RS-SHELL

A REVERSE SHELL WITH COOL FEATURES IN RUST

WHOAMI /ALL

- Guillaume DAUMAS - @BlWasp_
- Auditeur chez Advens 
  - TI interne
  - Red Team
- Toulouse
- Projets parallèles
  - Auteur de cheatsheets AD / AD -Python / ADCS / SCCM / Pivoting 
  - Participant au projet The Hacker Recipes de Shutdown (@_nwodtuhs) 

  - GitHub : BlackWasp, quelques projets et PRs (Impacket, CME / NXC)

EN 2 MOTS

- Projet que je développe sur mon temps libre
- Il s'agit initialement d'un reverse shell en Rust...
  - TLS over TCP, avec un implant et un listener
  - Fonctionnalités d'upload et download
  - Tout embarqué dans un unique binaire
- ...qui a quelque peu évolué
  - HTTPS
  - Proxy aware
  - Injections de PE et shellcodes
  - Bypass AMSI
  - Win32API ou indirect syscalls

MOTIVATION INITIALE

- Mission Red Team à préparer
  - Pas mal de C2 à tester
  - Besoin de développer des packers
  - Pas vraiment d'outils d'initial access publics satisfaisants

- Alors autant développer son propre outillage
  - Choix du Rust car récent et safe, poussé par Microsoft, et LLVM par défaut
  - Création d'une sorte de sandbox pour tester…des trucs ?
    - Dev de premiers packers
    - Dev d'un premier reverse shell TCP ultra basic
    - …

VERSION 1

- **Un reverse shell TLS over TCP stable et performant**
  - Chiffrement des données avec ‘native-tls’
  - Gestion des erreurs et de la mémoire propre grâce au Rust
  - Serveur et implant embarqués dans le même binaire
  - Implants et serveur cross-plateformes

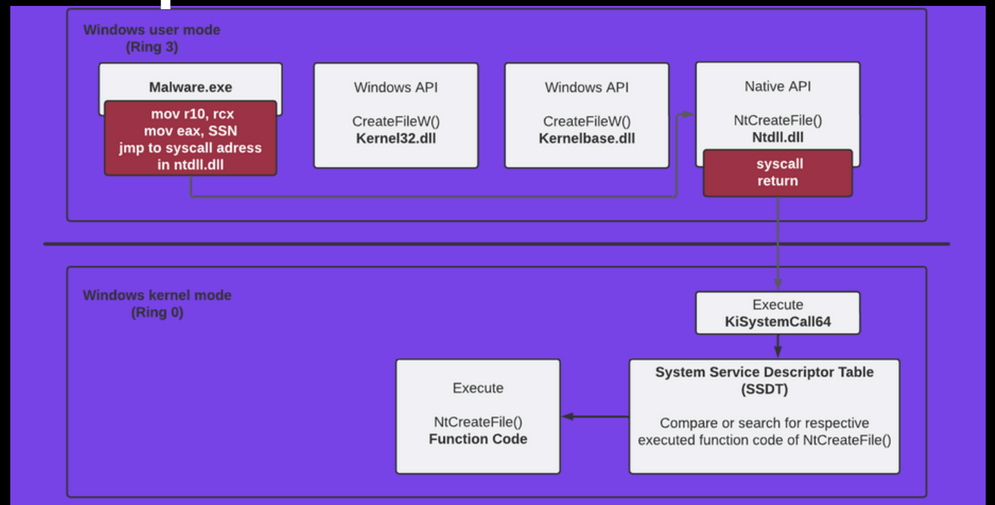
- **Fonctions avancées pour l’implant Windows**
  - Injection locale de PE et Process Hollowing
  - Injection d’un shellcode dans un processus distant sans l’écrire sur le disque
  - Ouverture d’une session PowerShell interactive avec bypass AMSI en mémoire

- **Upload / download / et d’autres features sympa**

VERSION 1.5 - UNLEASHED SYSCALLS

- **Toutes les fonctionnalités avancées Windows peuvent maintenant être réalisées via indirect syscalls**
  - **Toujours possible d'appeler les versions Win32API**

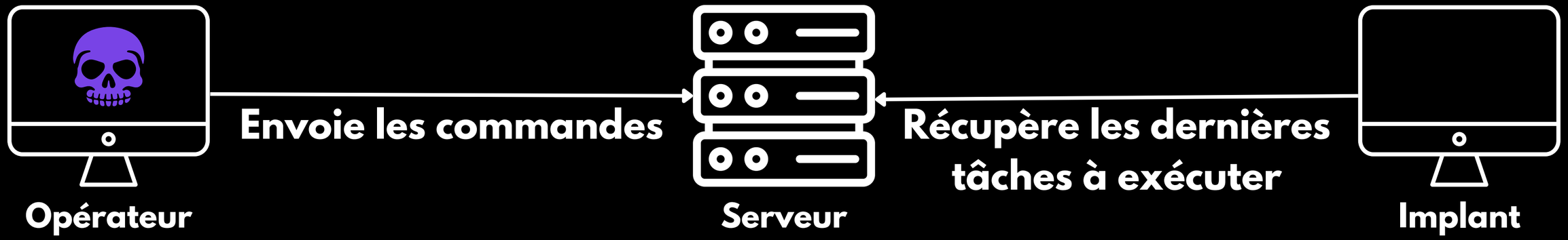
- **Utilisation d'une version personnalisée du projet 'mordor-rs' de memN0ps**
  - **Projet permettant d'appeler les fonctions de la NTAPI via direct/indirect syscalls**
  - **Récupération dynamique des adresses de fonctions et SSN**



VERSION 2 - HTTPS & PROXY AWARE

- Retour à la réalité : les entreprises ont des proxies sortants…

- Nouveau mode ✨HTTPS✨
  - Basé sur Actix et Rustls
  - Infrastructure avec implant/serveur/opérateur
  - L'implant Windows est proxy aware grâce à WinINet
    - Récupération de la configuration du proxy dans le registre
    - Gestion de l'authentification via le Credentials Manager de Windows



DEMO TIME

CE QU'IL RESTE À FAIRE

- Rendre le mode HTTPS plus “propre”
  - Gestion des accès, que ce soit pour les opérateurs ou les implants
  - Gestion des commandes
- Reverse socks proxy
- Port forwarding
- Un petit ‘getsystem()’ pour la route
- D’autres techniques d’injection ?

L’OBJECTIF N’EST PAS QUE ÇA DEVIENNE UN C2 : NI LES COMPÉTENCES, NI LE TEMPS

LES LIENS

- RS-Shell : https://github.com/BlWasp/rs-shell
- syscalls-rs (ex mordor-rs) : https://github.com/BlWasp/syscalls-rs