Donanım Yazılımı Analizinin Önemi

written by Mert SARICA | 1 October 2015

Her siber güvenlik konferansında gerçekleştirdiğim sunumumdan sonra olduğu gibi yine geleneği bozmayarak, 2014 yılında IstSec ve geçtiğimiz Eylül ayında Hacktrick siber güvenlik konferanslarında gerçekleştirdiğim donanım yazılımı (bellenim / firmware) analizi sunumumdan sonra sunuma katılamayanlar için sunumu özetleyen bir blog yazısı yazmaya karar verdim.

Nesnelerin İnterneti (IoT) dediğimiz kavram hayatımıza girdi gireli, evimizde internete bağlanan birçok cihaz olduğunu görebiliyoruz. Bunlar arasında uydu alıcıları, ip kameraları, cep telefonlarını, raspberry pi gibi mini bilgisayarları en çok rastlanan nesneler arasında sayabiliriz. Hacktrick sunumunda, evinde 7/24 çalışan Raspberry Pi, Beagle Bone gibi mini bilgisayarı olanlar el kaldırsın dediğimde, havaya kalkan ellerin sayısının beklediğimden fazla olduğunu söyleyebilirim. Durum böyle olunca da IoTler, akıllı cihazlar ve benzerleri, hayatımıza getirdikleri kolaylıkların yanında güvenlik risklerini de beraberinde getiriyorlar dersek pek yanılmayız.

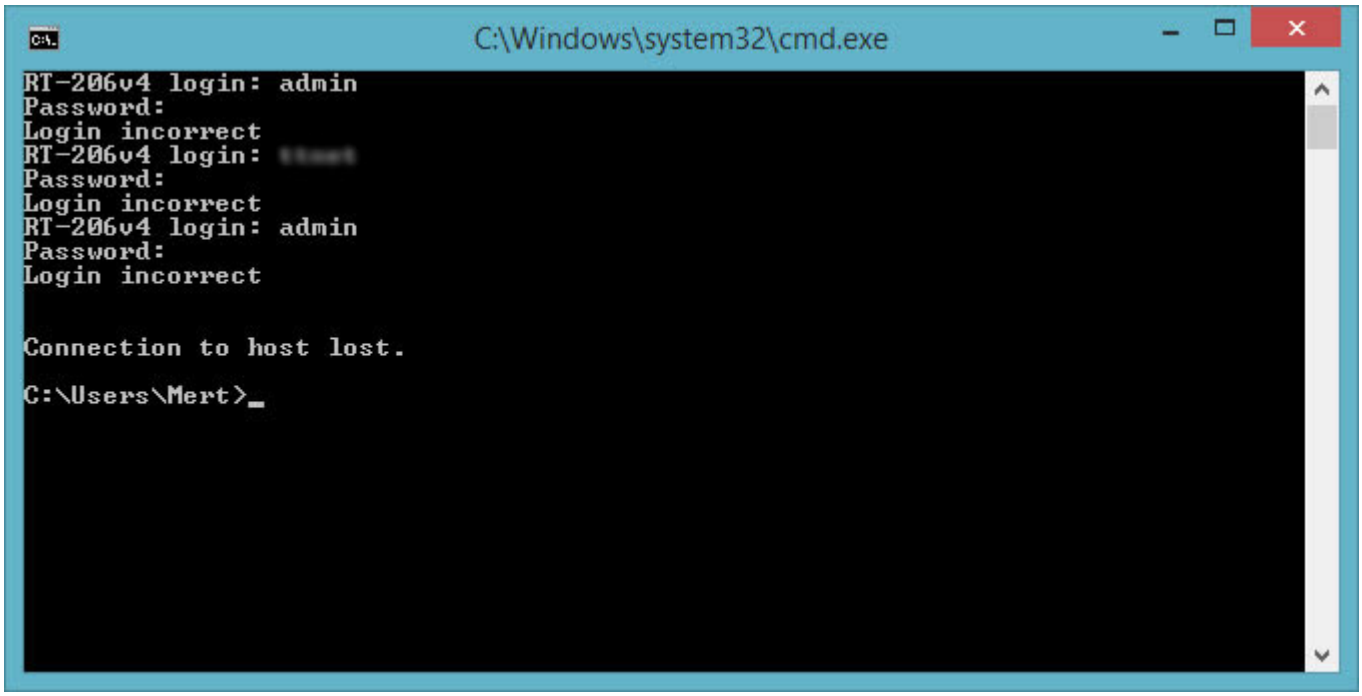
Aslında evlerimize soktuğumuz cihazlar akıllandıkça, casus olma potansiyeline de sahip olmaya başladılar. Hareketle kontrol edilen kameralı, akıllı televizyonunuz hacklendiğinde, başınıza gelebilecekleri bir düşünün, sevimsiz öyle değil mi ? :) Aslında haberlere baktığımızda bu söylediklerimin çok da uzak bir ihtimal olmadığını görüyoruz. Ağustos ayında Samsung'un akıllı buzdolabının hacklenerek Gmail kullanıcı adı ve şifre bilgilerinin çalınabildiği ortaya çıktı.

Bu tür cihazların hacklenebilmesi, modern işletim sistemlerine (Windows, Linux vs.) kıyasla daha kolay oluyor çünkü bu cihazlar çoğunlukla düşük donanımlarla çalışıyorlar. Düşük donanım dediğimizde de, ram, işlemci ve işletim sistemi açısından zayıf/kısıtlı olan bu cihazlar üzerinde örneğin Address Space Layout Randomization (ASLR), data execution prevention (DEP) gibi istismarı engelleyici kontroller bulunamayabiliyor. Donanım yazılımı geliştiricileri, modern işletim sistemi geliştiricileri gibi güvenliğe ön planda tutmadıkları için de çoğunlukla cihazlar üzerinde yer alan konfigürasyonlar örneğin modemlerde olduğu gibi zayıf ve istismara açık olabiliyor.

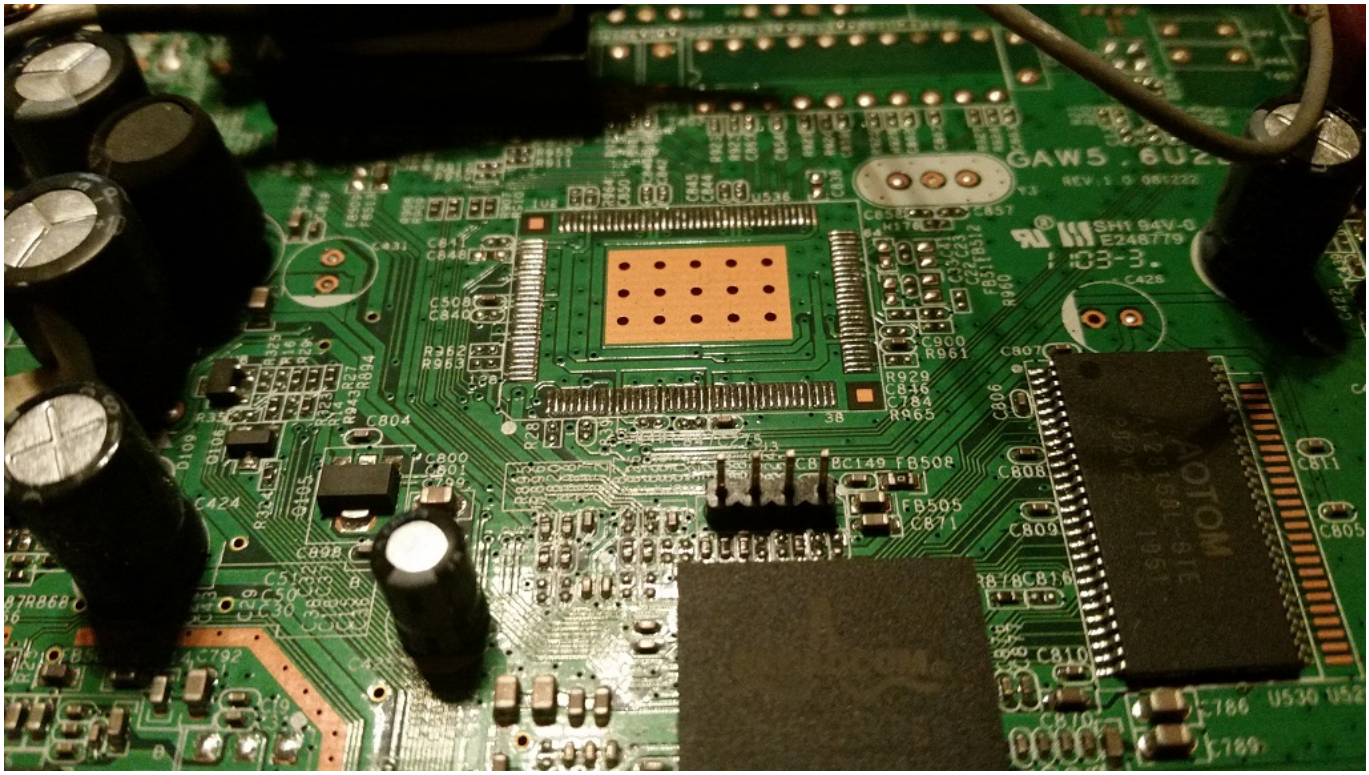
Bu cihazların güvenliği istenilen seviyelerde olmadığı sürece, güvenlik uzmanları ve son kullanıcılar olarak, ev ağımızda yer alan bu cihazların donanım yazılımlarını analiz ederek güvenlik zafiyetlerini tespit etmek, hem merakımızı gidermek için hem de bu cihazları güvenli bir şekilde kullanmak isteyen bizler için bir gereksinim haline gelebiliyor.

Örneğin elimizde internet servis sağlayıcısı tarafından bize kampanya dahilinde hediye edilmiş bir modem var ve bu modeme yönetici arayüzünden bağlanıp, modem üzerinde tanımlı kullanıcılar görüntülemek istiyoruz. Neden bunu istiyoruz çünkü modemler üzerinde kimi zaman varsayılan yönetici yetkisine sahip hesaplar olabiliyor veya internet servis sağlayıcısı uzaktan destek amacıyla kolay tahmin edilebilir parolaya sahip kullanıcı hesaplarını modemlere tanımlayabiliyorlar. Varsayılan hesaplar dışında modemin yönetici arayüzüne giriş yaptığımızda göremediğimiz ancak ilgili sayfayı direk çağırdığımızda ulaşabileceğimiz ve modem üzerindeki özel ayarları (TR-069 yönetim protokolü ayarları gibi) değiştirmemizi sağlayan gizli sayfalar olup olmadığını da kontrol etmek istiyoruz. Bu sorulara yanıt bulmak için modem ve modemin donanım yazılımı üzerinde çeşitli kontroller gerçekleştirebiliriz.

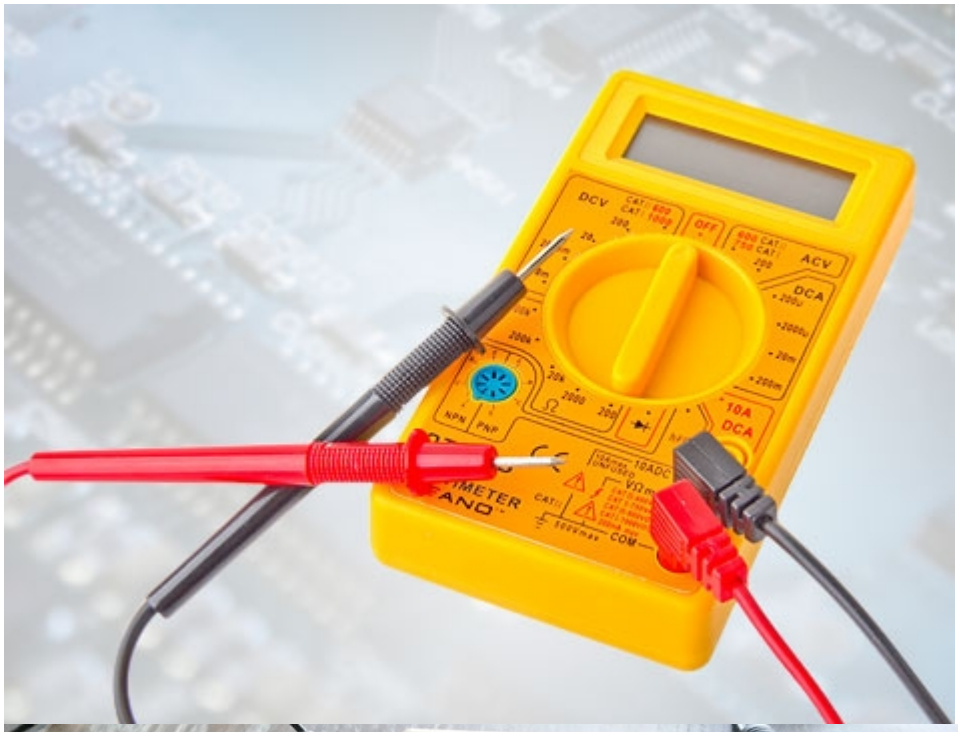
Gizli yönetici hesaplarını bulmak için yapacağımız ilk iş, modemin telnet servisine bağlandıktan sonra cat /etc/passwd komutu yazarak mevcut hesapları kontrol etmek olabilir ancak işler her zaman düşündüğümüz kadar kolay olmayabilir. Birincisi, yönetim paneline erişmek için kullandığımız kullanıcı adı ve şifrenin telnet servisine erişmek için yetkisi olmayabilir veya telnet servisi (telnetd) modem üzerinde açık/yüklü olmayabilir.

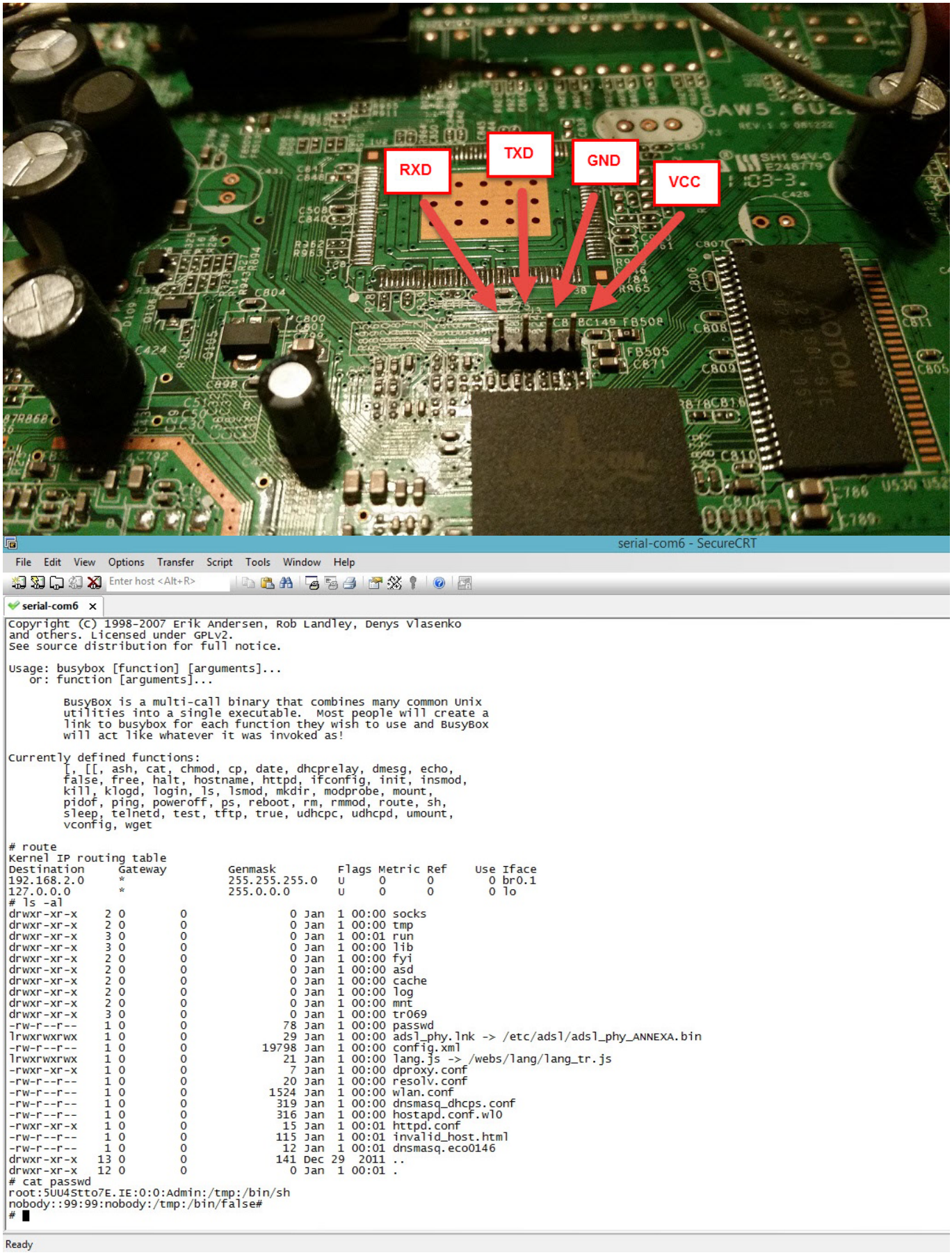


Böyle bir durumda yapmamız gereken bir tornavida seti alarak modemi açmak ve üzerinde UART seri bağlantı noktası aramak olabilir. Şanslıysak 4 PIN'den oluşan bu bağlantı noktasını çok geçmeden tespit edebiliriz.



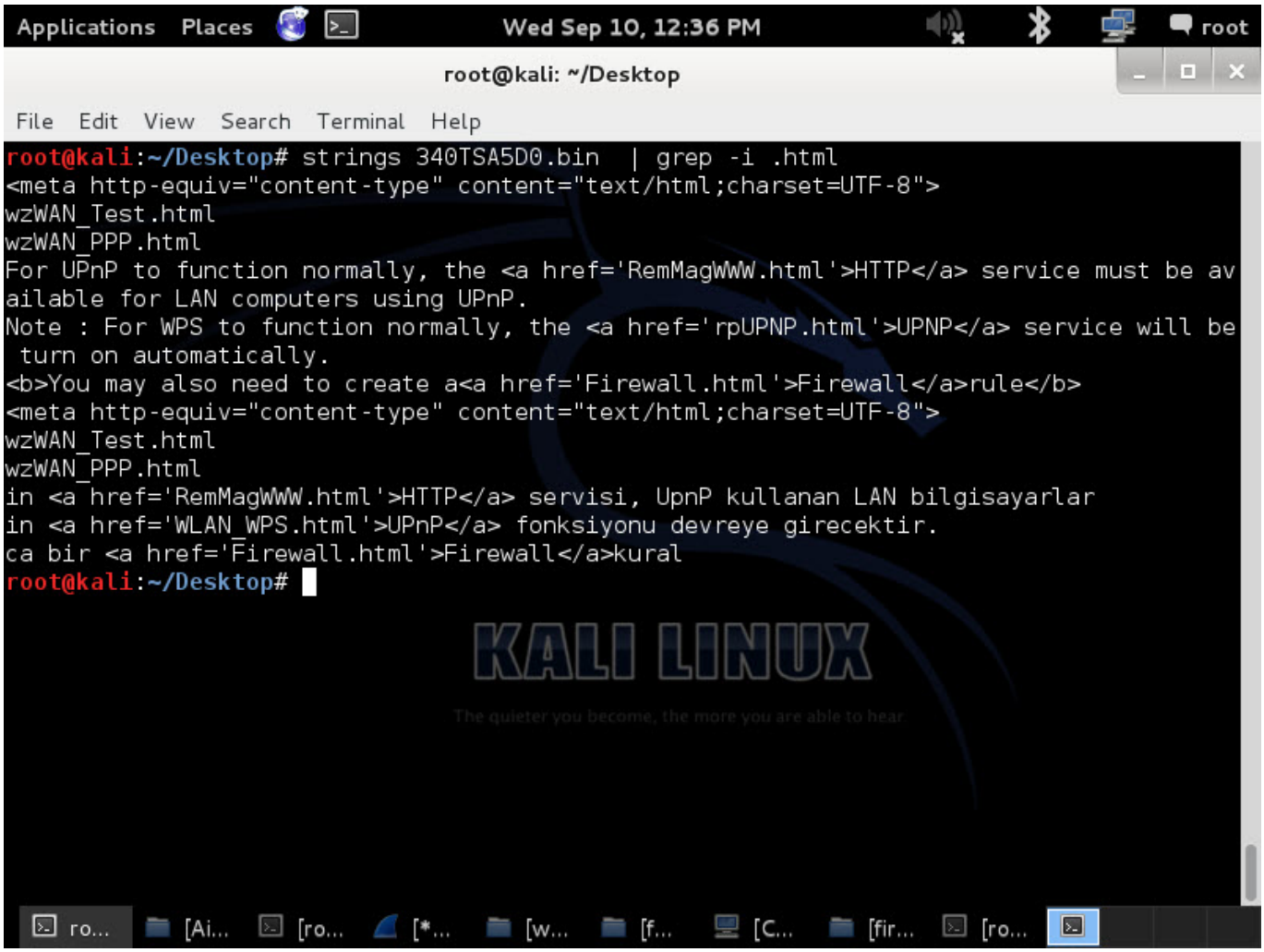
Tabii bu seri bağlantı noktası üzerinden modem ile iletişim kurabilmek için USB – TTL UART CP2104 çevirici gibi bir aygıta ihtiyaç duyacağız. Aygıtı bağlamak için öncelikle o dört pinden hangisi veri almak (RX), hangisi veri göndermek (TX) ve hangisi topraklama (GND) için kullanılıyor onu bilmemiz gerekiyor. Süreklilik testi sayesinde Dijital Avometre / Multimetre'de siyah ucu toprağa (işaretli bir kutup), kırmızı ucu ise pinlere sırasıyla dokundurduğumuzda bir ses duyuyorsak o zaman bu pinin toprak (GND) pini olduğunu anlayabiliriz. Ardından RX, TX pinlerini ve baud oranını deneme yanılma yolu ile SecureCRT veya Putty ile tespit ederek komut satırına erişim sağlayabiliriz.



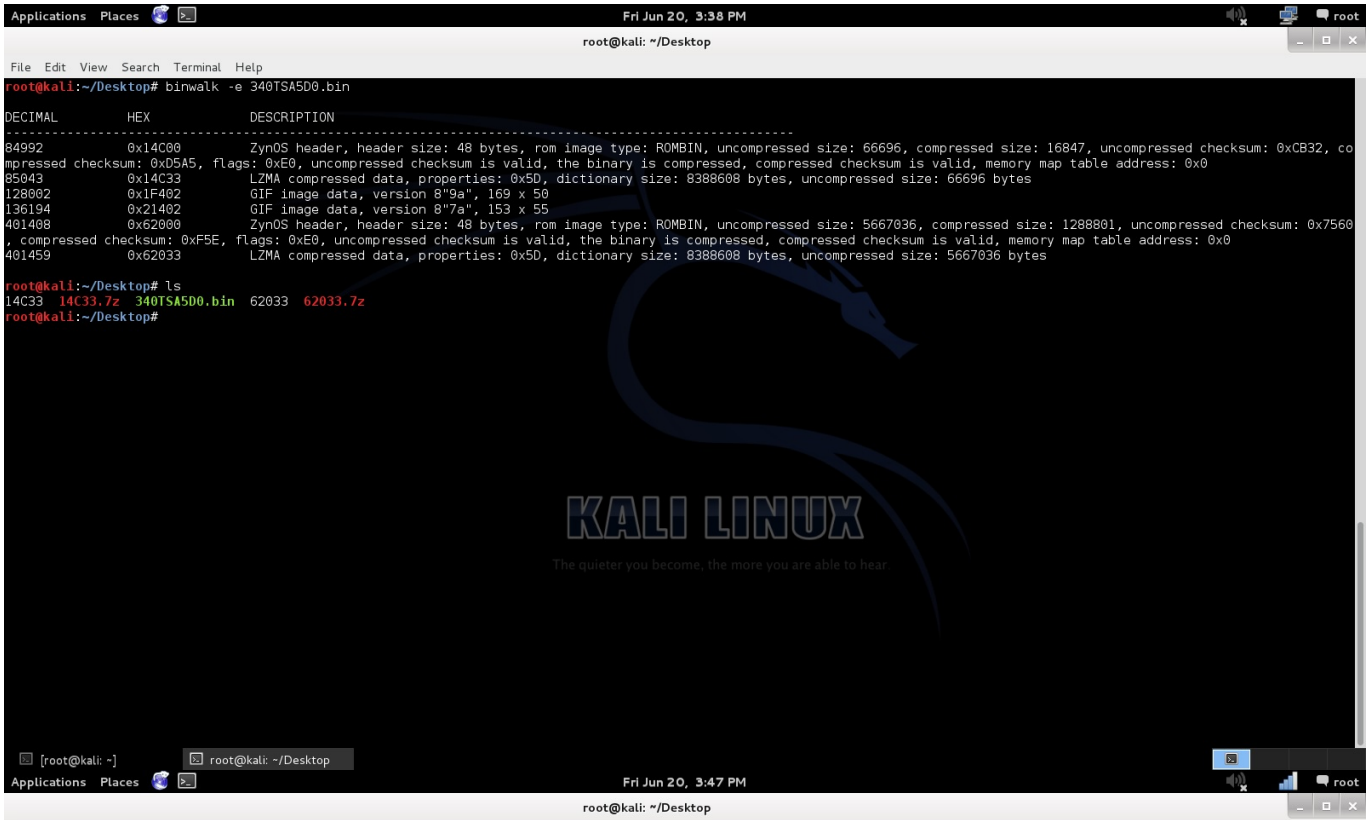


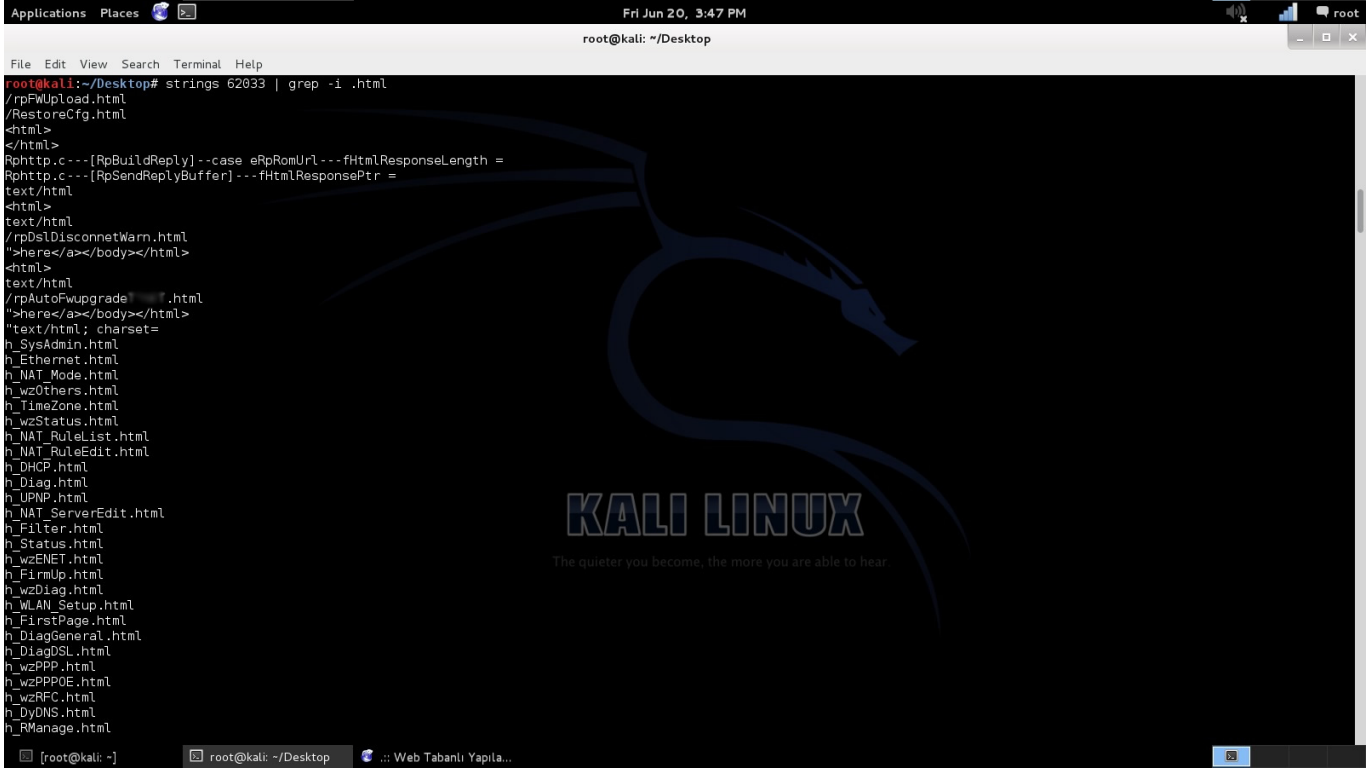
Bir diğer örnekte ise modemin yönetici arayüzündeki gizli sayfaları tespit

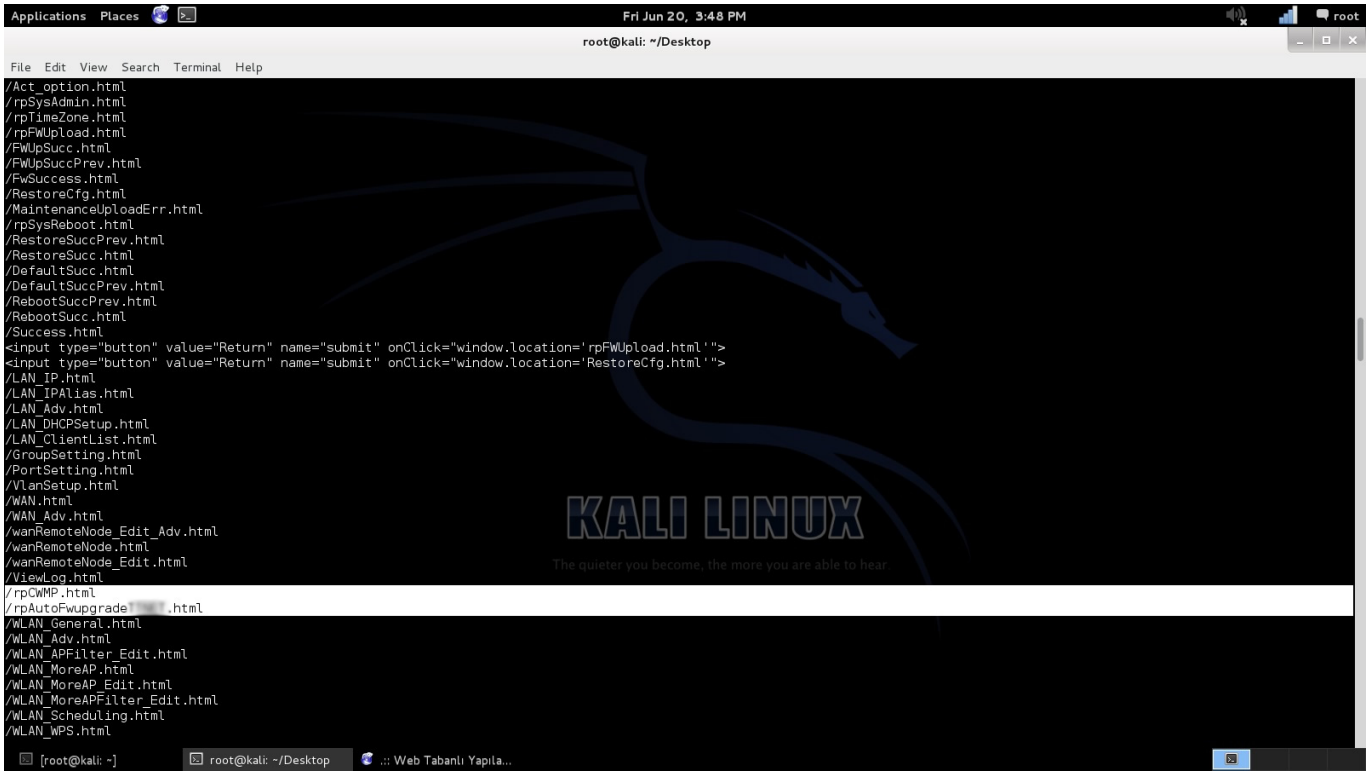
etmek istiyoruz. Bunun için ilk iş, donanım yazılımını (bellenim) üreticinin veya internet servis sağlayıcısının web sitesinden indirmek olacaktır. Ardından strings aracını bu donanım yazılımı üzerinde çalıştırabiliriz. Eğer aracın çıktısı aşağıdaki örnekte olduğu gibi sayıca az html sayfa adı veriyorsa ancak biz arayüzde çok daha fazla sayıda html sayfa olduğunu biliyorsak, binwalk gibi farklı bir araç ile analizi bir adım ileriye taşıyabiliriz.



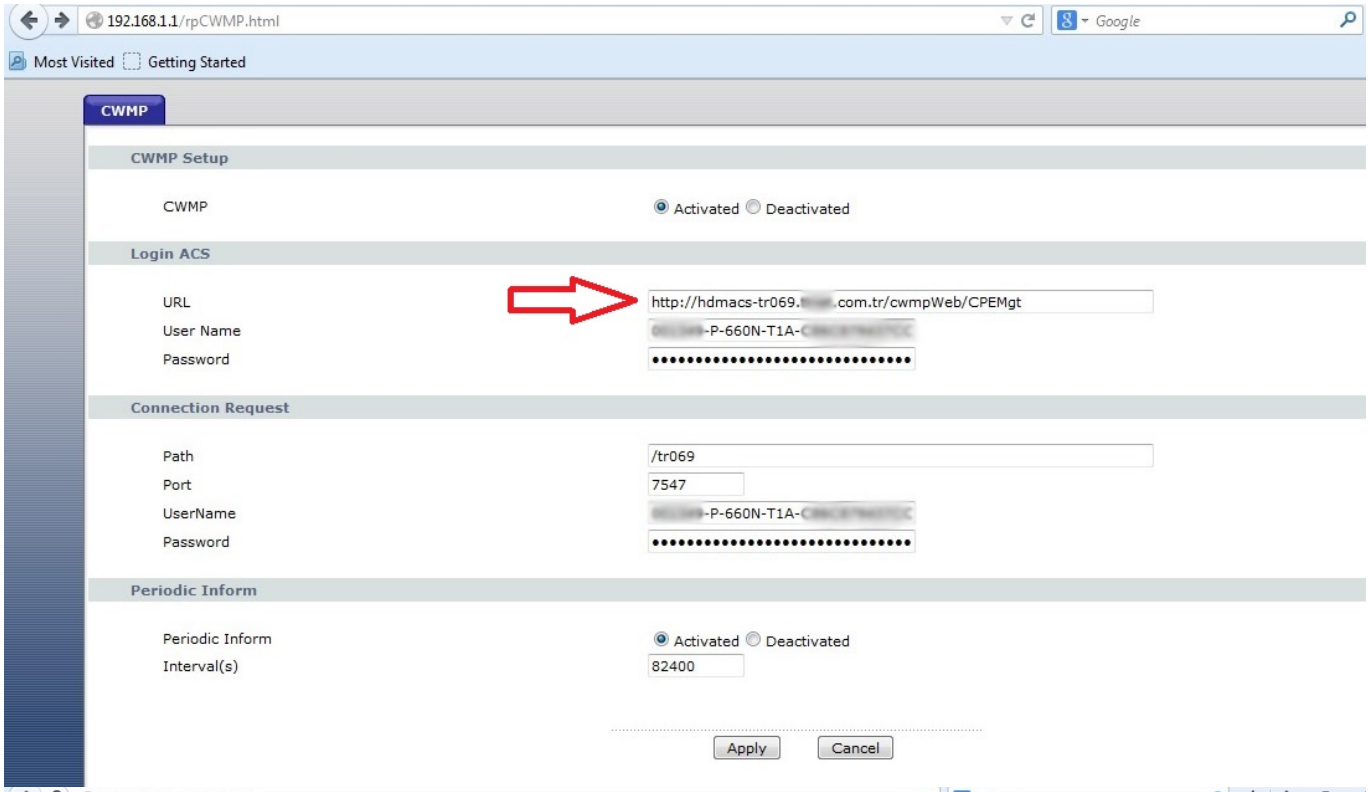
Donanım yazılımı analizi için biçilmiş kaftan olan binwalk aracı ile donanım yazılımını açtıktan (extract) sonra, çıkan dosyalar üzerinde strings komutunu çalıştırdığımızda çok daha fazla sayıda html dosya olduğunu görebiliriz.

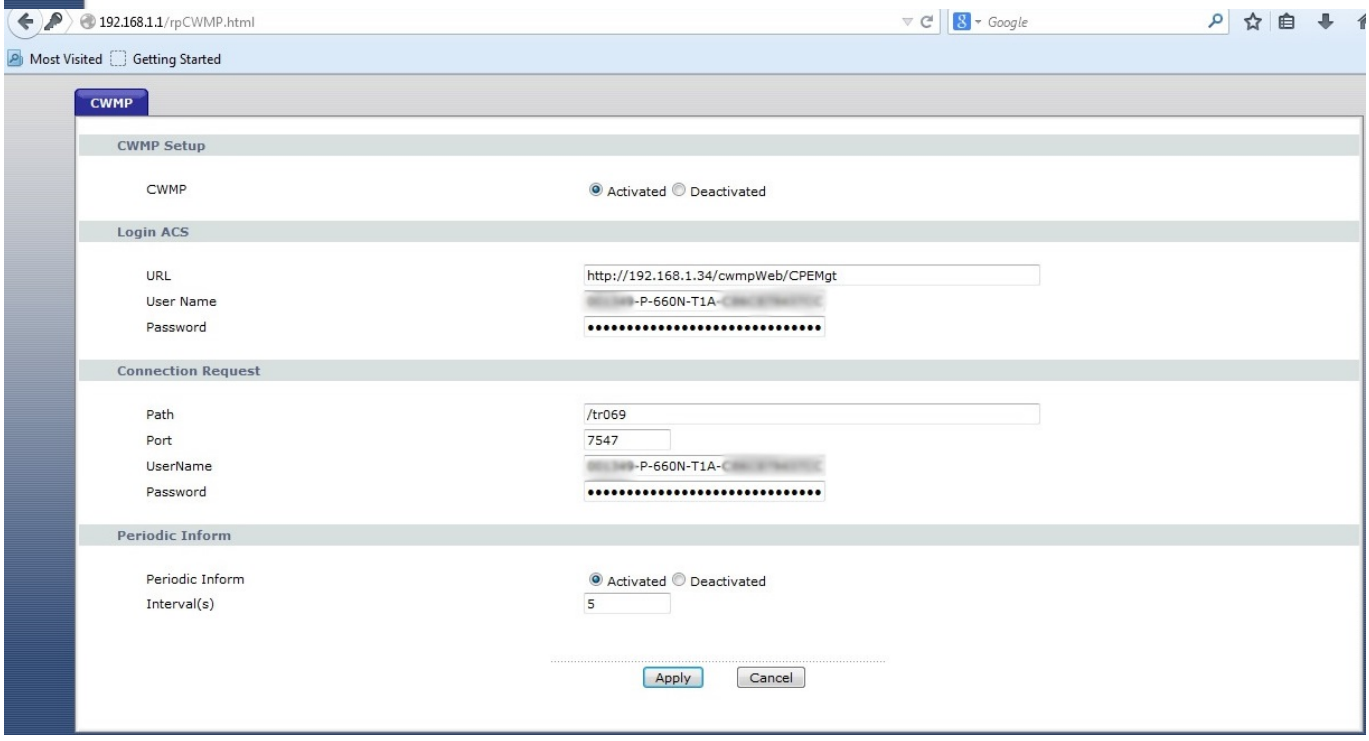

Applications Places
Fri Jun 20, 3:38 PM
root
root@kali: ~/Desktop
File Edit View Search Terminal Help
root@kali:~/Desktop# binwalk -e 340TSA5D0.bin
DECIMAL HEX DESCRIPTION
84992 0x14C00 ZynOS header, header size: 48 bytes, rom image type: ROMBIN, uncompressed size: 66696, compressed size: 16847, uncompressed checksum: 0xCB32, co
mpressed checksum: 0xD5A5, flags: 0xE0, uncompressed checksum is valid, the binary is compressed, compressed checksum is valid, memory map table address: 0x0
85043 0x14C33 LZMA compressed data, properties: 0x5D, dictionary size: 8388608 bytes, uncompressed size: 66696 bytes
128002 0x1F402 GIF image data, version 8"9a", 169 x 50
136194 0x21402 GIF image data, version 8"7a", 153 x 55
401408 0x62000 ZynOS header, header size: 48 bytes, rom image type: ROMBIN, uncompressed size: 5667036, compressed size: 1288801, uncompressed checksum: 0x7560
, compressed checksum: 0xF5E, flags: 0xE0, uncompressed checksum is valid, the binary is compressed, compressed checksum is valid, memory map table address: 0x0
401459 0x62033 LZMA compressed data, properties: 0x5D, dictionary size: 8388608 bytes, uncompressed size: 5667036 bytes
root@kali:~/Desktop# ls
14C33 14C33.7z 340TSA5D0.bin 62033 62033.7z
root@kali:~/Desktop#
KALI LINUX
The quieter you become, the more you are able to hear
[root@kali: ~]
root@kali: ~/Desktop

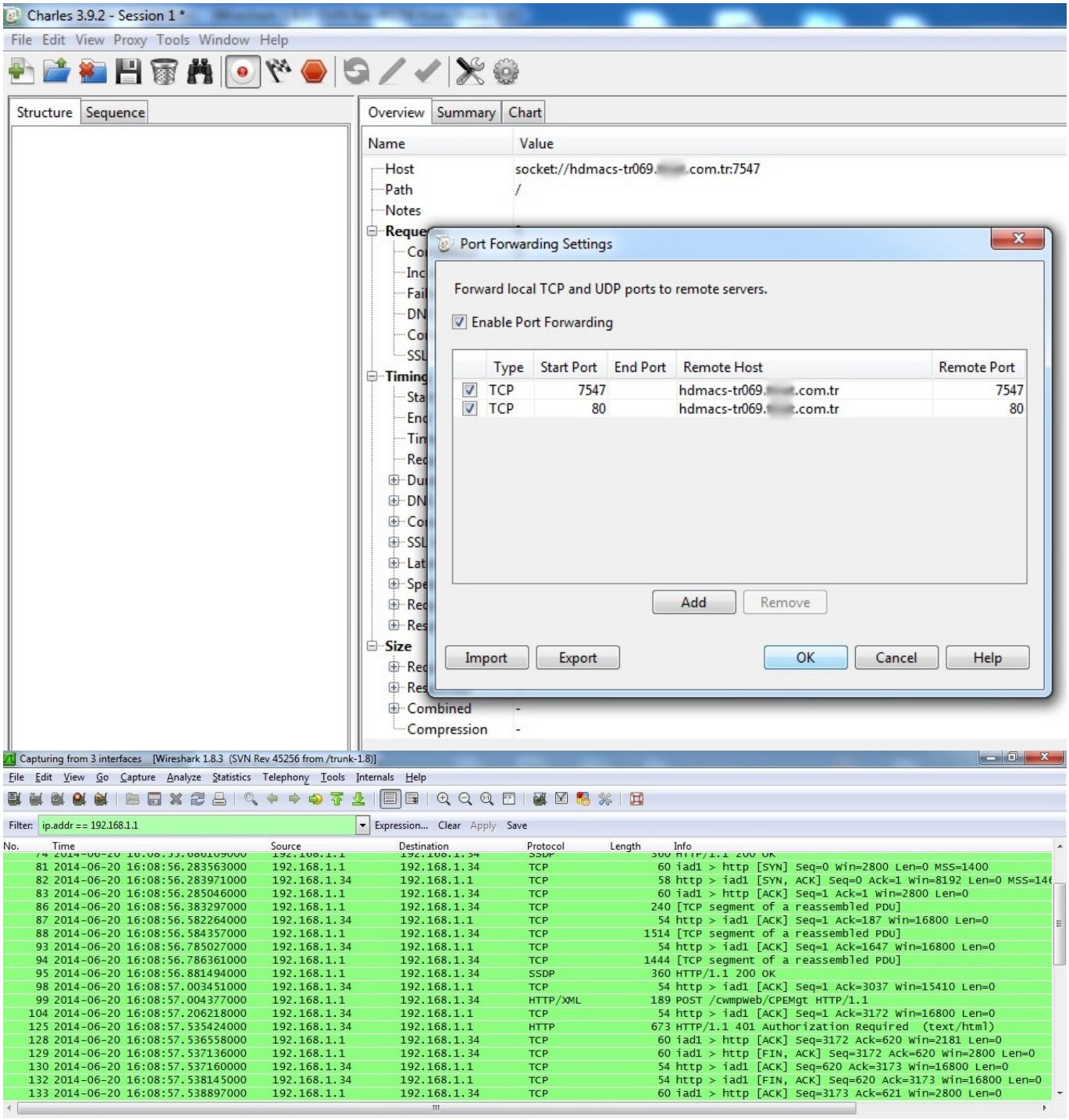

Applications Places
Fri Jun 20, 3:47 PM
root
root@kali: ~/Desktop
File Edit View Search Terminal Help
root@kali:~/Desktop# strings 62033 | grep -i .html
/rpFWUpload.html
/RestoreCfg.html
<html>
</html>
Rphttp.c---[RpBuildReply]--case eRpRomUrl---fHtmlResponseLength =
Rphttp.c---[RpSendReplyBuffer]---fHtmlResponsePtr =
text/html
<html>
text/html
/rpDslDisconnetWarn.html
">here</body></html>
<html>
text/html
/rpAutoFwupgrade[illegible].html
">here</body></html>
"text/html; charset=
h_SysAdmin.html
h_Ethernet.html
h_NAT_Mode.html
h_wzOthers.html
h_TimeZone.html
h_wzStatus.html
h_NAT_RuleList.html
h_NAT_RuleEdit.html
h_DHCP.html
h_Diag.html
h_UPNP.html
h_NAT_ServerEdit.html
h_Filter.html
h_Status.html
h_wzENET.html
h_FirmUp.html
h_wzDiag.html
h_WLAN_Setup.html
h_FirstPage.html
h_DiagGeneral.html
h_DiagDSL.html
h_wzPPP.html
h_wzPPPOE.html
h_wzRFC.html
h_DyDNS.html
h_RManage.html
KALI LINUX
The quieter you become, the more you are able to hear
[root@kali: ~]
root@kali: ~/Desktop
::: Web Tabanlı Yapıla...



rpCWMP.html dosyasının adından da anlaşılacağı üzere TR-069 yönetim protokolü ile ilgili ayarların yapıldığı sayfa olduğunu hemen anlayabiliriz. Sayfayı çağırdığımız zaman gelen internet servis sağlayıcısının Auto Configuration Servers (ACS) adresini, Charles Proxy aracının sistem üzerinde dinlediği adresi ve bağlantı noktası (port) ile değiştirip, Charles'a gelen istekleri de internet servis sağlayıcısının ACS adresine yönlendirdiğimizde, bu gizli sayfa sayesinde başarıyla ACS ile modem arasında gerçekleşen trafiği izleyebiliriz.

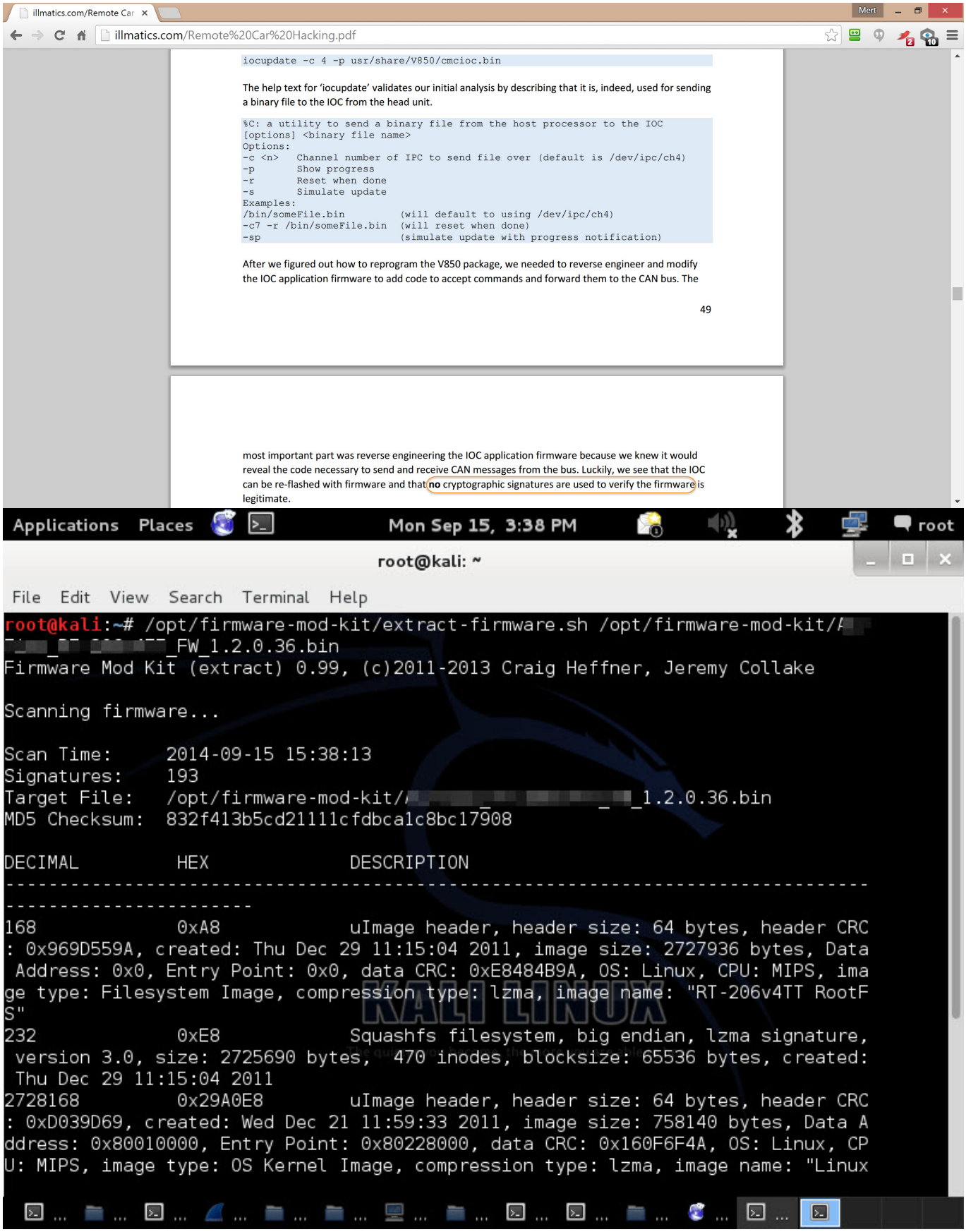

192.168.1.1/rpCWMP.html
Google
Most Visited
Getting Started
CWMP
CWMP Setup
CWMP
Activated
Deactivated
Login ACS
URL
http://hdmacs-tr069.
.com.tr/cwmpWeb/CPEMgt
User Name
-P-660N-T1A-
Password
Connection Request
Path
/tr069
Port
7547
UserName
-P-660N-T1A-
Password
Periodic Inform
Periodic Inform
Activated
Deactivated
Interval(s)
82400
Apply
Cancel

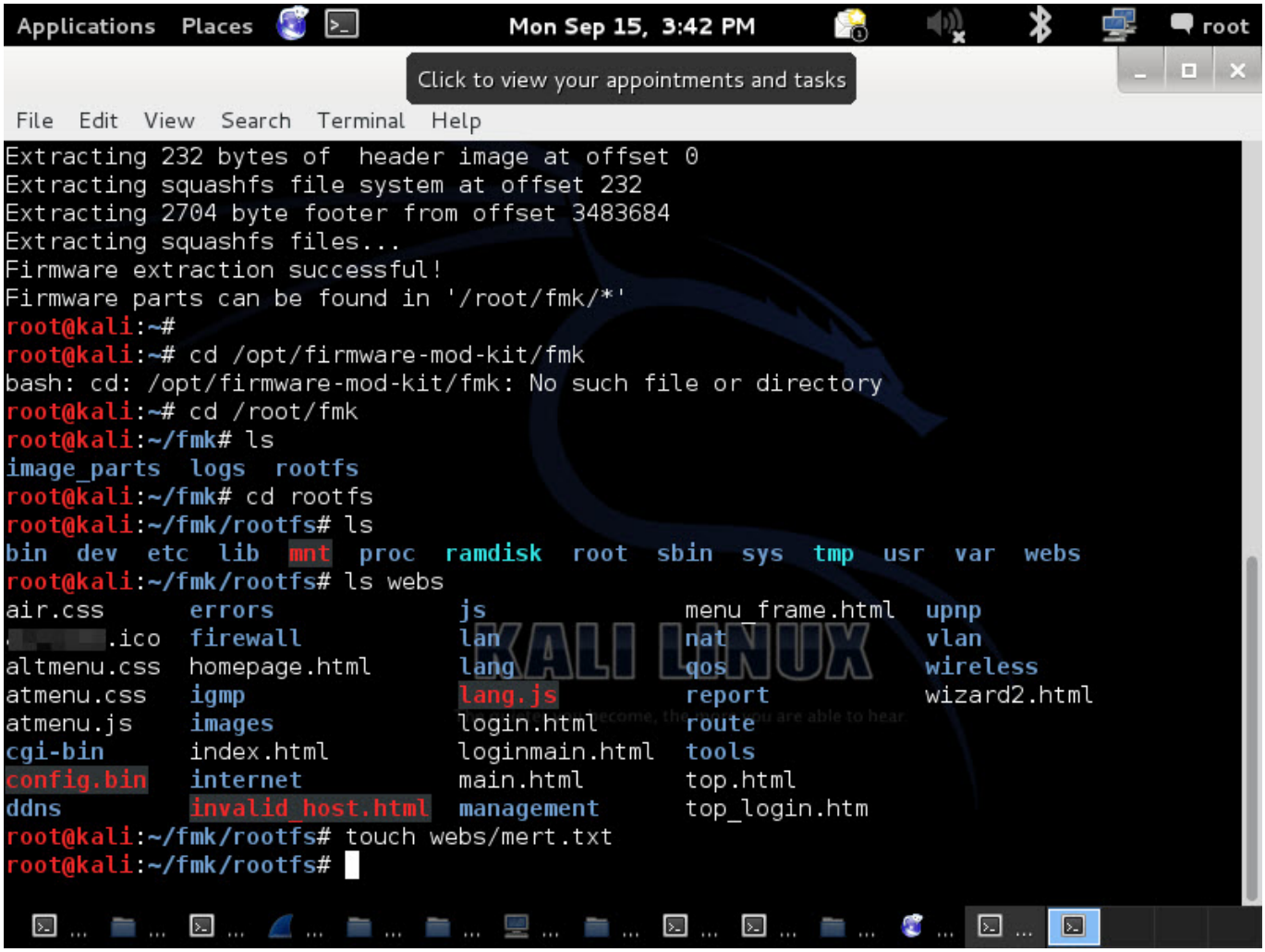

192.168.1.1/rpCWMP.html
Google
Most Visited
Getting Started
CWMP
CWMP Setup
CWMP
Activated
Deactivated
Login ACS
URL
http://192.168.1.34/cwmpWeb/CPEMgt
User Name
-P-660N-T1A-
Password
Connection Request
Path
/tr069
Port
7547
UserName
-P-660N-T1A-
Password
Periodic Inform
Periodic Inform
Activated
Deactivated
Interval(s)
5
Apply
Cancel

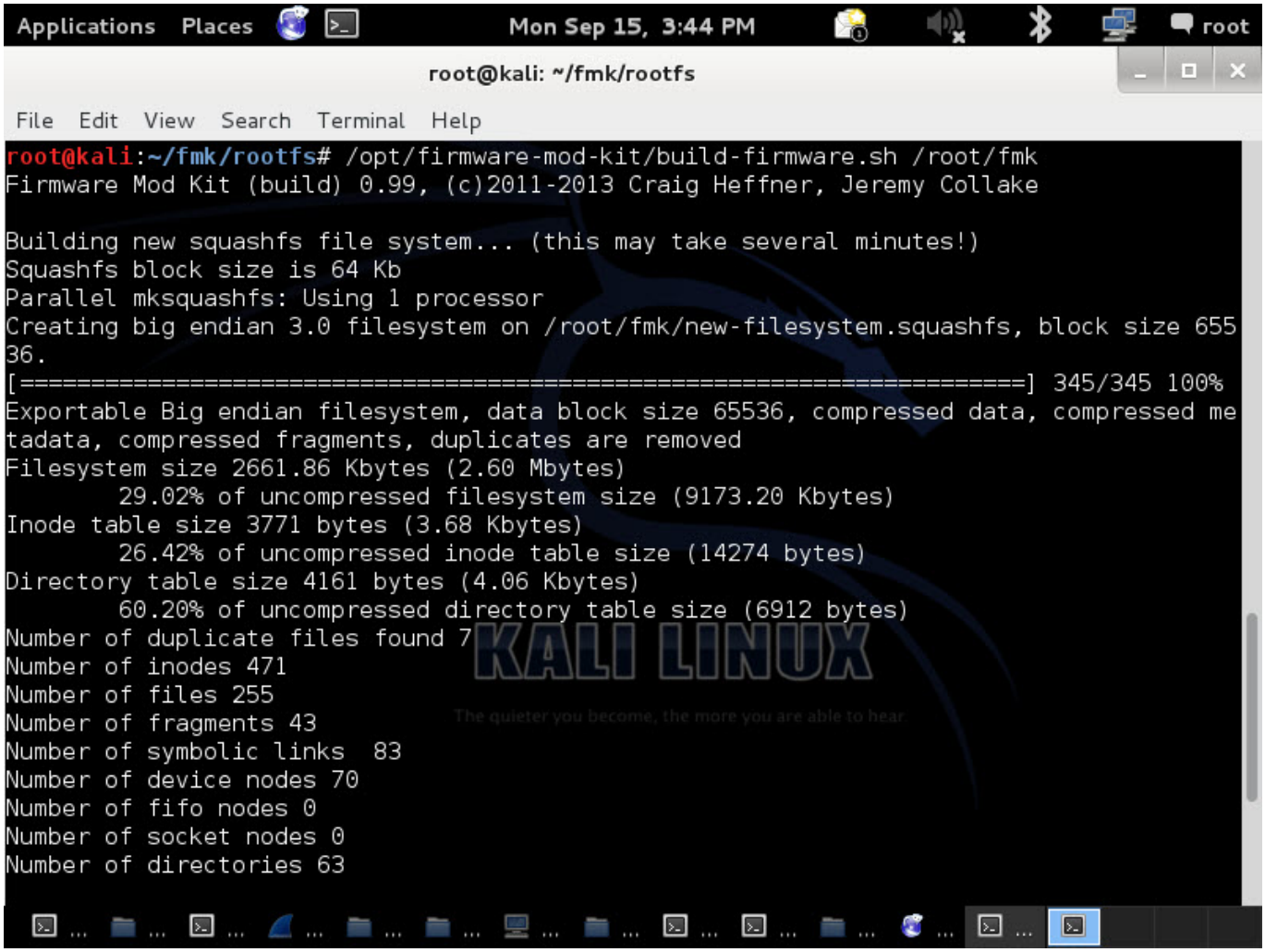


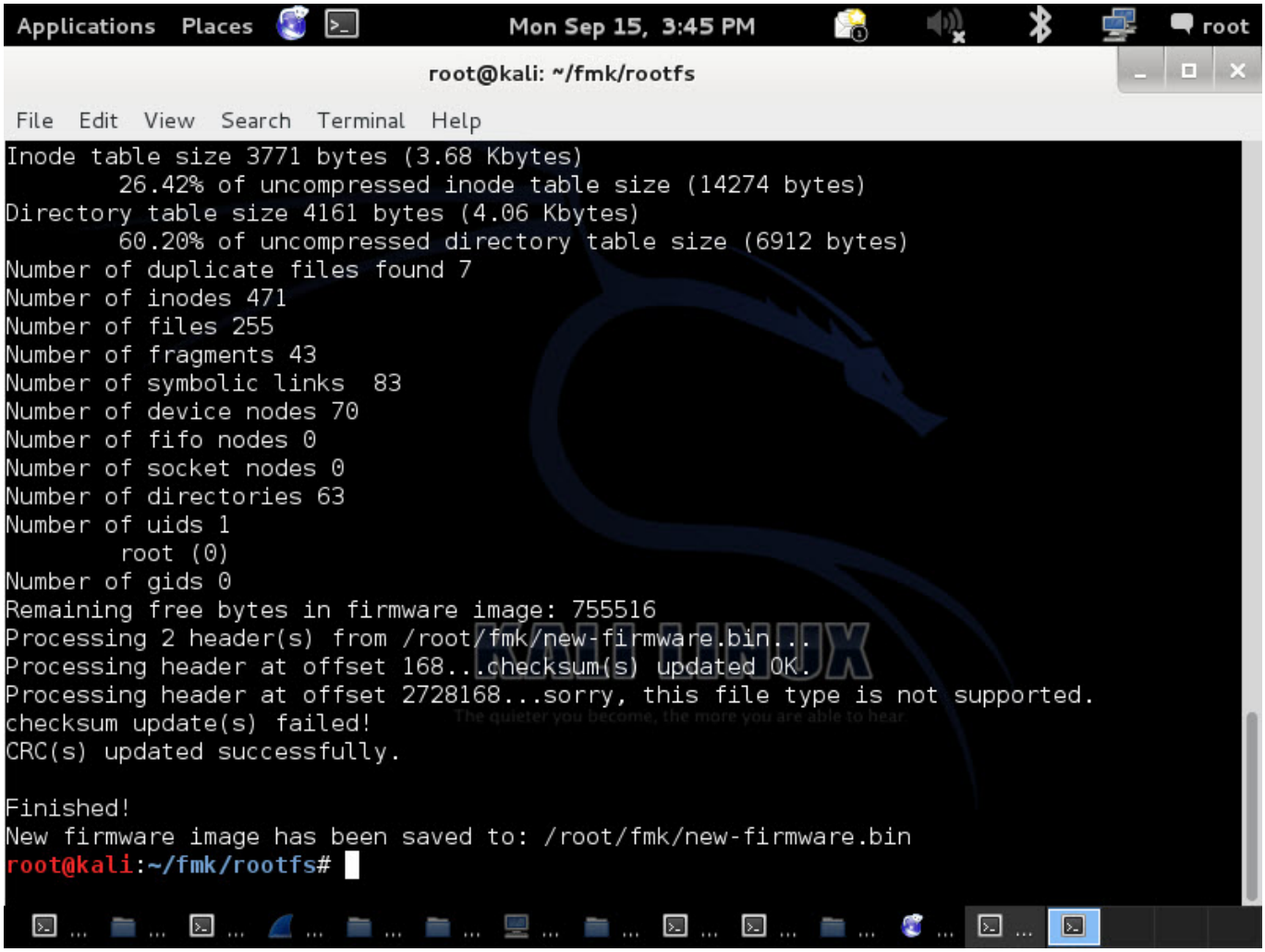
Peki donanım yazılımını analiz ettikten sonra Charlie Miller ile Chris Valasek'in Cherokee Jeep'i hacklerken yaptıkları gibi donanım yazılımını manipüle edip (patching) cihaza yüklemek istersek, donanım yazılımını tekrar paketlemek için faydalabileceğimiz Firmware Modification Kit (FMK) aracından da kısaca bahsetmek gerekir. İlgili donanım yazılımının dosya sisteminde yer alan dosyalarını, FMK'da yer alan extract-firmware.sh betiği ile diske açtıktan ve değişiklikler yaptıktan sonra yine aynı araçta yer alan build-firmware.sh betiği ile paketlememiz mümkün. Bu sayede hedef cihazın imza

kontrolü yapmadan donanım yazılımını güncellemeye izin verip vermediğini de kolaylıkla kontrol edebiliriz.

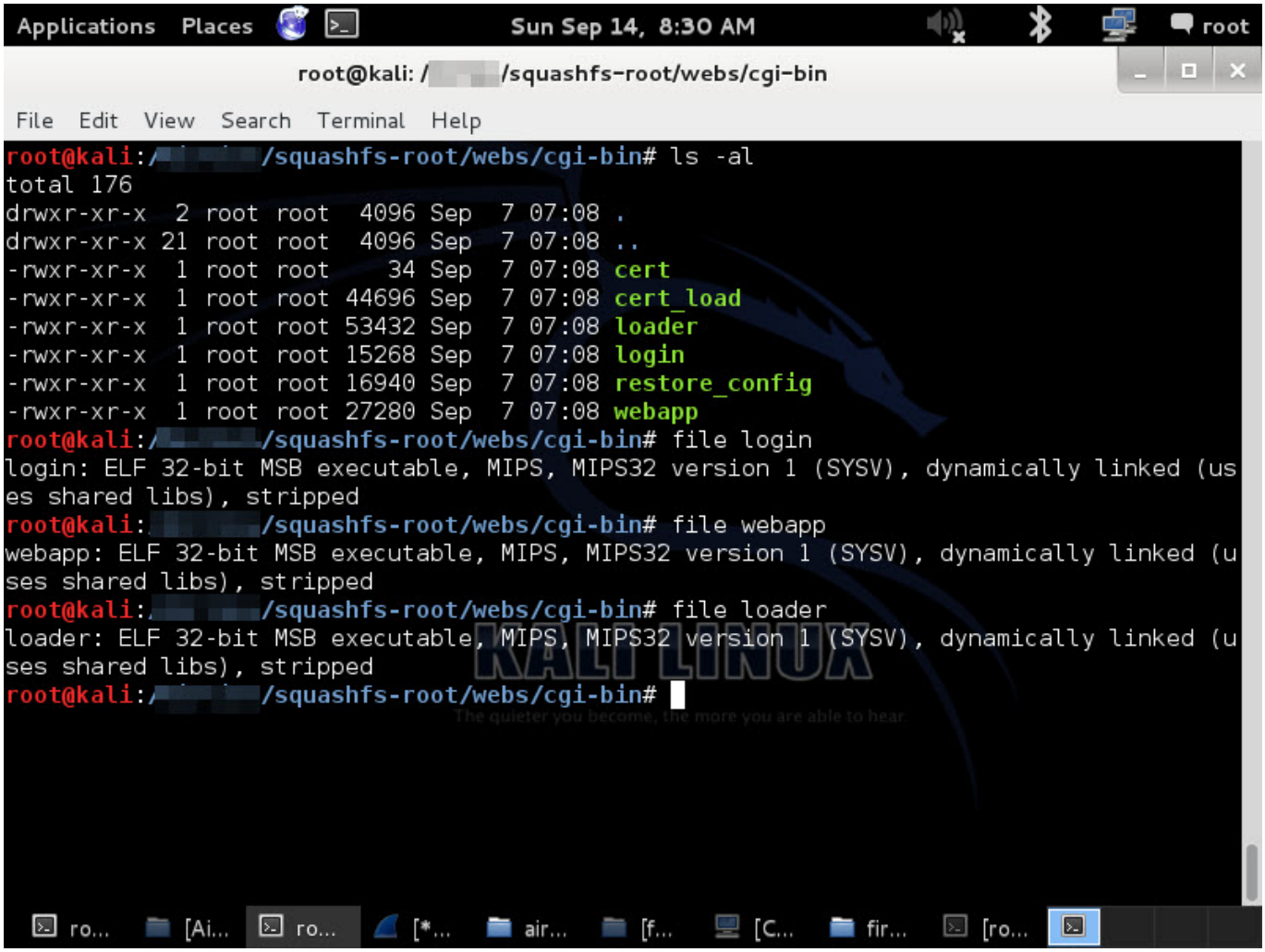


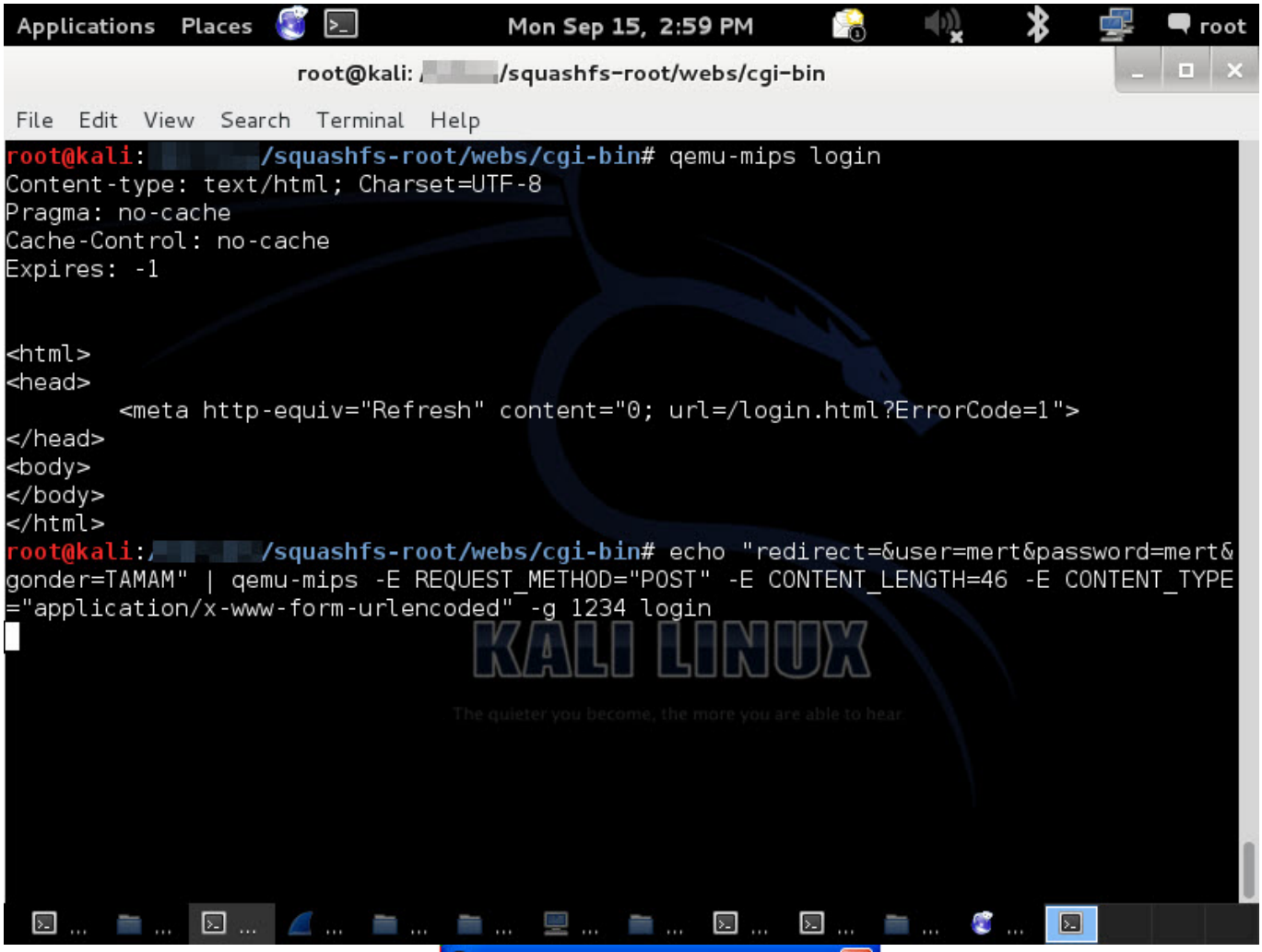

Applications Places
Mon Sep 15, 3:42 PM
root
Click to view your appointments and tasks
File Edit View Search Terminal Help
Extracting 232 bytes of header image at offset 0
Extracting squashfs file system at offset 232
Extracting 2704 byte footer from offset 3483684
Extracting squashfs files...
Firmware extraction successful!
Firmware parts can be found in '/root/fmk/*'
root@kali:~#
root@kali:~# cd /opt/firmware-mod-kit/fmk
bash: cd: /opt/firmware-mod-kit/fmk: No such file or directory
root@kali:~# cd /root/fmk
root@kali:~/fmk# ls
image_parts logs rootfs
root@kali:~/fmk# cd rootfs
root@kali:~/fmk/rootfs# ls
bin dev etc lib mnt proc ramdisk root sbin sys tmp usr var webs
root@kali:~/fmk/rootfs# ls webs
air.css errors js menu_frame.html upnp
.ico firewall lan nat vlan
altmenu.css homepage.html lang qos wireless
atmenu.css igmp lang.js report wizard2.html
atmenu.js images login.html route
cgi-bin index.html loginmain.html tools
config.bin internet main.html top.html
ddns invalid_host.html management top_login.htm
root@kali:~/fmk/rootfs# touch webs/mert.txt
root@kali:~/fmk/rootfs#

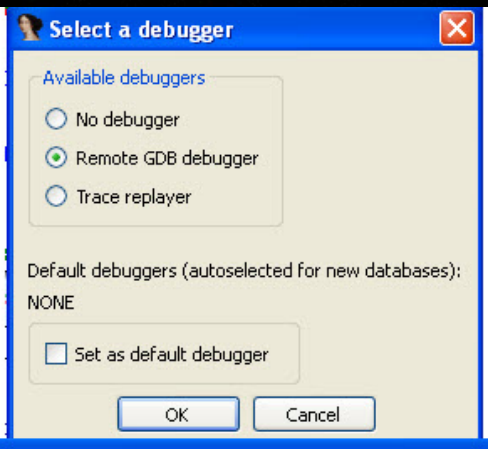

Applications Places Mon Sep 15, 3:44 PM root
root@kali: ~/fmk/rootfs
File Edit View Search Terminal Help
root@kali:~/fmk/rootfs# /opt/firmware-mod-kit/build-firmware.sh /root/fmk
Firmware Mod Kit (build) 0.99, (c)2011-2013 Craig Heffner, Jeremy Collake
Building new squashfs file system... (this may take several minutes!)
Squashfs block size is 64 Kb
Parallel mksquashfs: Using 1 processor
Creating big endian 3.0 filesystem on /root/fmk/new-filesystem.squashfs, block size 655
36.
[==========] 345/345 100%
Exportable Big endian filesystem, data block size 65536, compressed data, compressed me
tadata, compressed fragments, duplicates are removed
Filesystem size 2661.86 Kbytes (2.60 Mbytes)
29.02% of uncompressed filesystem size (9173.20 Kbytes)
Inode table size 3771 bytes (3.68 Kbytes)
26.42% of uncompressed inode table size (14274 bytes)
Directory table size 4161 bytes (4.06 Kbytes)
60.20% of uncompressed directory table size (6912 bytes)
Number of duplicate files found 7
Number of inodes 471
Number of files 255
Number of fragments 43
Number of symbolic links 83
Number of device nodes 70
Number of fifo nodes 0
Number of socket nodes 0
Number of directories 63
KALI LINUX
The quieter you become, the more you are able to hear

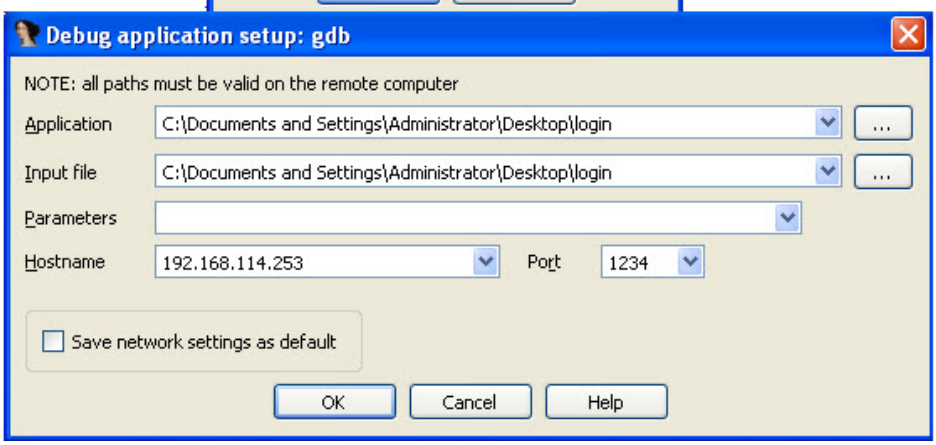


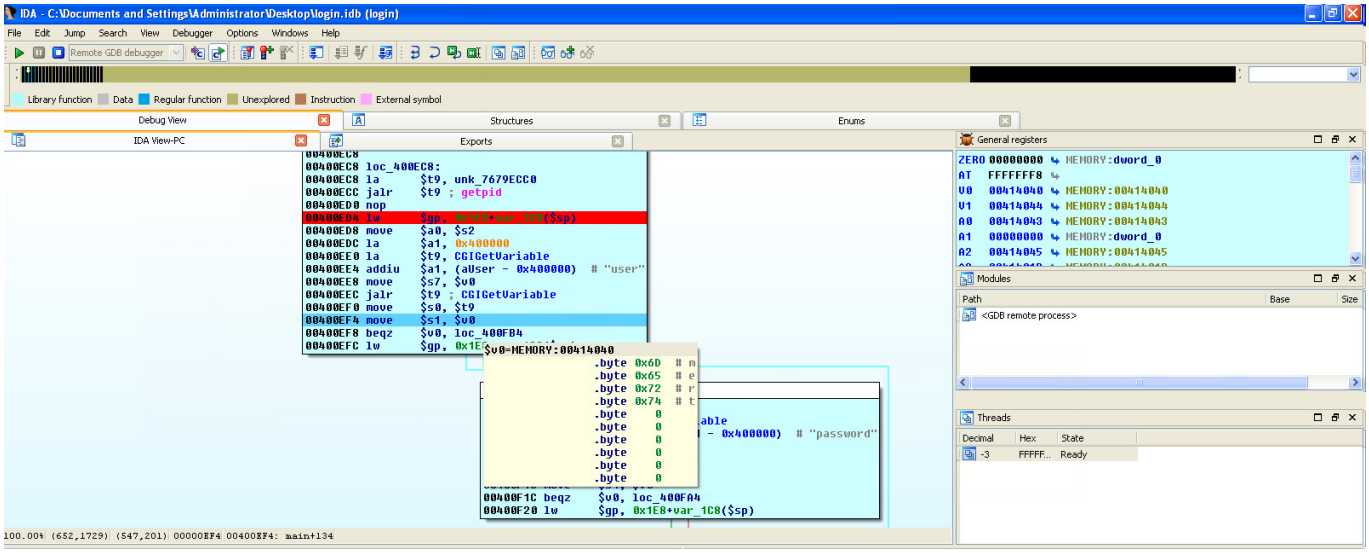
Statik olarak değil de dinamik olarak donanım yazılımında yer alan programları teker teker analiz etmek istiyoruz dersek de o zaman, QEMU öykünücüsü (emulator) ve IDA Pro aracı sayesinde aşağıdaki ekran görüntüsülerinde yer aldığı şekilde programları (örnek: login) detaylı bir şekilde analiz edebiliriz.


Applications Places Sun Sep 14, 8:30 AM root
root@kali: /squashfs-root/webs/cgi-bin
File Edit View Search Terminal Help
root@kali:/squashfs-root/webs/cgi-bin# ls -al
total 176
drwxr-xr-x 2 root root 4096 Sep 7 07:08 .
drwxr-xr-x 21 root root 4096 Sep 7 07:08 ..
-rwxr-xr-x 1 root root 34 Sep 7 07:08 cert
-rwxr-xr-x 1 root root 44696 Sep 7 07:08 cert_load
-rwxr-xr-x 1 root root 53432 Sep 7 07:08 loader
-rwxr-xr-x 1 root root 15268 Sep 7 07:08 login
-rwxr-xr-x 1 root root 16940 Sep 7 07:08 restore_config
-rwxr-xr-x 1 root root 27280 Sep 7 07:08 webapp
root@kali:/squashfs-root/webs/cgi-bin# file login
login: ELF 32-bit MSB executable, MIPS, MIPS32 version 1 (SYSV), dynamically linked (uses shared libs), stripped
root@kali:/squashfs-root/webs/cgi-bin# file webapp
webapp: ELF 32-bit MSB executable, MIPS, MIPS32 version 1 (SYSV), dynamically linked (uses shared libs), stripped
root@kali:/squashfs-root/webs/cgi-bin# file loader
loader: ELF 32-bit MSB executable, MIPS, MIPS32 version 1 (SYSV), dynamically linked (uses shared libs), stripped
root@kali:/squashfs-root/webs/cgi-bin#
KALI LINUX
The quieter you become, the more you are able to hear


Applications Places Mon Sep 15, 2:59 PM root
root@kali: /squashfs-root/webs/cgi-bin
File Edit View Search Terminal Help
root@kali: /squashfs-root/webs/cgi-bin# qemu-mips login
Content-type: text/html; Charset=UTF-8
Pragma: no-cache
Cache-Control: no-cache
Expires: -1
<html>
<head>
<meta http-equiv="Refresh" content="0; url=/login.html?ErrorCode=1">
</head>
<body>
</body>
</html>
root@kali: /squashfs-root/webs/cgi-bin# echo "redirect=&user=mert&password=mert&
gonder=TAMAM" | qemu-mips -E REQUEST_METHOD="POST" -E CONTENT_LENGTH=46 -E CONTENT_TYPE
="application/x-www-form-urlencoded" -g 1234 login
KALI LINUX
The quieter you become, the more you are able to hear


Select a debugger
Available debuggers
No debugger
Remote GDB debugger
Trace replayer
Default debuggers (autoselected for new databases):
NONE
Set as default debugger
OK
Cancel


Debug application setup: gdb
NOTE: all paths must be valid on the remote computer
Application C:\Documents and Settings\Administrator\Desktop\login
Input file C:\Documents and Settings\Administrator\Desktop\login
Parameters
Hostname 192.168.114.253
Port 1234
Save network settings as default
OK
Cancel
Help



Çok daha fazlasını öğrenmek ve uygulamak istiyorum diyenleriniz için ise Abusing the Internet of Things: Blackouts, Freakouts, and Stakeouts kitabına bir göz atmalarını tavsiye edebilirim.

Bir sonraki yazıda görüşmek dileğiyle herkese güvenli günler dilerim.