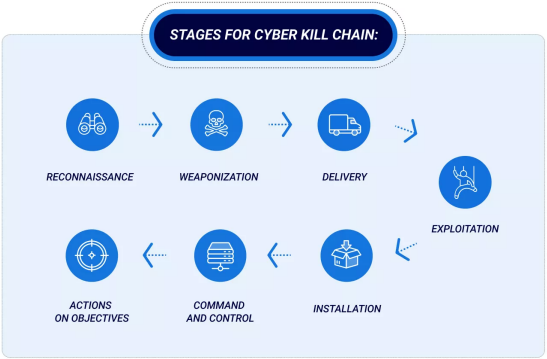


Fig: Cyber Kill Chain selon Lockheed Martin.

Comment décrire un scénario d'attaque?

- **Graphe d'attaque** : description d'une attaque à travers une architecture.
- Choix de représentations : un noeud peut être
  - Une représentation de l'état d'un attaquant [1].
  - Une représentation de vulnérabilités [2].
  - **Hôte-centré** : les noeuds du graphe sont des emplacements à l'intérieur du réseau [3].

[1] Phillips et al. , A graph-based system for network-vulnerability analysis, 1998.
[2] Ingols et al., Practical Attack Graph Generation for Network Defense, 2006.
[3] Ammann et al., A host-based approach to network attack chaining analysis, 2005.

Etat de l'art : descriptions de scénarios d'attaque.

- Mensah et al.[a], Berady et al.[b] : Les noeuds sont des **positions d'attaque** : (User, Machine).
- Les **transitions** sont des actions de l'attaquant.

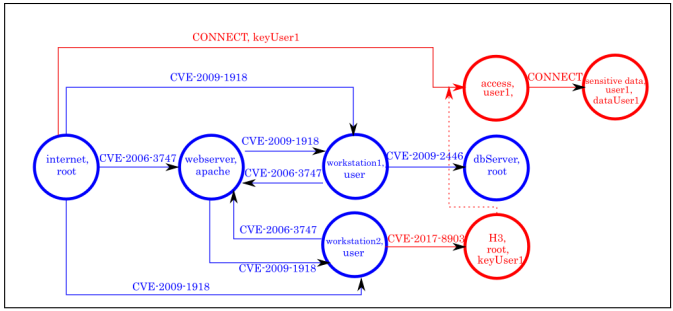
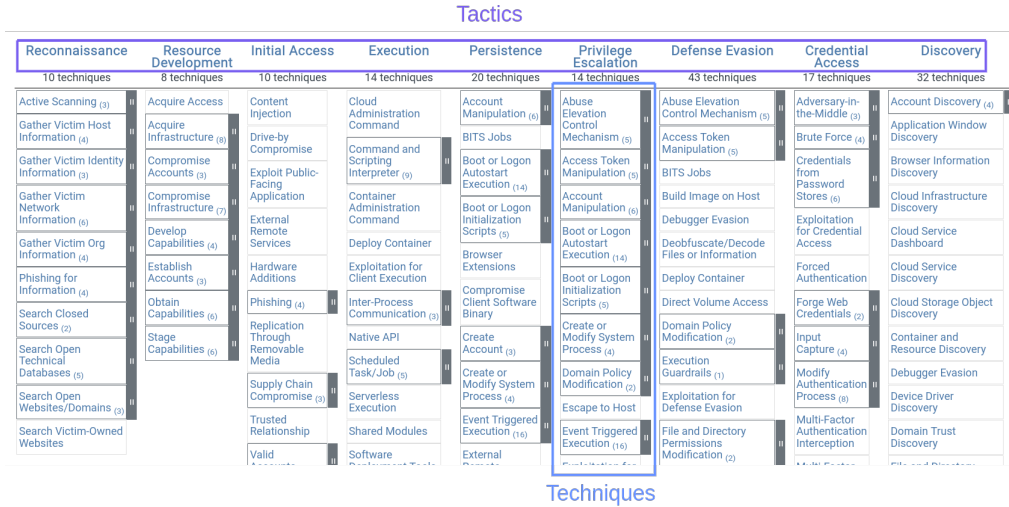


Fig : Graphe d'attaque selon Mensah et al.

[a] Connectivity graph reconstruction for networking cloud infrastructures (NCA, 2017)

[b] From TTP to IoC: Advanced Persistent Graphs for Threat Hunting (TNSM, 2021)

Vers un entre-deux? MITRE ATT&CK



Vers un entre-deux? MITRE ATT&CK

Exploitation for Privilege Escalation

Technique

Adversaries may exploit software vulnerabilities in an attempt to elevate privileges. Exploitation of a software vulnerability occurs when an adversary takes advantage of a programming error in a program, service, or within the operating system software or kernel itself to execute adversary-controlled code. Security constructs such as permission levels will often hinder access to information and use of certain techniques, so adversaries will likely need to perform privilege escalation to include use of software exploitation to circumvent those restrictions.

When initially gaining access to a system, an adversary may be operating within a lower privileged process which will prevent them from accessing certain resources on the system. Vulnerabilities may exist, usually in operating system components and software commonly running at higher permissions, that can be exploited to gain higher levels of access on the system. This could enable someone to move from unprivileged or user level permissions to SYSTEM or root permissions depending on the component that is vulnerable. This could also enable an adversary to move from a virtualized environment, such as within a virtual machine or container, onto the underlying host. This may be a necessary step for an adversary compromising an endpoint system that has been properly configured and limits other privilege escalation methods.

Adversaries may bring a signed vulnerable driver onto a compromised machine so that they can exploit the vulnerability to execute code in kernel mode. This process is sometimes referred to as Bring Your Own Vulnerable Driver (BYOVD).[1][2] Adversaries may include the vulnerable driver with files delivered during Initial Access or download it to a compromised system via Ingress Tool Transfer or Lateral Tool Transfer.

ID: T1068
Sub-techniques: No sub-techniques
Tactic: Privilege Escalation
Tactic
Platforms: Containers, Linux, Windows, macOS
Permissions Required: User
Effective Permissions: User
Contributors: David Tayouri; Idan Revivo, @idanr86, Team Nautilus Aqua Security; Joas Antonio dos Santos, @C0d3Cr4zy, Inmetrics; Yaniv Agman, @AgmanYaniv, Team Nautilus Aqua Security
Version: 1.5
Created: 31 May 2017
Last Modified: 07 April 2023

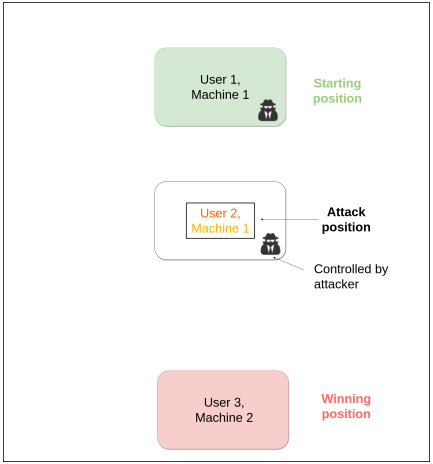
Version Permalink

Procedures

Procedure Examples

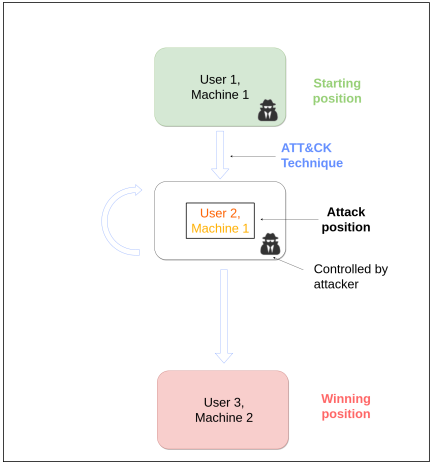
ID	Name	Description
G0007	APT28	APT28 has exploited CVE-2014-4076, CVE-2015-2387, CVE-2015-1701, CVE-2017-0263 to escalate privileges.[3][4][5]
G0016	APT29	APT29 has exploited CVE-2021-36934 to escalate privileges on a compromised host.[6]
G0050	APT32	APT32 has used CVE-2016-7255 to escalate privileges.[7]
G0064	APT33	APT33 has used a publicly available exploit for CVE-2017-0213 to escalate privileges on a local system.[8]

Description de scénario d'attaque (niveau technique)



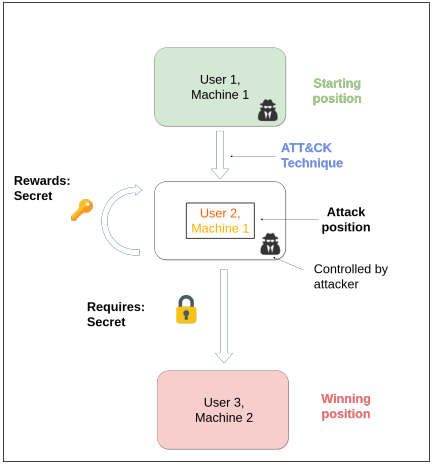
- **Position d'attaque** : Couple (User , Machine).

Description de scénario d'attaque (niveau technique)



- **Position d'attaque** : Couple (User , Machine).
- **Transitions** : ATT&CK techniques.

Description de scénario d'attaque (niveau technique)



- **Positions d'attaque** : Couples (User , Machine).
- **Transitions** : ATT&CK techniques.
- **Secrets** : certaines transitions requièrent ou donnent des informations (mots de passe, clefs...).

Exemple de formalisation

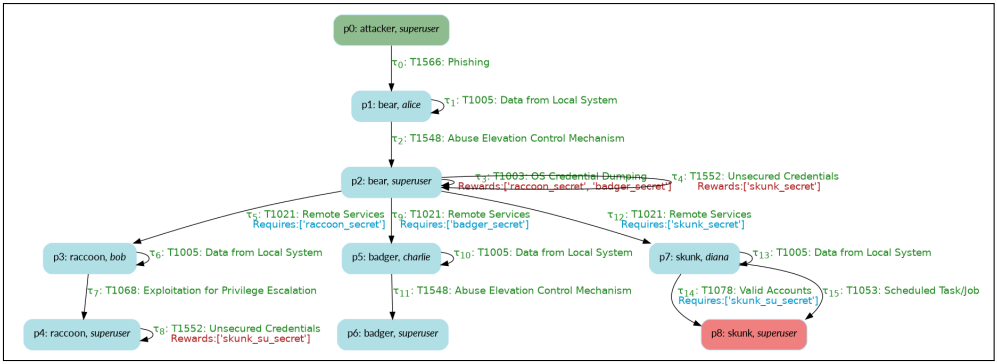
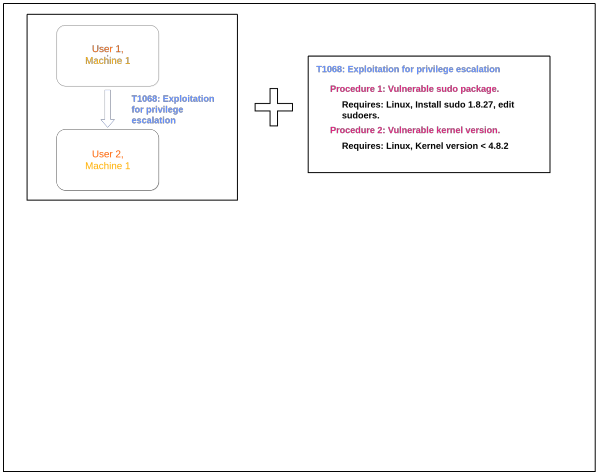


Fig : Formalisation de scénario d'attaque au niveau technique (inspiré de APT3), généré par URSID.

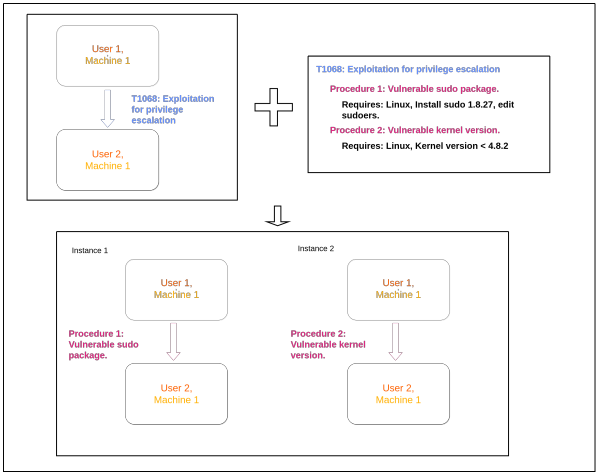
Table des matières

Une description au niveau technique



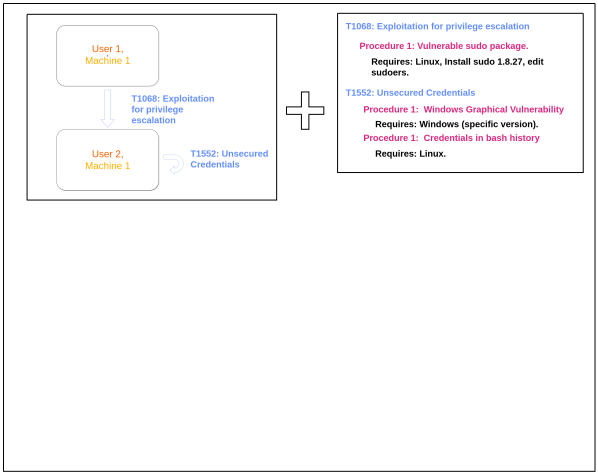
- Chaque technique est associée avec une ou plusieurs procédures.
- Certaines procédures requièrent des configurations spécifiques.

Plusieurs descriptions au niveau procédure

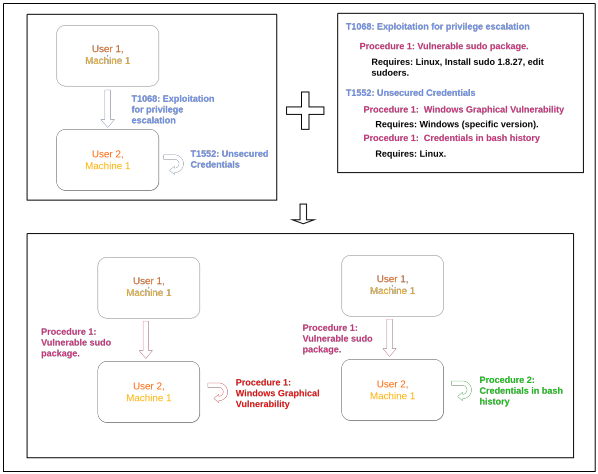


- Chaque technique est associée avec une ou plusieurs procédures.
- Certaines procédures requierent des configurations spécifiques.

Contraintes d'architecture



Contraintes d'architecture



Choix des procédures

- Attention, procédures = configurations particulières $\rightarrow$ potentielles **incompatibilités**.

Contraintes d'architecture et raffinement.

- Problème de **résolution de contraintes**.
- Pour chaque technique, **raffinement procédural** :
  - Choix d'une procédure.
  - Comparaison de ses contraintes d'architecture avec celles choisies précédemment : **OS**, **Software**, **Files**, **Account**.
  - Si incompatibilité : choix d'une autre procédure (*backtracking*).

Résultat

Cohérence du scénario

- Scénario **cohérent** : pas d'incompabitibilités procédurales.
- Garanti par le raffinement procédural.

Jouabilité du scénario

- Scénario **gagnable** : l'attaquant est garanti de pouvoir accéder à son objectif.
- Garanti par la représentation au niveau technique du scénario.

- Contient assez d'information sur leur configuration pour être **déployable**.

Déploiement

- Conversion en fichiers **Vagrant** (descriptions du type de machines à déployer) et **Ansible** (descriptions du contenu à installer sur les machines).
- En sortie : Réseau virtuel configuré pour être **vulnérable** à toutes les procédures choisies.

Correspondance

- Tous les scénarios en sortie correspondent à des variations procédurales du scénario d'entrée.
- Même scénario, mêmes **techniques**, différentes **procédures**.

URSID

- Disponible en ligne sous license GPL 3[a].
- Utilisé pour 2 experiences durant cette thèse : CERBERE et Casinolimit.
- 9 techniques, 20 procédures.

[a] https://gitlab.inria.fr/pirat/ursid

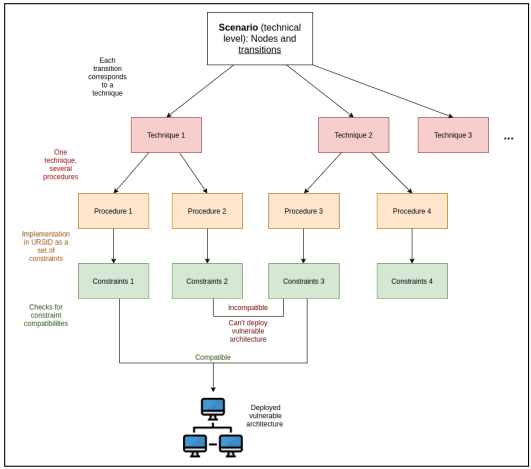


Fig : Fonctionnement d'URSID.

Table des matières

CERBERE

- Exercice d'initiation au pentesting destiné à des étudiants (EUR Cyber School).

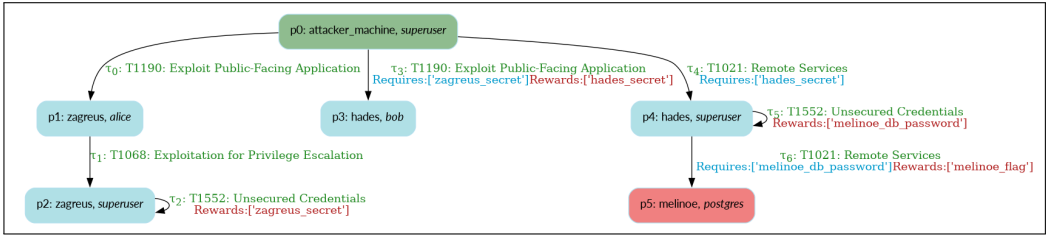
Objectifs

- **Familiarisation** d'étudiants avec les métiers liés au pentesting (red team et blue team).
- Récupération de **jeux de données** d'attaquants.
- Test concret d'URSID.

Apports de l'outil

- **Reproductibilité**: de l'experience, du jeu de donnée.
- **Variabilité**: richesse des données, interêt pédagogique.

CERBERE : Scenario au niveau technique



CERBERE scenario au niveau technique.

- 3 machines, 6 positions d'attaque, 7 transitions au total + chemin gagnant.
- 4 techniques, 9 procedures.
- 20 variations générées.

Résultat

- 13 étudiants, 4 instances ont été intégralement compromises.
- 900 Mo de logs systèmes + réseau.
- **Publication** à CyberHunt 2023 : expérience reproductible + dataset.

Bilan

- Travail collaboratif facilité par URSID.
- Questions sur la difficulté et le design du scénario.
- Labélisation semi-automatisée des données et chemins d'attaque.

Analyse automatisée des résultats

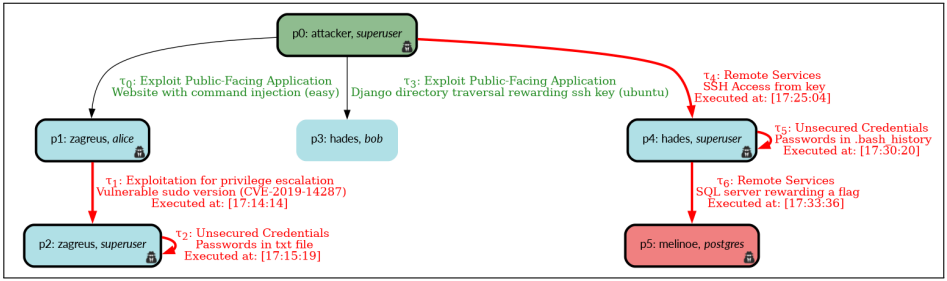


Fig: Trajet d'attaque pour l'instance 7.

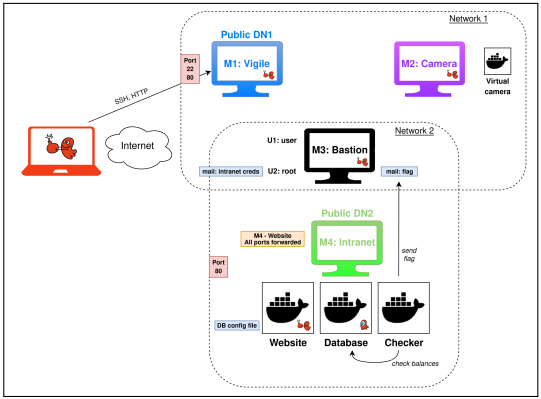
Casinolimit

- Stand au **BreizhCTF**, une compétition de type Capture-The-Flag.
- 600 participants répartis en 120 équipes.

Objectifs scientifiques

- Passage à l'échelle.
- Données d'attaquants chevronnés.
- Données sur un scénario plus complet.

Casinolimit



- Contrainte de l'événement: pas de variation.
- 5 positions d'attaque, 8 transitions.

Résultats

BreizhCTF!

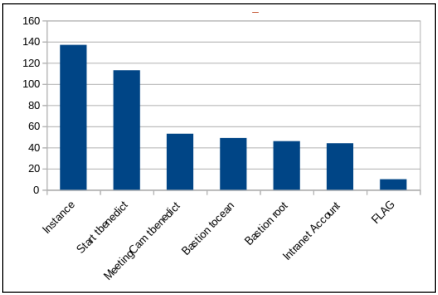


Fig : Nombre d'équipes par étape du scénario.

- 38 Go de logs sytème, 353 Go de logs réseau.
- Données labellisées disponibles publiquement dans le futur.

Bilan d'experiences

- Passages à l'échelle réussis.
- Jeux de données à fort interêt pour la recherche.
- Interêts de la variabilité, mais optionnelle.
- **Modularité** : une fois une procédure ajoutée il est facile de la réutiliser.

Pistes d'amélioration

- Découpage en sous-réseaux.
- Automatisation de pipelines de déploiement.
- Design du scenario (difficulté).

Table des matières

Progression de l'état de l'art

Name	Date	Goal	Open source?	Scenario variability	Description level
SecGen	2017	Educational	Yes	A lot Manual	Low/Mid
CyRIS	2016	Educational	Yes	A lot Manual	Low
SOCBED	2021	Log gathering	Yes	None -	-
Kyoushi	2021	Log gathering	Yes	Some Manual	Low
NASimEmu	2023	AI training	Yes	Some Manual	Mid
VSDL	2022	Versatile	No	Some Manual	Low
Yamin *et al*	2022	Versatile	No	A lot Manual	Low/mid
VulnerVAN	2019	Red team training	No	A lot Manual	Mid
URSID	2024	Education Log gathering	Yes	A lot Automated	Low/High

Conclusion

Première contribution

- Nouveau **formalisme de scénario d'attaque** adapté au déploiement de cyber range → publié et présenté à FPS 2023.

Deuxième contribution

- Deux expériences à objectifs scientifiques et éducatifs.
- CERBERE (publié avec jeu de données à Cyberhunt 2023) et Casinolimit (jeu de données en cours de labelisation).

URSID

- Utilisations futures pour les travaux de PIRAT\'); .
- Présenté à plusieurs acteurs de la cyber-défense (Orange, Silicom...).

Travaux futurs

Questions d'ingénierie

- Facilitation de l'ajout de procédures.
- Intégration de pipelines de déploiement (Cloud, Airbus Cyber Range...)
- Enrichissement de la bibliothèque de procédures.

Questions scientifiques

- Évaluation de la difficulté de scenarios.
- Automatisation de la labellisation de datasets.
- Applications pots de miel $\rightarrow$ analyse d'attaquants réels.

Démonstration de l'outil

Raffinement procédural : Architectural constraints

- 4 types de **constraintes**, chacune avec ses sous-contraintes :
  - **OS**: Type (Windows, Ubuntu...) et Version.
  - **Software**: Type, Version, Port (optionel).
  - **Account**: Name, Group, Privileges, Services, Credentials.
  - **Files**: Path, Permissions, Content.

Procedural refinement : Architectural constraints.

- 4 types de **constraintes**, chacune avec ses sous-contraintes :
  - **OS**: Type (Windows, Ubuntu...) et Version.
  - **Software**: Type, Version, Port (optionel).
  - **Account**: Name, Group, Privileges, Services, Credentials.
  - **Files**: Path, Permissions, Content.
- Les contraintes sont **combinées** entre elles pour vérifier d'eventuelles incompatibilitées :
  - Par exemple ("Linux", "Any version") x ("Ubuntu", "=16.04") renvoie (Ubuntu, "16.04").
  - Mais ("Sudo", ">1.9") x ("Sudo", "=1.8.2") renvoie une incompatibilitée.

Casinolimit : Scénario d'attaque

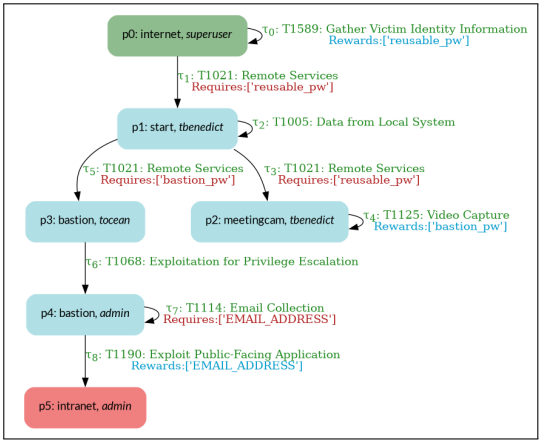


Fig : Scénario d'attaque au niveau technique.