Microsoft Security

Microsoft Defender and vulnerability disclosure

Jonathan Bar Or, RSA 2022

```
jbo@McJbo ~ % mdatp

Microsoft Defender

Expected one of:
  config                Manage product configuration
  connectivity          Troubleshoot cloud connectivity
  definitions           Manage security intelligence updates
  diagnostic            Troubleshoot product issues and collect diagnostics
  edr                   Manage Endpoint Detection & Response (EDR) configuration
  exclusion             Manage antivirus exclusions
  health                Display product health information
  help                  Display all available options for this tool
  log                   Manage product logging
  notice                Display the Third-Party Notice
  scan                  Scan for malicious software
  device-control        Manage device control
  system-extension      Manage system extensions
  network-protection    Manage network protection
  threat                Manage threats and configure threat handling policies
  version               Display the product version

jbo@McJbo ~ % mdatp health
healthy                                     : true
health_issues                               : []
licensed                                    : true
engine_version                              : "1.1.19100.5"
app_version                                 : "101.66.54"
org_id                                      : "d7c7c745-195f-4223-9c7a-99fb420fd000"
log_level                                   : "info"
machine_guid                                : "1d2d6684-be58-5861-ab0d-095a19c9353c"
release_ring                                : "Internal"
product_expiration                          : Oct 19, 2022 at 08:01:27 PM
cloud_enabled                               : true
cloud_automatic_sample_submission_consent   : "safe"
cloud_diagnostic_enabled                    : true
passive_mode_enabled                        : false
real_time_protection_enabled                : true
real_time_protection_available              : true
real_time_protection_subsystem              : "endpoint_security_extension"
network_events_subsystem                    : "network_filter_extension"
device_control_enforcement_level            : "audit"
tamper_protection                           : "block"
automatic_definition_update_enabled         : true
definitions_updated                         : May 18, 2022 at 12:45:59 PM
definitions_updated_minutes_ago             : 8
definitions_version                         : "1.367.96.0"
```

Agenda

Motivation

- The motivation behind proactively discovering and disclosing vulnerabilities

Disclosures

- Disclosures so far
- MacOS SIP bypass
- MacOS TCC bypass
- Linux EoP

Conclusions

- Win-win-win
- Food for thought

Who am I?

- Jonathan Bar Or (“JBO”)
- Microsoft Defender research architect for Cross-platform.
  - Protecting everything that does not run Windows.
  - Validate product truth (red teaming, penetration testing).
  - Strategic-technical leadership of both research and engineering.
- Mix of offensive and defensive security.

Motivation

- In many cases we might want to create a complete attack chain.
- Example (macOS):
  - Start from a document with malicious macro on macOS (sandbox escape).
  - Implant persists and elevates privileges to root (elevation of privilege).
  - Implant steals browser cookies.
  - Implants silently turns on the microphone and starts recording (TCC bypass).
  - Implant loads a malicious kernel extension (SIP bypass).
- This is a win-win-win situation.
  - Responsibly disclosing bugs makes the world safer.
  - Our team gets better understanding of the technologies we work to protect.
  - We prove product truth and challenge our own detections.

Disclosures so far

MacOS

- SIP bypass
- TCC bypass
- Sandbox escape
- Elevation of Privilege

Linux

- Elevation of Privilege
- Information leak

Android

- Remote code execution
- Elevation of Privilege
- Information leak

Chrome OS

- Remote code execution

Example – SIP bypass

- System Integrity Protection (SIP) is a macOS mechanism that protects operations even from root.
  - Bypassing it effectively lowers the operating system's guards.
- For attackers, bypassing it could enable various operations:
  - Rootkits installation
  - Create undeletable files
  - Preventing updates

Example – SIP bypass (cont'd)

- While assessing SIP, we have discovered a way to bypass it by abusing a permission mechanism in macOS called "entitlements".
  - Worked with Apple to fix the issue.
  - Created a blogpost and generic detections.
- In 2022, a similar SIP bypass was reported by Perception Point.
  - Running it on our systems was detected out of the box.

```
root@JBO-MAC ~ # csrutil status
System Integrity Protection status: enabled.
root@JBO-MAC ~ # head -n 1 /Library/Apple/System/Library/Extensions/AppleKextExcludeList.kext/Contents/Info.plist
<?xml version="1.0" encoding="UTF-8"?>
root@JBO-MAC ~ # echo hi > /Library/Apple/System/Library/Extensions/AppleKextExcludeList.kext/Contents/Info.plist
zsh: operation not permitted: /Library/Apple/System/Library/Extensions/AppleKextExcludeList.kext/Contents/Info.plist
root@JBO-MAC ~ # ./shrootless.sh "echo hi > /Library/Apple/System/Library/Extensions/AppleKextExcludeList.kext/Contents/Info.plist"

SHROOTLESS
SIP bypass by Jonathan Bar Or ("JBO")

Checking command line arguments ..................... [ OK ]
Checking if running as root ......................... [ OK ]
Checking for system_installd ........................ [ OK ]
Downloading Apple-signed package .................... [ OK ]
Writing '/etc/zshenv' payload ....................... [ OK ]
Running installer ................................... [ OK ]
Cleaning up ......................................... [ OK ]

> Great, the specified command should have run with no SIP restrictions. Hurray!
> Quitting.
root@JBO-MAC ~ # cat /Library/Apple/System/Library/Extensions/AppleKextExcludeList.kext/Contents/Info.plist
hi
root@JBO-MAC ~ # ls -l@O /Library/Apple/System/Library/Extensions/AppleKextExcludeList.kext/Contents/Info.plist
-rw-r--r--  1 root  wheel  restricted 3 Jul 28 20:30 /Library/Apple/System/Library/Extensions/AppleKextExcludeList.kext/Contents/Info.plist
root@JBO-MAC ~ #
```

MacOS SIP bypass exploit

Example – TCC bypass

- TCC (Transparency, Consent and Control) is a mechanism that protects access to user’s private data
  - Private files
  - Camera and microphone
  - Calendar
- Very attractive for attackers:
  - Taking pictures silently
  - Silent microphone access
  - Private file access

Example – TCC bypass (cont'd)

- While assessing TCC, we have discovered numerous ways to completely bypass TCC checks.
  - Worked with Apple to fix the issues.
  - Created a blogpost and generic detections.
- New TCC bypasses were found later by independent researchers.
  - Microsoft Defender's detections were able to detect them with no changes.

```
root@JBO-MAC ~ # /tmp/powerdir_exploit.sh /Applications/Microsoft\ Teams.app
```

POWERDIR

TCC bypass by Jonathan Bar Or ("JBO")

```
Checking command line arguments ........................ [ OK ]
Checking if running as root ........................... [ OK ]
Checking user's old home directory .................... [ OK ]
Preparing fake directory structure .................... [ OK ]
Getting app's csreq blob .............................. [ OK ]
Building fake TCC.db .................................. [ OK ]
Changing home directory ............................... [ OK ]
Restarting user's TCC daemon .......................... [ OK ]
```

MacOS TCC bypass exploit

```
> Great, application 'com.microsoft.teams' now enjoys microphone and camera access. Hurray!
> Quitting.
```

Potential Transparency, Consent and Control bypass

Part of incident: Potential Transparency, Consent and Control bypass on one endpoint View incident page

McJbo Risk level Medium
Data sensitivity: General +3

root

ALERT STORY

11/12/2021 11:52:50 AM [19693] sudo

11:52:50 AM [19693] zsh

11/15/2021 11:16:25 AM [28722] zsh

11:16:25 AM [28722] sh ./powerdir_confiqd.sh "/Applications/Microsoft Te...

11:16:25 AM [28752] sh ./powerdir_confiqd.sh "/Applications/Micros...

11:16:25 AM [28752] sqlite3 "/tmp/pwdir_tcc_bypass/jbo/Lib...

Potential Transparency, Consent and Control bypas...
Medium Detected New

11:16:25 AM sh accessed sqlite database through invocation of sq...

Medium Detected New

Potential Transparency, Consent and Control bypass

Medium Detected New

Manage alert See in timeline

Alert description

A potential Transparency, Consent and Control (TCC) bypass was detected. TCC protects user data by enforcing security controls when resources that could compromise user privacy are accessed, such as via microphone, camera, calendar, or full disks. Bypassing TCC enables attackers to gain access to user-sensitive data.

Alert recommended actions

1) Review the process performing the bypass and its command line.
2) Review the process tree for other suspicious processes.
3) Review the machine timeline for

TCC bypass detection by Microsoft Defender

Example – Linux D-Bus Elevation of Privilege

- Our team started assessing an inter-process communication channel known as D-Bus.
  - Discovered several vulnerabilities - timing bugs, directory traversal and symlink attacks.
  - By chaining the vulnerabilities together, we were able to get root arbitrary code execution.
  - Our exploit was detected by Microsoft Defender at development stages!
- Worked with the D-Bus service maintainer to get the fix out.
  - Reported to RedHat as well
  - Added further generic detections in case of different exploitation strategies

```
jbo@jbo-nix:~/1337$ ./nimbuspwn.py
```

NIMBUSPWN

networkd-dispatcher Linux EoP by Jonathan Bar Or ("JBO")

```
Attempting to own dbus name org.freedesktop.network1 ..................... [  OK  ]
Validating name patterns ................................................... [  OK  ]
Planting base directory .................................................... [  OK  ]
Planting symlink ........................................................... [  OK  ]
Planting payload ........................................................... [  OK  ]
Attempting to win the race [1/6] ........................................... [ RETR ]
Attempting to win the race [2/6] ........................................... [ RETR ]
Attempting to win the race [3/6] ........................................... [ RETR ]
Attempting to win the race [4/6] ........................................... [  OK  ]
> Great, we now have a root backdoor. Hurray!
> Enjoy your root privileges.

# head -n1 /etc/shadow
root:!:18267:0:99999:7:::
# id -u
0
# exit
jbo@jbo-nix:~/1337$
```

Linux D-Bus EoP exploit

Suspicious execution of SUID/SGID process

jbo-nix Risk level Low

NonWindows

ALERT STORY

Expand all

[2503] apt-get

[2509] apt-get -q update

[2509] dash sh -c "/usr/lib/update-notifier/update-motd-updates-available 2>/dev/null || true"

[2510] dash sh -c "/usr/lib/update-notifier/update-motd-updates-available 2>/dev/null || true"

[2510] dash /bin/sh -e /usr/lib/update-notifier/update-motd-updates-available

[2519] dash /bin/sh -e /usr/lib/update-notifier/update-motd-updates-available

[2519] apt-config shell SourceList Dir::Etc::sourcelist

[2520] nimbuspwn

Details

Suspicious execution of SUID/SGID process

Low New

See in timeline

Consult a threat expert

Manage alert

Classify this alert

True alert

False alert

Status New

Classification Select classification...

Assign to Unassigned

Alert details

Incident Suspicious execution of SUID/SGID process on one endpoint (open in Microsoft 365 Defender)

Detection technology Behavioral

Linux generic detection by Microsoft Defender

Conclusions

- By responsibly disclosing vulnerabilities, we:
  - Get ahead of attackers.
  - Collaborate to benefit end-users everywhere.
  - Make Microsoft Defender better by handling classes of attacks.
  - Challenge our own blue teams.
- We are actively working on more interesting disclosures, stay tuned!

Thank you