Sophisticated technique of the year goes to...

Author:
Kalpesh Mantri
Quick Heal – India
kalpesh.mantri@quickheal.com

Abstract:

2019, as has been the case in recent past, was again full of new malware campaigns and APT attack discoveries. Some were discovered for the first time; while many made a comeback. We have been tracking such attacks for several years and have observed variety of techniques being used in them. In this talk we would share few highly sophisticated techniques used by attackers; that have helped the attacks to stay undetected for years. These techniques are not very prevalent at this point; however, we suspect more and more attacks in future would adopt them.

This paper will focus on some highly sophisticated techniques used in malware campaigns and APTs in 2019. In this talk I will discuss following techniques / attacks:

- An APT actor was found communicating with Command and Control servers over VPN. This APT was able to bypass Two Factor Authentication [2FA] as well!
- Are Password managers safe? Should we use any? They are being increasingly targeted by threat actors to get credentials
- We will explore how a ransomware group is using Wake-on-LAN [WoL] feature to increase monetization of infections
- Web skimmers started using a retired technique called Steganography and are still successfully evading security solution. Would this technique make a comeback in 2020?

During the talk, I would share insights on the techniques used in these attacks and would discuss the questions called out above. This paper's intent is to bring these sophisticated techniques to defenders' notice so that we all can work on proactively blocking attacks using them.

Introduction

Cyber Threat actors constantly invest in improving their tools & techniques, to stay ahead of latest security solutions. This necessitates defenders to evolve & come up with even better ways to tackle Cyber Attacks. One of the approaches that's catching up in Defenders Community is to move upwards in "The Pyramid of Pain" ladder [introduced by David Bianco].

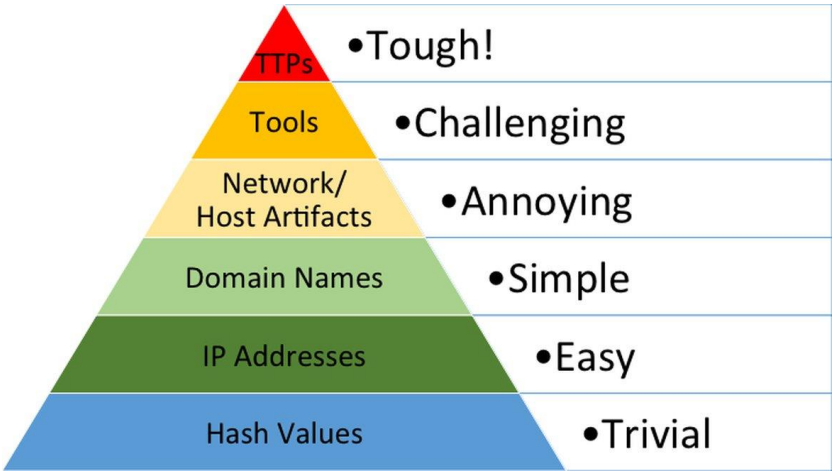

Image: Pyramid of Pain

As per the pyramid, if defenders can prevent, hunt, & detect attacks based on the attacker Tactics, Techniques and Procedures [TTPs], then they make cyber-attacks very expensive for the attackers to pivot their path.

As part of our ongoing tracking of cyber-attacks; we have been analyzing the TTPs used in them. This paper will talk about some sophisticated attack techniques observed during last year. These techniques are not ubiquitous at this point; however, we believe it's only a matter of time before that happens.

Exfiltration via VPN with 2-Factor Authentication:

A VPN solution provides secure communication channel by extending private network over public internet. Almost all VPN solutions today support end-to-end encryption & multi factor authentication to enhance security posture.

Most organizations use VPN solutions to enable remote connectivity to on-premise resources. Interestingly, in a recent case of cyber-attack, actor used organization's VPN channel to exfiltrate data to their servers. They even bypassed VPN solution's Multi-Factor authentication mechanism.

In this case, target organization was using RSA SecurID token generation software to create 2-factor codes on endpoints. SecurID token is usually generated for a "specific system" and is supposed to be tied with that system. But this software had a bug; it checked the "specific system value" only when importing SecurID Token Seed, but didn't use it at the time of generating actual 2-factor tokens. So by patching the verification bytes, the software could be made to work on any system.

Image: RSA SecurID generating valid 2 factor codes

Attacker, in this case, stole a RSA SecurID Software Token and then patched that simple instruction. Afterwards they could generate valid tokens on any machine and use it to exfiltrate data. With this approach attackers were able to hide their traffic inside usual VPN traffic of the organization.

This technique seems to be gaining popularity among other groups as well. Recently few banking trojans and android malwares were seen making use of similar technique.

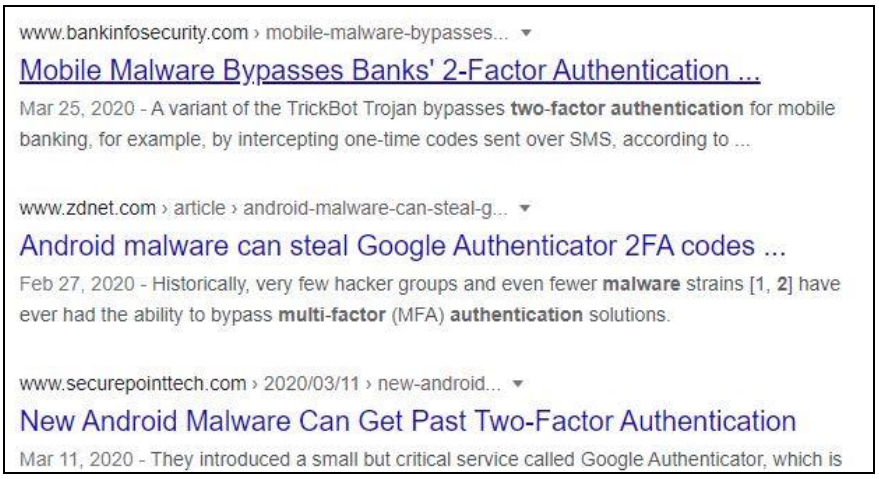

Image: 2FA TTP used by recent malwares

Defender community and organizations should pay attention to this technique as attacks using this mechanism will likely grow in coming years.

Password Managers are not so safe!

Several organizations use password managers to assist employees to store passwords safely. KeePass, one such Password Manager tool, was targeted in a recent attack.

In this attack, the attacker retrieved passwords saved in KeePass vault & then used them to infiltrate further inside the organization. This approach completely does away with brute force attempts to break passwords, which may alert the organizations security solutions. With this approach, attacker could navigate silently and remain undetected for longer period.

Attackers used KeeThief, an open source PowerShell tool, to recover master password from running KeePass process. KeeThief is infact a popular credentials dumping tool and is available on github.

https://github.com/HarmJ0y/KeeThief/

When KeePass is **running** and the **database is unlocked**, KeeThief is able to recover the following information from memory:

- Database Location
- KeePass Version and Location
- Master Password
- Key File (base64)
- Windows User Account (base64)

This injection only requires permission to modify the KeePass process space (which the current user running `KeePass.exe` has); **it doesn't require administrative rights**.

Image: KeeThief functionality

The attacker apparently took inspiration from Will Schroeder's [@harmj0y] BSidesNola 2017 presentations and followed similar steps. On target systems, the contents of password managers were directly targeted and retrieved.

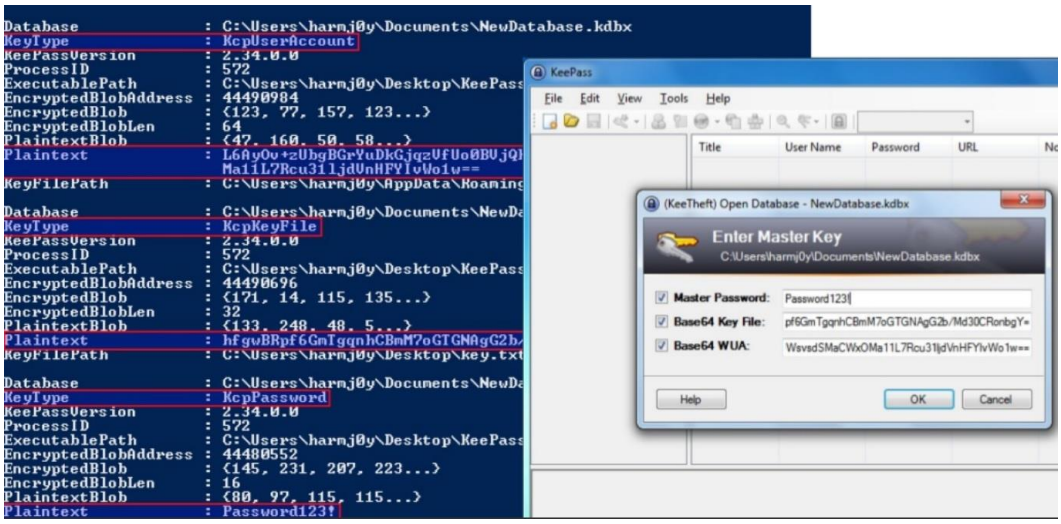

Image from @harmj0y slide: Recovering the plaintext master password for KeePass DB

This technique can be monitored and detected via:

- Keeping watch on cross-process interaction (OpenProcess, CreateRemoteThread, ReadProcessMemory, WriteProcessMemory) via Host-based monitoring tools like Sysmon.
- Monitor changes to KeePass config file from non-KeePass process.
- Powershell and WMI events to KeePass modules.

Wake-up, I wanna infect you!

Wakeup-on-LAN (WoL) is a networking standard that allows a computer to be turned on or awakened by a network message. The message is usually sent to the target computer by a program executed on a device connected to the same local area network. Since this message is sent over data link or OSI-2 layer, it is susceptible to be abused by anyone on the LAN.

Infact in some of the attacks last year, we saw WoL being used by Ryuk Ransomware to encrypt even the sleeping machines, thus increasing the reach & impact of the attack. Our preliminary analysis showed that infecting more systems via WoL, helped Ruyk raise more money in form of ransom payments. This leads us to believe that more ransomware and malware will adopt this technique in coming years.

```
d_GetIpNetTable(0, &v17, 1);
v2 = (unsigned int *)d_VirtualAlloc(0, v17, 4096, 4);
d_GetIpNetTable(v2, &v17, 1);
v15 = d_VirtualAlloc(0, 24 * *v2, 4096, 4);
GlobalAlloc(0x40u, 0x4000u);
if ( *v2 > 0 )
{
  v3 = (int *)(v2 + 5);
  do
  {
    if ( *(v3 - 3) )
    {
      v16 = *v3;
      ExtractIPv4Address_Buffer(2, &v16, &cp);
      v8 = *((_BYTE *)v3 - 8);
      v9 = *((_BYTE *)v3 - 7);
      v10 = *((_BYTE *)v3 - 6);
      v11 = *((_BYTE *)v3 - 5);
      v12 = *((_BYTE *)v3 - 4);
      v13 = *((_BYTE *)v3 - 3);
      v4 = 0;
      v5 = 0;
      do
      {
        v6 = *(&v8 + v5);
        if ( v6 != 255 && v6 )
          --v4;
        else
          ++v4;
        ++v5;
      }
      while ( v5 < 6 );
      if ( v4 < 4 )
        initiate_nw_conn(&cp, (int)&v8);
```

Image: Extracting ARP Table of System

```
if ( WSAStartup(0x202u, &WSAData)
  || (v6 = socket(2, 2, 0x11), v6 == -1)
  || setsockopt(v6, 0xFFFF, SO_BROADCAST, &optval, 1)
  || (memset(&name, 0, 0x10),
      name.sa_family = 2,
      *(_DWORD *)&name.sa_data[2] = htonl(0),
      *(_WORD *)&name.sa_data[0] = htons(0),
      bind(v6, &name, 0x10))
  || (memset(&to, 0, 0x10),
      to.sa_family = 2,
      *(_DWORD *)&to.sa_data[2] = inet_addr(cp),
      *(_WORD *)&to.sa_data[0] = htons(7u),
      sendto(v6, buf, 102, 0, &to, 0x10) == -1) )
{
  result = 0;
}
else
{
  d_Sleep(250);
  closesocket(v6);
  WSACleanup();
```

Image: Magic Packet for WoL Implemented by Ryuk

Table: Code Implemented by Ryuk

A deep dive implementation of this technique in Ruyk can be seen here:
https://blogs.quickheal.com/deep-dive-wakeup-lan-wol-implementation-ryuk/

Will Steganography make a comeback?

Steganography is an old technique of hiding malicious code in images and other carrier files. Cyber criminals, in past, have used this technique successfully, to compromise machines just by getting users to visit a website where the image is hosted or by simply sending that image via email & luring user into opening the image.

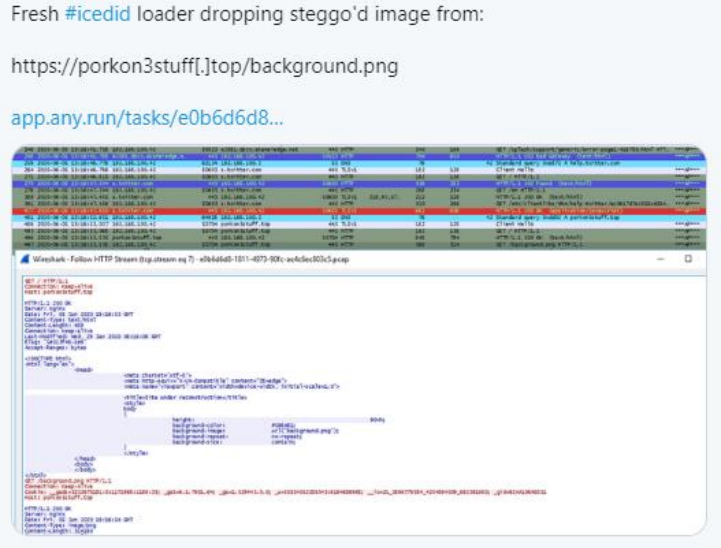

Image: Recent in-the-wild use of Steganography technique

We believe steganography technique has already started its comeback. From past few years, there has been a notable increase of in-the-wild malware campaigns using steganography and similar tricks to embed malicious code in pictures and other carrier files. Here are some notable in-the-wild usage of this technique (source: SentinelOne):

- AdGholas – this malware hides malicious JavaScript in image, text, and HTML files
- Cerber – embeds malicious code in image files
- DNSChanger – uses PNG LSBs to hide malware AES encryption key
- Stegano – PNG formatted banner ads containing malicious code
- Stegoloadr – this malware uses both steganography and cryptography to conceal an encrypted URL to deliver later stage payloads
- Sundown – white PNG files are used to conceal exploit code or exfiltrate user data
- SyncCrypt – ransomware that hides part of its core code in image files
- TeslaCrypt – HTML comment tags in an HTTP 404 error page contain C2 server commands
- Vawtrak – hides a URL in the LSBs of favicons to download a malicious payload
- VeryMal – malware targets macOS users with malicious JavaScript embedded in white bar
- Zbot – appends data to the end of a JPEG file containing hidden data
- ZeroT – Chinese malware that uses steganography to hide malware in an image of Britney Spears

Perhaps the more worrying trend is apparent use of steganography in targeted attacks. The technique is gradually becoming a part of some major Cyber Attack groups' arsenal.

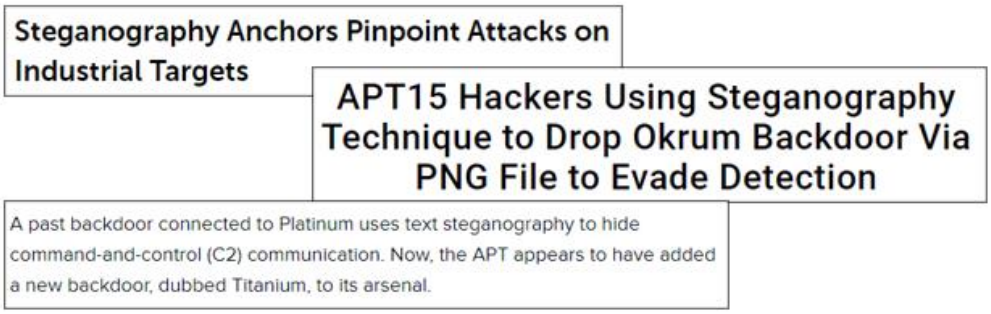

Image: Articles showing steganography use by APTs

Given the prevalence of image-based advertisements and popularity of image sharing on social websites, we expect use of this technique in malware to grow.

Conclusion:

The techniques highlighted in this paper are relatively simple to implement & are also used for legitimate purposes. This means attackers do not have to work very hard to hide them; on the other hand, it is much harder for security products to isolate these attacks. As a result attackers have started using these techniques & we expect them to become prevalent in time to come

The intent of this paper is to bring these sophisticated techniques to defenders' notice so that we all can work on proactively blocking attacks that use them.

Based on what we know and what we've gleaned from others' publications, and through industry sharing, the amount of information collected on such techniques is so large that it has not been possible to include all other details that relate it to threat groups. Therefore, our analysis and finding of the sophisticated techniques will continue in several directions and we will continue to publish such sophisticated techniques of each coming years.