SCMKit: Source Code Management Attack Toolkit

Brett Hawkins (@h4wkst3r)
Adversary Simulation, IBM X-Force Red

X-Force Red

IBM

Agenda

- Introduction
- Background
- SCMKit
- Demos
- Defensive Considerations

Introduction

Who am I?

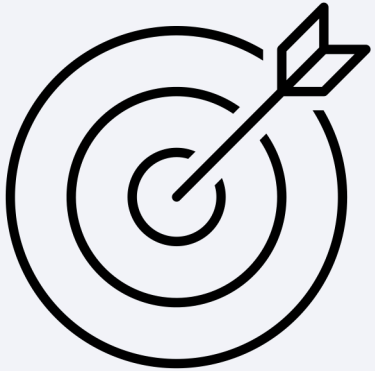
- **Current Role** – Adversary Simulation, IBM X-Force Red
- **Previous Roles** – Mandiant, J.P. Morgan Chase, J.M. Smucker Company
- **Conference Speaker** – DerbyCon, Wild West Hackin’ Fest, BSides, Hackers Teaching Hackers
- **Open-Source Tool Author** – SharPersist, DueDLLigence, InvisibilityCloak, SCMKit

How did this tool come about?

- Needed tooling to attack source code management systems hosted internally
- Ability to abuse stolen/discovered API keys in environments

Goals of SCMKit

- Abuse functionality of multiple popular SCM systems
- Provide tool that can be used in-memory or on-disk
- Modular approach



Background

What is a Source Code Management System?

- Manages source code repositories
- Allows multiple developers to work on code at same time
- Supports integrations into other systems within DevOps pipeline

Popular Systems

- GitHub Enterprise

- GitLab Enterprise

- Bitbucket

REST API Functionality

- SCM systems have REST API that can be interacted with

- Includes functionality for:
  - Repositories
  - SSH Keys
  - Users
  - Admin functionality
  - And much more…

SCMKit

Background

Source **C**ode **M**anagement Attack Tool**kit** written in C#

- https://github.com/xforcered/SCMKit

Supported SCM Systems:

- GitHub Enterprise, GitLab Enterprise, Bitbucket Server

Modules include:

- Reconnaissance, Privilege Escalation, Persistence

Arguments/Options

Required

-c, -credential – credential for auth (username:password or API key)

-s, -system – system to attack (github, gitlab, bitbucket)

-u, -url – URL of system to attack

-m, -module – module to run

Optional

-o, -option – options (when applicable)

Supported Systems

- **github** – GitHub Enterprise
- **gitlab** – GitLab Enterprise
- **bitbucket** – Bitbucket Server

Reconnaissance Modules

- **listrepo** – List all repos current user can see
- **searchrepo** – Search for a given repo
- **searchcode** – Search for code containing keyword search term
- **searchfile** – Search for filename containing keyword search term
- **listsnippet** – List all snippets of current user
- **listrunner** – List all GitLab runners available to current user
- **listgist** – List all gists of current user
- **listorg** – List all orgs current user belongs to
- **privs** – Get privs of current API token
- **adminstats** – Get admin stats (users, repos, orgs, gists)

Privilege Escalation Modules

- **addadmin** – Promote given user to admin role

- **removeadmin** – Demote given user from admin role

Persistence Modules

- **createpat** – Create personal access token for target user
- **listpat** – List personal access tokens for target user
- **removepat** – Remove personal access token for target user
- **createsshkey** – Create SSH key for current user
- **listsshkey** – List SSH keys for current user
- **removesshkey** – Remove SSH key for current user

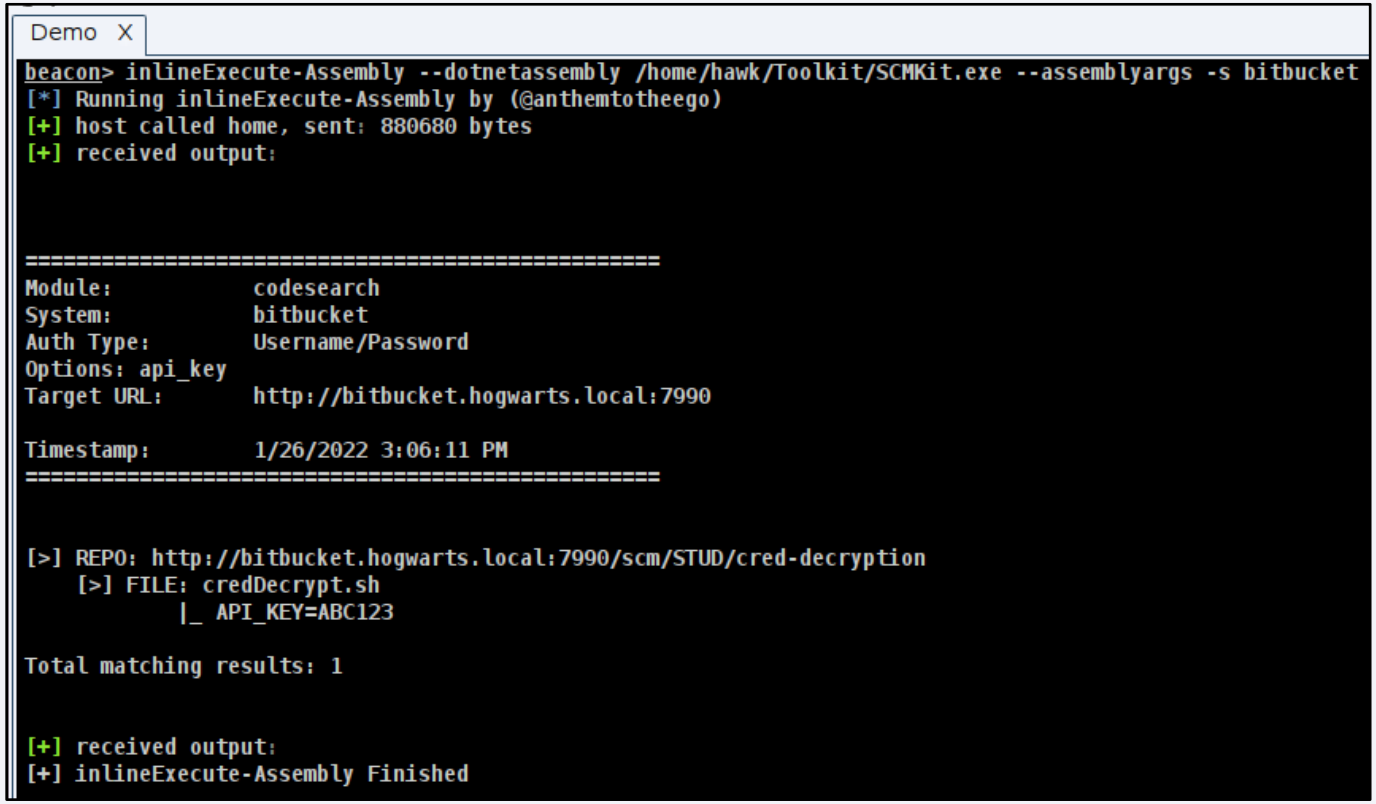
Module Details

Attack Scenario	Module	Requires Admin?	GitHub Enterprise	GitLab Enterprise	Bitbucket Server
Recon	listrepo	No	Yes	Yes	Yes
Recon	searchrepo	No	Yes	Yes	Yes
Recon	searchcode	No	Yes	Yes	Yes
Recon	searchfile	No	Yes	Yes	Yes
Recon	listsnippet	No	No	Yes	No
Recon	listrunner	No	No	Yes	No
Recon	listgist	No	Yes	No	No
Recon	listorg	No	Yes	No	No
Recon	privs	No	Yes	Yes	No
Recon	adminstats	Yes	Yes	No	No

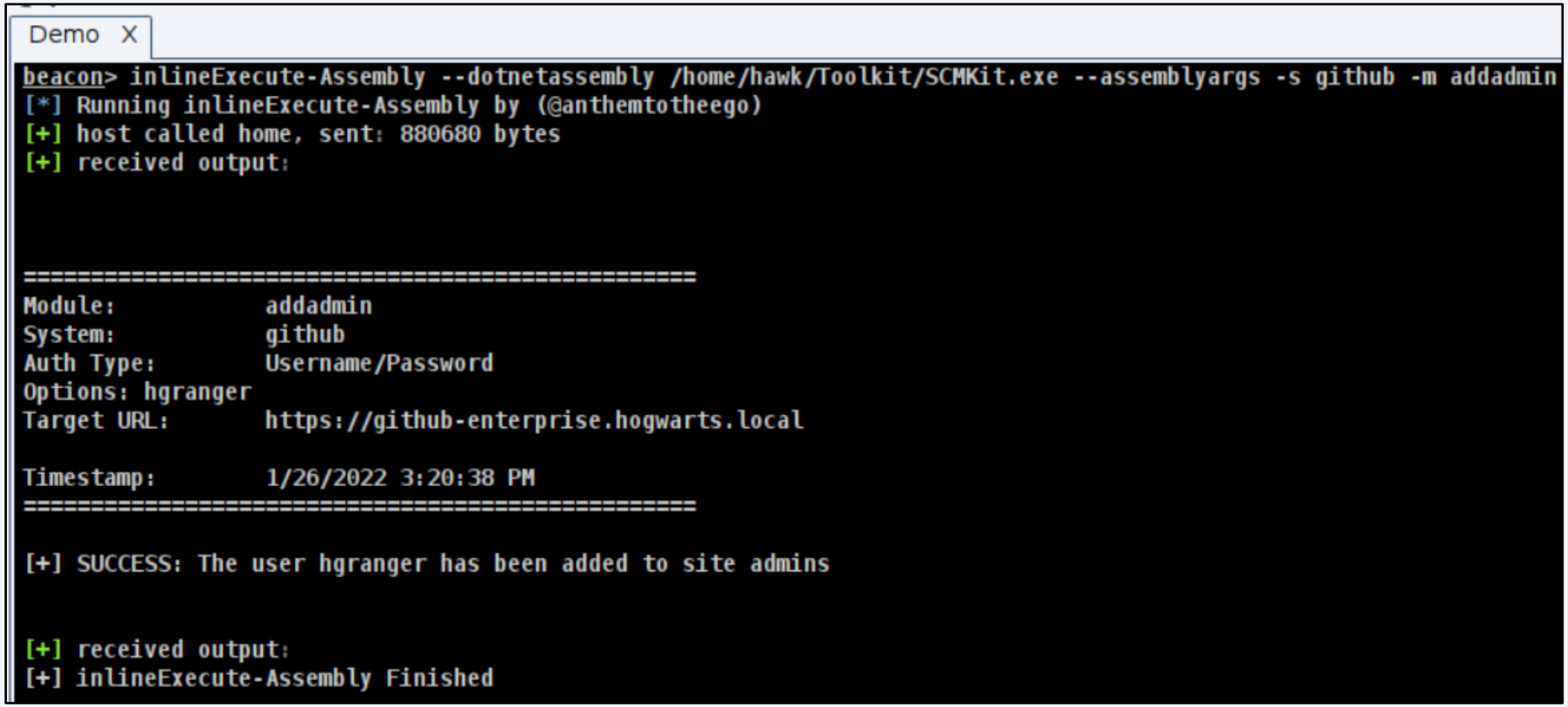
Module Details

Attack Scenario	Module	Requires Admin?	GitHub Enterprise	GitLab Enterprise	Bitbucket Server
Persistence	listsshkey	No	Yes	Yes	Yes
Persistence	removesshkey	No	Yes	Yes	Yes
Persistence	createsshkey	No	Yes	Yes	Yes
Persistence	listpat	No	No	Yes	Yes
Persistence	removepat	No	No	Yes	Yes
Persistence	createpat	Yes (GitLab Enterprise only)	No	Yes	Yes
Privilege Escalation	addadmin	Yes	Yes	Yes	Yes
Privilege Escalation	removeadmin	Yes	Yes	Yes	Yes

Example – Searching for Credentials



Example – Adding User to Site Admin



Example – Creating Personal Access Token

```
beacon> inlineExecute-Assembly --dotnetassembly /home/hawk/Toolkit/SCMKit.exe --assemblyargs -s gitlab
[*] Running inlineExecute-Assembly by (@anthemtotheego)
[+] host called home, sent: 880669 bytes
[+] received output:



================================================
Module:           createpat
System:           gitlab
Auth Type:        API Key
Options: hgranger
Target URL:       https://gitlab.hogwarts.local

Timestamp:        1/26/2022 3:10:13 PM
================================================

   ID |         Name |                          Token
------------------------------------------------------
   61 | SCMKIT-oHQpZ |          G4RzYez1_6Qzr1n48R_U

[+] SUCCESS: The hgranger user personal access token was successfully added.



[+] received output:
[+] inlineExecute-Assembly Finished
```

Demos

Demos

Demo 1 – Using SCMKit against Bitbucket Server

- Reconnaissance, Persistence, Privilege Escalation

Demo 2 - Using SCMKit against GitLab Enterprise

- Reconnaissance, Persistence, Privilege Escalation

Demo 3 - Using SCMKit against GitHub Enterprise

- Reconnaissance, Persistence, Privilege Escalation

Defensive Considerations

Defensive Considerations

- Static signatures within YARA rule file in SCMKit repo
- Static User Agent String (`SCMKIT-5dc493ada400c79dd318abbe770dac7c`)
- All access token and SSH key names created in SCM systems prepended with “SCMKIT-”

Questions?

Twitter: @h4wkst3r 

Discord: @h4wkst3r#9627 

SCMKit Tool:

https://github.com/xforcered/SCMKit


IBM