



# Conoce a tu enemigo y conóciate a ti mismo

Antonio López - Kevin Gonzalvo





# Antonio López

Experiencia en tests de intrusión, ejercicios de Red Team y revisiones de seguridad en sistemas/aplicaciones

- Red Team Operator en **Innotec Security**



@behindthebreach



<https://github.com/t0-n1>



<https://bit.ly/in-alopez>





# Kevin Gonzalvo

Un friki de la informática en general, y de la seguridad en particular

- IT Security Analyst en **Ackcent Cybersecurity**



@interh4ck



<https://github.com/interhack86>



<https://bit.ly/in-kgonzalvo>





# Índice

1. Presentación
2. Motivación
3. Solución propuesta
4. Objetivos
5. Colección de archivos de consulta
6. Herramienta
7. Demos
8. Conclusiones

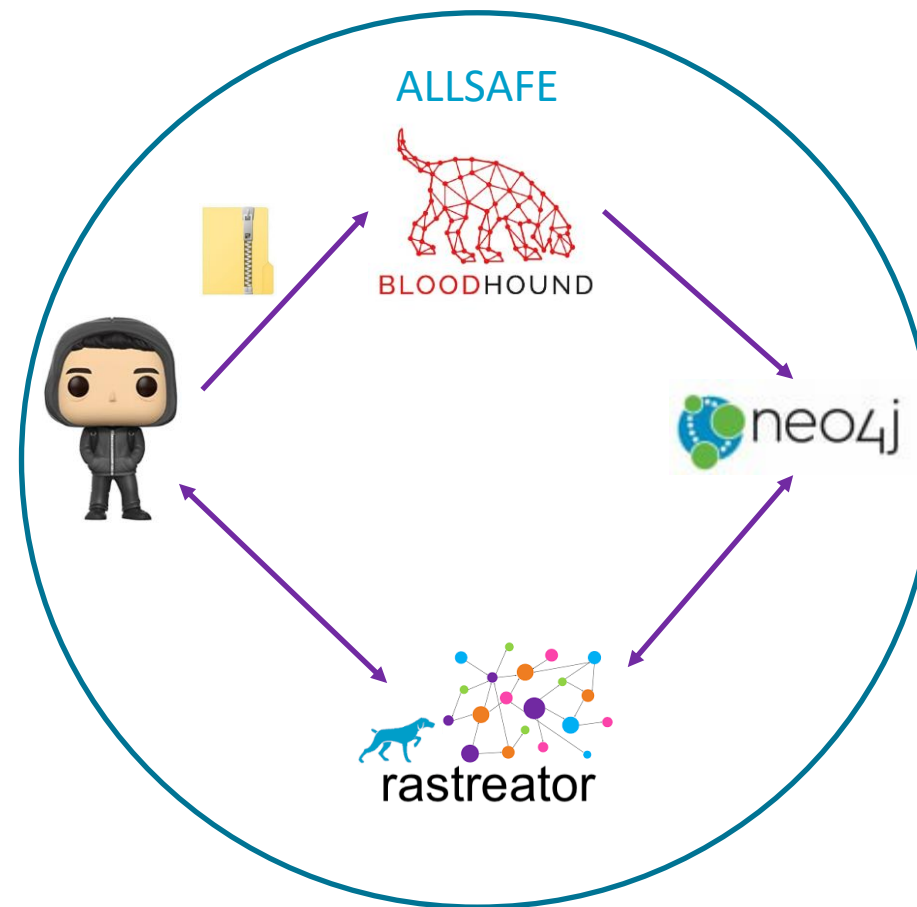
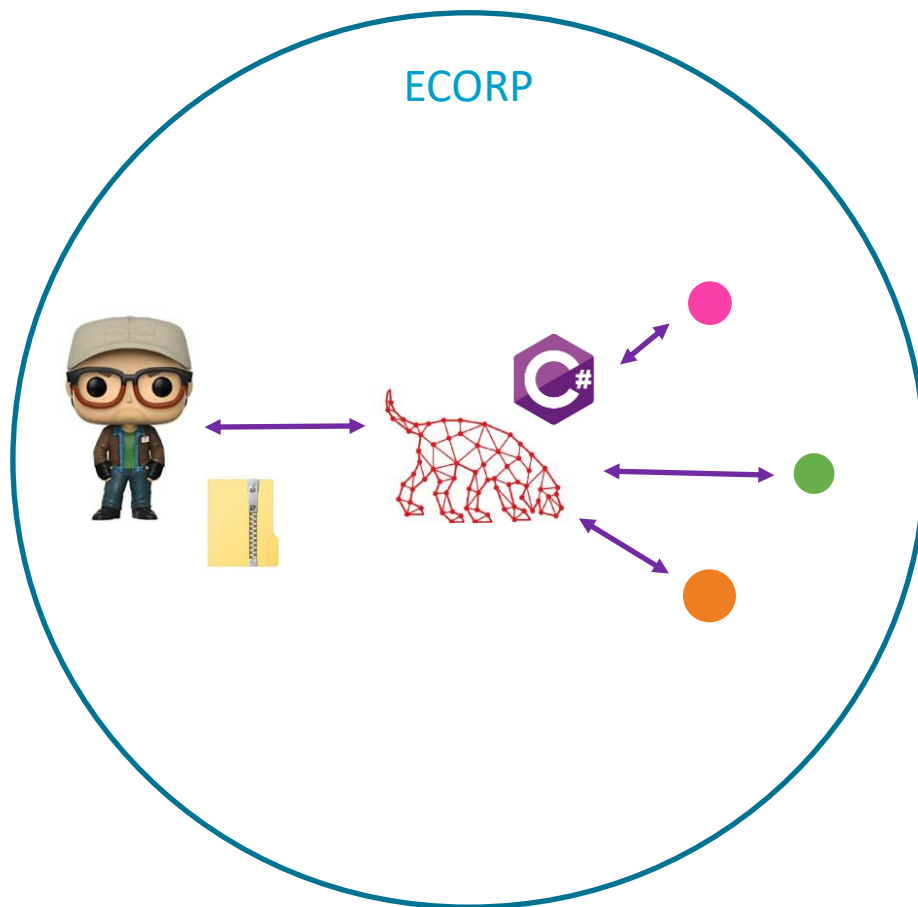


# Presentación

- Es una herramienta desarrollada en Python (rastreator.py) que
  - obtiene información
  - propone ataques potenciales
  - descubre problemas de configuración en el Directorio Activo de Microsoft
- Colección de archivos de consulta
- Acepta archivos de consulta con sentencias Cypher
- Proporciona resultados en varios formatos
- Dependencias
  - SharpHound / BloodHound: Recolección e inserción de información
  - Neo4j: Base de datos orientada a grafos



# Presentación





# Motivación

- BloodHound es un gran herramienta de exploración visual de datos pero desde nuestro punto de vista
- Acabamos
  - Desarrollando sentencias propias y probándolas en Neo4j
  - Recopilando sentencias de internet y adaptándolas a otros idiomas
- Resulta tedioso
  - Tener que ejecutar manualmente una sentencia detrás de la otra
  - Exportar los resultados obtenidos
  - Mantener un único archivo con todas las sentencias Cypher





# Solución propuesta

## Características interesantes

- Modos de operación
  - interactive
  - check
  - command
  - execute
- Diferentes formatos de resultados: CSV, JSON, YAML
- Colección de consultas para: Red/Blue Teams, Pentester y Auditores
- Archivos de consulta con metadatos: tactic, tag, nextsteps, references





# Solución propuesta

## Características interesantes

- Soporte multilenguaje y multidominio en consultas Cypher

`rastreator / conf / languages.yaml`

```
1  en:
2    RAS-A: ADMINISTRATORS
3    RAS-DA: DOMAIN ADMINS
4    RAS-DC: DOMAIN CONTROLLERS
5    RAS-EA: ENTERPRISE ADMINS
6  es:
7    RAS-A: ADMINISTRADORES
8    RAS-DA: ADMINS. DEL DOMINIO
9    RAS-DC: CONTROLADORES DE DOMINIO
10   RAS-EA: ADMINISTRADORES DE ORGANIZACIÓN
```



# Objetivos

- Mejorar la colección de archivos de consulta
  - Compartir y centralizar
  - Investigar y crear nuevas sentencias Cypher
  - Recopilar sentencias Cypher
  - Promover la colaboración de la comunidad
- A corto plazo
  - Añadir consultas para Azure
  - Mejorar el filtrado de las consultas
- A medio plazo
  - Eliminar dependencia de BloodHound
  - Ingestor propio










# Colección de archivos de consulta

- Es la parte más importante del proyecto
- Clasificada en directorios usando las tácticas de Mitre ATT&CK
  - Selección de directorio con las consultas a ejecutar
- Animamos a la comunidad a participar

WE WANT YOU!



[rastreator](#) / [queries](#) /

	collection
	credential_access
	discovery
	execution
	lateral_movement
	persistence
	privilege_escalation



# Colección de archivos de consulta

- Un archivo de consulta puede ser de tipo
  - raw, test o default
- Los campos de un archivo default son

## Obligatorios

author  
name  
state  
tactic  
tag  
description  
statement

## Opcionales

reference  
nextsteps

`rastreator / queries / credential_access / as-rep_roasting.yaml`

```
1  author: rastreator
2  name: as-rep_roasting
3  state: enabled
4  tactic: credential access
5  tag: attack
6  description: Get users with dontreqpreauth (AS-REP roasting).
7  statement:
8    table: >-
9      match (u:User{enabled:true, dontreqpreauth:true})
10     return u.name
11     order by u.name asc
12  reference:
13    - https://www.harmj0y.net/blog/activedirectory/roasting-as-reps/
14  nextsteps:
15    rt:
16      - Request TGT for all user without preauthentication and crack them.
17    bt:
18      - Ensure all service accounts have a long, complex passwords.
19      - Use a strong password policy.
20      - Remove RC4 encryption via group policy.
21      - Create a service account honeypot.
```



# Herramienta

- Diferentes modos de operación (check, command, execute, interactive)

```
RastreatorTeam@localhost$ python3 rastreator.py -h
usage: rastreator.py [-h] {check,command,execute,interactive} ...

Rastreator
> Tool with a collection of query files to explore Microsoft Active Directory
> Developed by @interh4ck and @t0-n1

positional arguments:
  {check,command,execute,interactive}
    check                Check mode
    command              Command mode
    execute              Execute mode
    interactive          Interactive mode

optional arguments:
  -h, --help            show this help message and exit
```



# Herramienta

python3 rastreator.py **check** -h

```
-v {quiet,default,debug}
    Verbose mode
-I INPUT_DIRECTORY_OR_FILE
    Input directory or specific query file
-O OUTPUT_DIRECTORY
    Output directory to save results
-o {none,yaml}
    File format to save executed query results
```



# Herramienta

python3 rastreator.py **command** -h

```
-v {quiet,default,debug}      Verbose mode
-H NEO4J_HOST                 Neo4j host to connect
-P NEO4J_PORT                 Neo4j port to connect
-u NEO4J_USERNAME             Neo4j username
-p NEO4J_PASSWORD             Neo4j password
-e {off,on}                   Neo4j encrypted communication
-c COMMAND                    Semicolon separated shell commands inside single/double quotes
```



# Herramienta

python3 rastreator.py **execute** -h

```
-v {quiet,default,debug}
                        Verbose mode
-H NEO4J_HOST           Neo4j host to connect
-P NEO4J_PORT           Neo4j port to connect
-u NEO4J_USERNAME       Neo4j username
-p NEO4J_PASSWORD       Neo4j password
-e {off,on}             Neo4j encrypted communication
-I INPUT_DIRECTORY_OR_FILE
                        Input directory or specific query file
-O OUTPUT_DIRECTORY     Output directory to save results
-o {csv,json,none,yaml}
                        File format to save executed query results
-m {raw,test,default}
                        Execute submode
-f {csv,json,table,yaml}
                        Output format to show executed query results on screen
-l {en,es}             Active Directory language
-d AD_DOMAIN            Active Directory domain name
```





# Herramienta

python3 rastreator.py **interactive** -h

```
-v {quiet,default,debug}      Verbose mode
-H NEO4J_HOST                  Neo4j host to connect
-P NEO4J_PORT                  Neo4j port to connect
-u NEO4J_USERNAME              Neo4j username
-p NEO4J_PASSWORD              Neo4j password
-e {off,on}                    Neo4j encrypted communication
```



# Demos (mi primer archivo de consulta)

```
RastreatorTeam@localhost$
```

I



# Demos (bash-fu con rastreator)

```
RastreatorTeam@localhost$  
  
1
```



# Conclusiones

- Facilita la automatización y el descubrimiento de información
- Propone ataques y descubre vulnerabilidades
- Proporciona una colección de consultas para la comunidad
- Permite aprender y mantenerte al día acerca de la seguridad en los servicios del directorio activo de Microsoft



# Conclusiones

- Puede utilizarse para el

## Ataque

Conoce a tu enemigo (**puntos vulnerables**) y conócete a ti mismo (**vectores de ataque**)

## Defensa

Conoce a tu enemigo (**vectores de ataque**) y conócete a ti mismo (**puntos vulnerables**)

saldrás victorioso en 100 batallas

(Sun Tzu - El arte de la guerra)



# Preguntas

- ¿ Dónde conseguir el material de la presentación ?
  - <https://github.com/RastreatorTeam/presentations/2020/c0r0n4con>
- ¿ Dónde conseguir el código del proyecto ?
  - <https://github.com/RastreatorTeam/rastreator>
- ¿ Más preguntas ?