# Global Information Assurance Certification Paper

# Securing the Home IoT Network

*GIAC (GSEC) Gold Certification*

Author: Manuel Leos Rivas, MLeosRivas@mastersprogram.sans.edu
Advisor: Adam Kliarsky
Accepted: March 22nd 2017

Template Version September 2014

Abstract

The Internet of Things (IoT) has proven its ability to cause massive service disruption because of the lack of security in many devices. The vulnerabilities that allow those denial of service attacks are often caused due to poor or no security practices when developing or installing the products. The common home network is not designed to protect against the design errors in IoT devices that expose the privacy of the users. The affordable price of single board computers (SBC) and their small power requirements and customization capabilities can help improve the protection of the home IoT network. SBC can also add powerful features such as auditing, inspection, authentication, and authorization to improve controls pertaining to who and what can have access. Implementing a home-control gateway when properly configured reduces some common risks associated with IoT such as vendor-embedded backdoors and default credentials. Having an open source trusted device with a configuration shared and audited by many experts can reduce many of the bugs and misconfigurations introduced by vendor security program deficiencies.

# 1. Introduction

In the last few years, the internet bandwidth available to home users has increased dramatically from slow connections to reliable high-speed internet connections, while the price/speed ratio continues to decrease (Patrick Eha, 2013). The speed increases at a 50% rate every year, which is slightly slower to the 60%-increase rate in computer power (Nielsen, 1998). There have also been reductions in the size of electronic devices, and many previously simple devices have evolved to contain computers and microcontrollers to provide a new set of services for communication.

The increased number of such elements, its capabilities, and the need from vendors to reduce costs, have introduced new risks into an environment where the users have, in most cases, little to no expertise assessing and managing those risks.

Home networks are built for commodity and ease of use, leaving privacy and security as lesser concerns, leading to nonsecured configurations under the wrong premise that the home network is accessible only to trusted users with no intentions to do any harm. Some attacks abuse this premise by pivoting on the client to further compromise the home network or use its resources.

Usually, home networks have a single small firewall or router (Mitchell, 2016) owned by the internet provider in many countries. Those units are produced by the millions and may be managed remotely. The firewall provides network address translation to access the web on a single flat 802.11b/g/n/ac wireless network. In addition, some of the devices produced recently have incorporated a few improvements such as a demilitarized (DMZ) network and parental controls (Mitchell, 2016).

The common devices in these networks are smartphones, tablets, laptops, gaming consoles, cameras, DVRs, smart TVs, VoIP phones, and, more recently, sensors of all kinds, such as home appliances and alarms.

Home devices and networks are unmonitored, uncontrolled, have no or weak access controls, and accept connections by default from within the same network.

# 2. The typical home network

Many providers offer fiber optic or DSL internet connections, and in some cases, a modem router is the only unit to provide access to the web, while in others cases, an Ethernet port is delivered from the device and a commercial home router unit is connected to provide the access. Last decade devices provided no or limited connectivity options, while most recent devices provide at least four fast or gigabit Ethernet ports and 802.11b/g/n/ac wireless single or dual band access (Delaney, 2017). Wireless security by default has improved in the last few years using WPA2 and WPS; previously, it was common to have open or WEP wireless networks.

However, some default settings opens the home network to attacks coming from the Internet, or the internal network, and the most exposed elements in the network are those connected in the so-called DMZ network and the many devices with no security protections, default passwords, or with vendor hidden backdoors or vulnerabilities. Once compromised, any element in the network may be used to pivot and attack the internal network, connect to the internet, join a botnet, abuse the resources of the available network, steal data, or spy on the user.

## 2.1. Common home network devices

Current networks at home are full of devices that may use the network to access the Internet or connect only locally to provide some service. Smartphones, tablets, and laptops use many applications and protocols, connect with many different networks for email access and web browsing, and are nomadic by nature. Video cameras for surveillance and baby monitors and DVRs are accessible locally or from the internet and use protocols such as HTTP and RTSP (Schulzrinne, 1998). Gaming consoles use multiple TCP and UDP ports for connecting to game servers, playing online, streaming audio and video, sending and receiving emails, and web browsing. Devices such as chrome cast, Roku, iTV, firestick, and smartTVs use several protocols such as HTTP. SmartLights, smartThings, sensors, meters, and DIY devices use HTTP and MQTT protocols and may be accessible locally or may connect to remote locations over the Internet, while DHCP and DNS services are used by most of the devices mentioned. In

addition, some other home devices may use other network types such as 802.15.4 and Bluetooth.

## 2.2. Architecture

The typical home network has a separate interface for the WAN network and a single network switch to connect the entire LAN network, as shown in Figure 1, but not all the devices are built the same. For example, some home routers use a DMZ host or network feature. DMZ for home routers vendor definition is "a host on the internal network that has all UDP and TCP ports open and exposed, except those ports otherwise forwarded" (TP-link, 2016). There are home routers that have all WAN and LAN ports directly interconnected at network layer two as the system-on-a-chip (SoC) has a single RGMII (switch-chip), the SoC separates the ports traffic by using VLANs only (Banana-pi, 2016/7) which is ok for switches but not for routers.

**Figure 1: Typical home network architecture**

# 3. Home network and IoT exploitation

To be assumed safe the internal home network should only have trusted systems and no malicious activities or attacks from one system to another. Some people or vendors made that assumption while designed their networks or products. That sense of absolute trust may lead to a severe lack of security controls in home networks.

## 3.1. Weak or hardcoded credentials

The idea that only trusted elements exist in the internal home network has made some vendors and users use weak default passwords with the thinking that no one else but the legitimate user will ever have access to those systems. During the last few years, the leaked credentials from hacked sites show that the common passwords in Figure 2 have been used by millions of users:

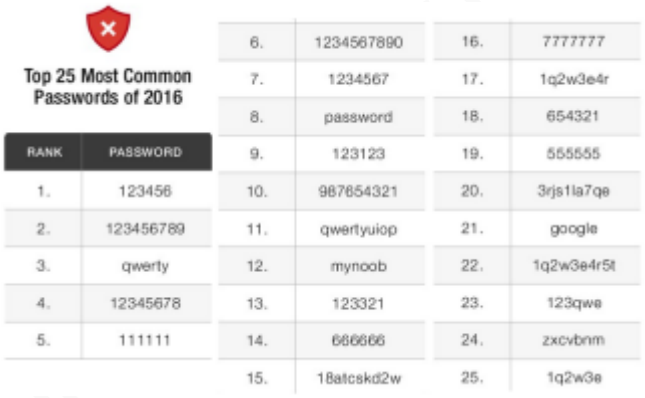


**Figure 2: Top worst passwords list 2016 (Keeper, 2017)**

Default or hardcoded credentials in common devices place the end user at risk many times without his or her knowledge, and sometimes, it is not possible to change them in some of the devices.

"Mirai" malware abused of multiple default credentials to perform attacks in 2016 that caused severe network disruptions on the internet, the passwords used by the malware at that time are shown in Figure 3.

| Username/Password | | |
|---|---|---|
| | root/zlxx | root/00000000 |
| admin/123456 | root/juantech | root/realtek |
| root/anko | root/xc3511 | admin/1111111 |
| root/pass | root/hi3518 | root/xmhdipc |
| root/vizxv | root/klv123 | admin/smcadmin |
| root/888888 | root/klv1234 | root/ikwb |
| root/666666 | root/jvbzd | ubnt/ubnt |
| root/7ujMko0vizxv | root/admin | supervisor/supervisor |
| root/7ujMko0admin | root/system | root/<none> |
| 666666/666666 | admin/meinsm | admin/1111 |
| root/dreambox | root/54321 | root/Zte521 |

**Figure 3: Common device passwords used by "Mirai" malware (Krebs, 2016)**

The passwords listed in "Mirai" source code belonged to internet connected devices like IP cameras, DVRs, and home routers. Malware as "Nya" first attempted to connect directly to the devices over simple protocols such as telnet with common credentials to then start the infection either by executing commands directly or planting back door binaries (MalwareMustDie, 2016).

## 3.2. Design flaws

Another vulnerability present in home devices is not checking whether a call to a resource was performed by a user surfing the management interface or by an external entity. This particular flaw is abused by cross-site request forgery attacks by hiding a script or a link concealed in a web page or email. This tricks the user browser or client to execute the code and to perform a call over the local network to the vulnerable device. The client then performs actions such as enabling services, opening ports, enabling remote administration over the Internet, opening a connection back to the attacker, and any other action available on the administration interface.

Home routers have other design flaws, such as the one found in the Zyxel and other brands TR-69 management protocol that allowed unauthenticated remote command execution. This vulnerability present in devices distributed by a large ISP in Europe left several hundred thousand DSL modems exposed to attacks (Goodin, 2016).

Universal Plug and Play (UPnP) protocol can also be abused to open communication channels from the Internet to the "isolated" internal network by allowing the automatic configuration without human intervention. If the DMZ feature is enabled on the router, then UPnP can provide free access to an internal host.

## 3.3. Backdoors

Some insecure development practices of vendors cause vulnerabilities. For example, the practice of leaving some backdoors open during the early testing phase. Backdoors are intended for expediting the troubleshooting, but they are sometimes forgotten and make their way to the production releases of the product. Some of these have been detected by using the common password lists, brute force, or by reverse engineering the firmware.

The firmware in most devices is not obfuscated or encrypted so that only the intended device can read it; consequently, once a firmware copy is obtained, information such as the operating system, files, hardcoded credentials and hashes, keys, and source code of scripts and documents can be extracted. The SEC Consult Security firm used was able to identify vulnerabilities in multiple IP cameras (Consult, 2016) and even developed tools to detect them.

## 3.4. Attacks

Once hacked, vulnerable software and devices can be used in many ways as part of a botnet. Some common attacks are sending spam, relaying or proxy traffic, DDoS, stealing account and banking credentials, bitcoin mining, distributing or spreading malware, phishing, storing piracy or porn, spying on the user and network activities, pivoting to different hosts and networks, or encrypting the accessible data to then ask for ransom.

### 3.4.1. Denial of service

The infrastructure layer accounts for most of the distributed denial of service attacks, as shown in Figure 4. The application layer denial of service attacks are less common than the often "point and click" infrastructure layer attacks. "While automation and bot programs can carry out many attack tasks, web application attacks are mainly initiated and monitored by humans" (McKeay, 2016).

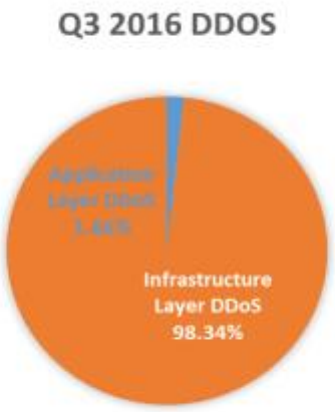


**Figure 4: Akamai DDoS mitigated attacks (McKeay, 2016).**

Using IoT devices to attack other systems can produce an overwhelming amount of traffic with a current record of a magnitude of 1.2 Tbps, as the attacks were generated by several hundred thousand compromised devices from "Mirai"- based botnets. One of the victims of DDoS attacks that caused severe service disruptions to many companies was against Dyn- managed DNS infrastructure (Hilton, 2016). This DDoS attack resulted in many companies lossing internet connectivity. More unsecured devices and faster consumer networks produce larger attacks. In late 2016, the "Mirai" source code was made available, as shown in Figure 5 below. Since then it has been used in other DDoS attacks.

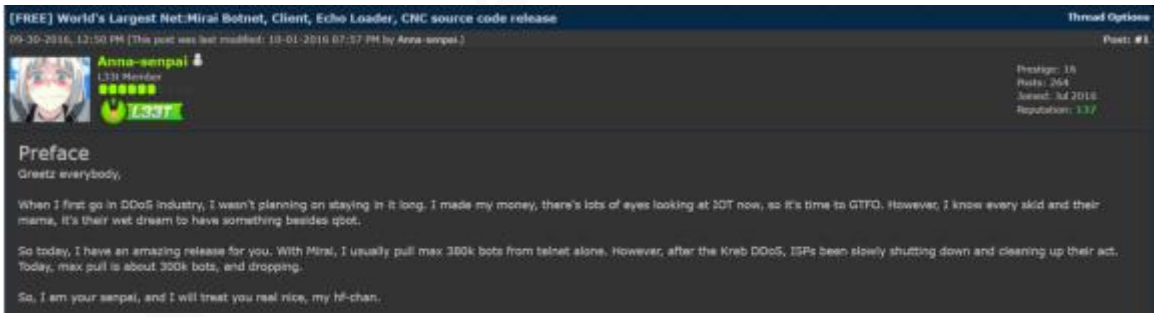


**Figure 5: "Mirai" source is available to the public since Sep, 2016 (Jelic, 2016).**

### 3.4.2. Spying

Nowdays, HTTPS usage is more common than HTTP, and the trend is to reduce the usage of the former (Google, 2017). Many of the major websites do automatic forwarding to the encryption protected version of the site. Web browsers are also pushing to use secure encryption by adding visual warnings to let the user know his or her data may be at risk. Clear text HTTP messages can be read by anyone on the path between the exchanging endpoints, as shown in Figure 6. Often HTTP cannot detect if the data exchanges have been manipulated.


**Figure 6: Clear text data visibility in HTTP.**

Encryption does not prevent eavesdropping from ever happening. Depending on many factors, the data may be exposed as some of the major issues with HTTPS are related to the certificate authorities trusted by the endpoints, as shown in Figure 7. If a computer vendor, company, or CA leaks its PKI key, it will put all systems that trust them at risk of impersonation and Man-In-The-Middle (MITM) attacks.


**Figure 7: Clear text data visibility in HTTPS.**

Another means to obtain private data is by accessing it directly from the endpoint, as malware running on the system can read it from memory or storage.

In the IoT- connected world, many devices may expose the user privacy, and if they are not protected, game consoles, televisions, cameras and baby monitors can send archived or real-time audio and video feeds over the internet.

### 3.4.3. Man in the middle

Using encryption makes the MITM attacks harder but not impossible, and this kind of attack can be performed by different means and scope.

Manipulating the manner that the network handles the traffic with ARP or route poisoning, as shown in Figure 8, requires direct access to the network path (any network between source and destination) and only those flows passing through it are exposed.

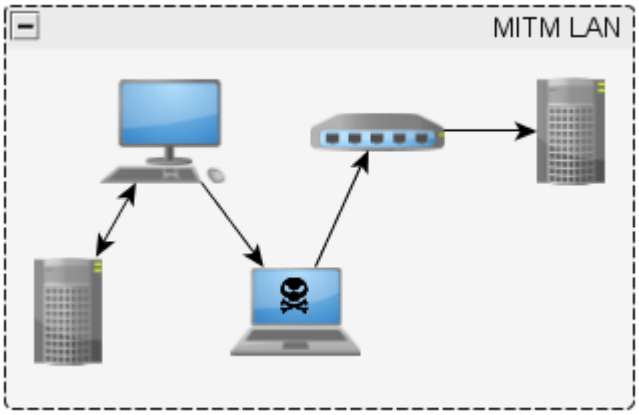


**Figure 8: ARP or routing attack.**

DNS attacks on the other hand as shown in Figure 9, have a broader scope as they affect all users of a given server or the entire service. The attacks can be at the cache or resource record hijacking. The first type may be limited to a company or provider, while the second affects every system or user using DNS to access that resource.

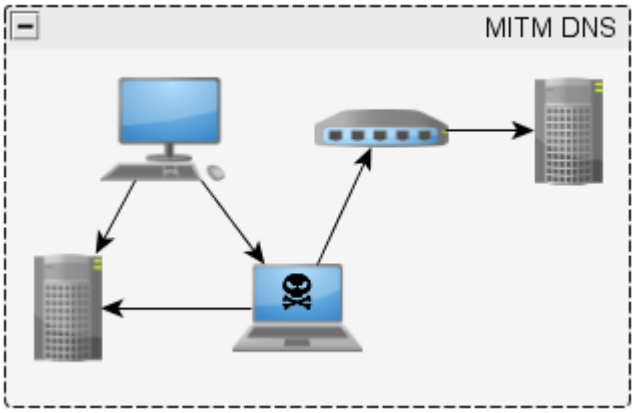


**Figure 9: DNS cache or resource record hijacking attack.**

The web browser can also play a role and trick the regular data flow to spy and tamper it. The browser MITM attack is local to the affected client and is usually performed by malware, as shown in Figure 10.

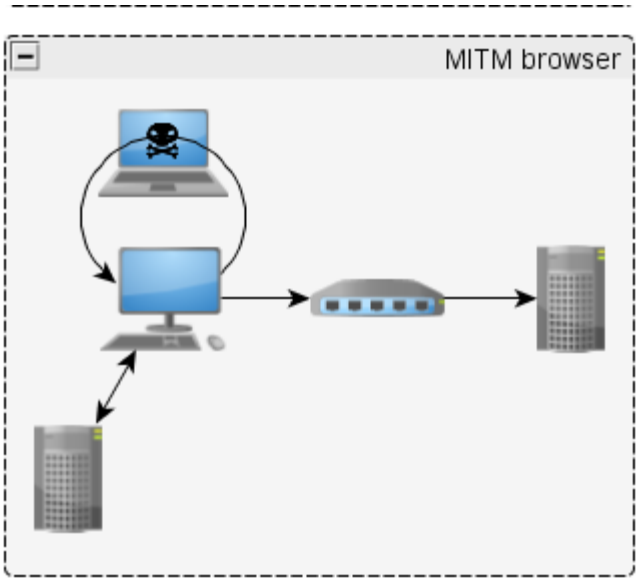


**Figure 10: Local man in the browser attack**

## 4. Defending the home and IoT network

Protecting and managing the home network must not be complicated and must not require unreasonable amounts of time, knowledge, or money to manage it. The controls deployed in the network should increase the security and privacy levels but may not reduce the performance of the commodity.

A vulnerability on the home router exposes the entire network to attacks. The same device manages all the basic network services such as DHCP, DNS, and routing required to maintain the internet service, as shown in Figure 11. In many cases, there are DMZ or port- forwarding features enabled to allow the user to have access to the network for gaming, personal servers, sensor monitoring, and IP cameras.

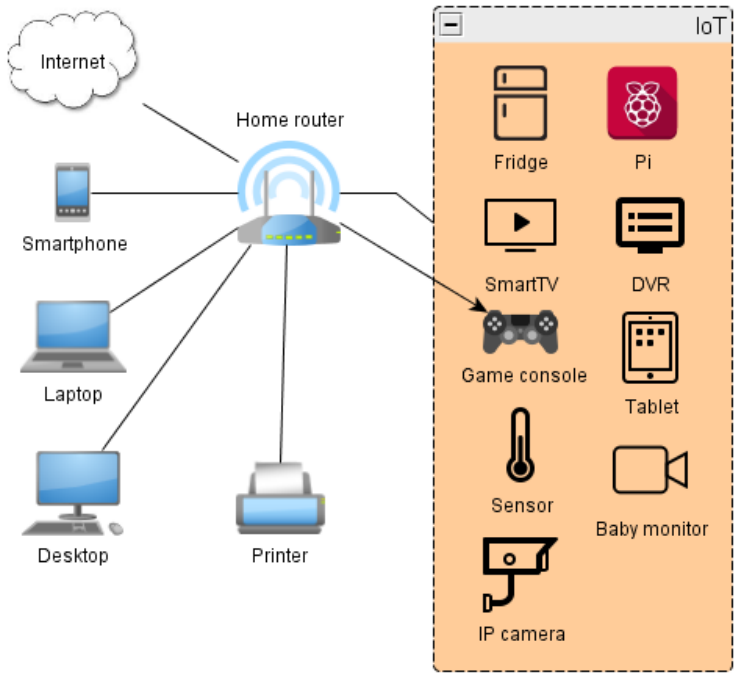


**Figure 11: Flat Wi-Fi or wired network with a single segment and game console in DMZ.**

## 4.1. Architecture

Architecture plays a major role in network security, the traditional single segment flat network enables all devices to have unrestricted access to each other and does not provide the appropriate protection to those items that are more sensitive. Further segmenting the network by using firewalls with appropriate access control to isolate the sensitive devices can limit the data exchanges to those that are strictly required.

Connecting a new home router cascading from the already existing router using the WAN port creates an entirely different LAN; this new segment should be used to connect all the sensitive devices and those that need to interact with each other like a device used to store video files and a television, but that seldom use the Internet. Any item that cannot be updated, uses default passwords, lacks access controls, or secure communication channels should be considered sensitive. IP cameras, DVRs, sensors, baby monitors, printers, are excellent candidates of this segment.

The most basic configuration should include at least a regular COTS home router; this may be the easier to configure, and it should preferably be from a different brand

than the previous router. Using different brands reduces the risk of having the same vulnerability in both, and replacing the firmware with DD-WRT in most cases can add more features. DD-WRT is a Linux- based open source firmware that has an easy-to-use interface and some features (embeDD GmbH, 2017). Most home routers enable only one additional segment which will limit how a network can be segregated in different zones depending on sensitivity and function.

Another alternative to use a COTS router is to use an old or small computer with two or more network adapters or a single board computer (SBC) with multiple USB dongles to perform the routing between multiple network segments to provide more granular control on the network. One PC or SBC with three networks creates two additional segments, one for IoT that requires many local and cloud connections to work and the other for those more sensitive in nature or that require more strict access control due to privacy or vulnerability issues.

Pfsense is an excellent choice for building a home router, since it is an open source full firewall software and has all the features that most expensive enterprise products contain (Rubicon Communications, 2014). There is no need to have Unix or command line skills, as this firewall software comes with an easy to use administration interface. The hardware has to be compatible to run it, for home usage, it may run with one Ethernet and one wireless NIC or an additional switch to connect multiple computers to the protected network. However, no support is available for the ARM platform.

For the DIY enthusiasts, building a firewall with a raspberry pi 3b+ is an interesting project, since it can accommodate the same features as pfsense plus build additional capabilities for the IoT world such as a broker, as shown in Figure 12. Installing the operating systems and the software packages requires some Linux and command line skills. Consequently, running a Pi has many advantages but requires more time for the initial setup, which the small footprint and low-power consumption make easy to conceal. The Pi has embedded ethernet and Wi-Fi adapters and the wi-fi range is low, but it includes four USB ports to connect other adapters. Alfa Network builds powerful USB IEEE 802.11b/g/n network adapters that are compatible with the Pi for wide-range coverage.

The price for a COTS home router varies from 50 to 600 USD. Pfsense is installed on regular and small form factor computers, and the prices for the hardware may start at 100 USD. In addition, the price of a raspberry pi 3b SBC, case, memory card, and power unit is around 55 USD, while Alfa network adapters price around 40 USD. If there are multiple walls of home appliances that cause interference with wireless networks, multiple access points may be required for good coverage or access points may be connected with ethernet cables or over the powerline.

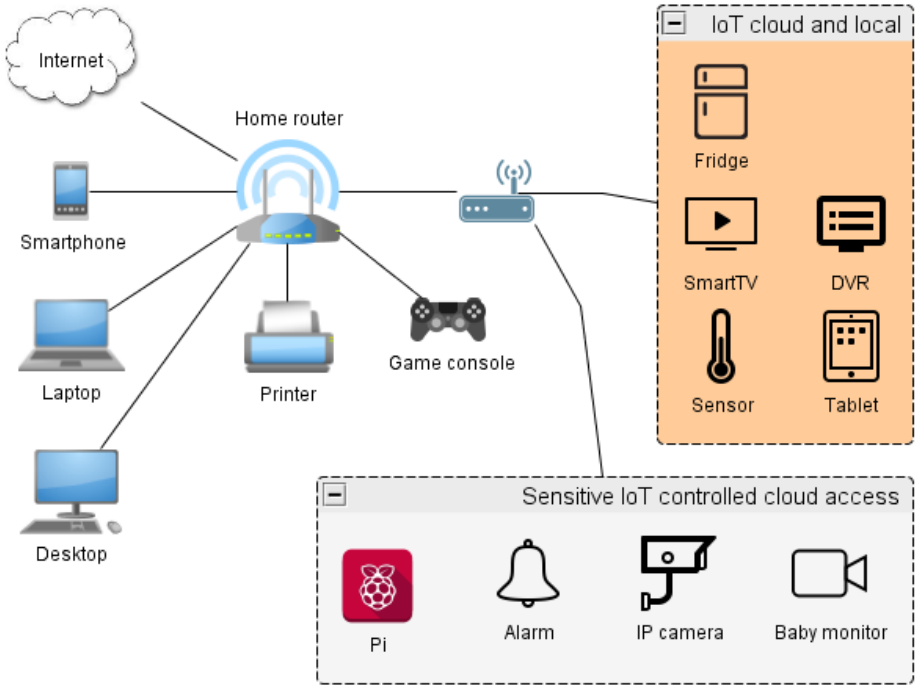


**Figure 12: Multiple private network segments with a game console in DMZ.**

Wireless signals usually do not have a delimmited boundary, and communications are susceptible to interception without requiring physical access. Consequently, using cabled networks for sensitive network communications is less prone to interception. Power lines can be used when ethernet cables to extend the network are not available, some of those adapters achieve faster network speed than wireless, and recent models perform encrypted transmissions.

In addition, securing data exchanges is crucial in wireless LANs and the Residential Wireless Audit Checklist (Farrington, 2005) can be used as a reference.

## 4.2. Basic network services

Doman name system (DNS) and dynamic host configuration protocol (DHCP) services are two of the most important services that allow devices on the network to interconnect and access resources.

### 4.2.1. DHCP

DHCP provides the local systems with the IP address and configurations they need to connect, and this service is usually running on the router providing access to the Internet. After segregating the network with an additional firewall, the systems connected on the other side will not reach the router, and if they do, the configuration provided may be inaccurate. The firewall protecting the private network, once configured for NAT, requires a different address space than the exterior network; thus, another DHCP server running on it provides IP addresses to the segment with the right network details, instructing the systems to use it as the default gateway.

### 4.2.2. DNS

The DNS service is critical for many applications; it translates the domain names into IP addresses. In most setups, the router relays the queries from the internal network to other DNS servers. By default, the ISP provides the DNS server used by the router. Configuring the device to use a trustable source for the DNS service provides additional protection against multiple threats. Some of these risks include: DNS cache poisoning, connecting to malware, phishing and scam sites or botnets, typosquatting, advertising, tracking activity, and accessing sites with inappropriate content such as pornography or drugs.

Comodo Secure DNS (Comodo, 2001) is a service that uses real-time block list (RBL) and displays a warning page whenever it detects access to a site containing potentially threatening content. This service does not require an account to use it, and the

server IP addresses are 8.26.56.26 and 8.20.247.20. The public service filters known malicious sites. In addition, Comodo Secure DNS offers the service to block certain addresses by category on its service for registered users.

Dyn Internet Guide (Dyn, 1998) blocks phishing and malware sites and 30 categories of content. The server addresses are resolver1.dyndnsinternetguide.com (216.146.35.35) and resolver2.dyndnsinternetguide.com (216.146.36.36). The service does not require an account unless customization of the categories is required.

Two other services are Norton ConnectSafe by Symantec (Symantec, 1995), and OpenDNS by Cisco (Cisco, 2017). The former one does not require an account and automatically blocks known unsafe, fraudulent, phishing and infected websites, while the later support also accounts for customizing the categories to block.

### 4.2.3. Dynamic DNS

The Internet IP addresses for most residential networks are dynamically assigned and change quite often. After every IP address change, the "A" record needs to be updated to the new address to allow the domain name to be resolved. Applications or devices such as video surveillance cameras, alarms, baby monitors, and sensors can be accessed more easily from the Internet by using domain names instead of IP addresses, and by setting up a domain name that is updated with every IP address change, making this process easy to remember and manage using mydomain.house instead of 456.123.789.153.

Duck DNS (Duck DNS, n.d.) is a free dynamic DNS service that allows up to five third-level domain registrations that sends a simple HTTPS request including the domain and a token that updates the record on the DNS. No client software or agents are required.

### 4.2.4. Network Intrusion Detection and Prevention Systems

Network intrusion detection and prevention systems (NIDS and NIPS) are present on some of the latest versions of home routers and firewalls, some of the most common

use software such as Snort, Suricata, or Bro. This type of software looks for anomalies in the network flow or specific pattern matches.

The ability to use configurable rules lets the software to be expanded to detect anomalies and protocols. In addition, rulesets are often updated, emerging threats and talos rulesets can be used together or independently, and the choice of rules in use requires frequent updates. The engine itself also has to be updated to fix any bugs or vulnerabilities that may expose the IDS system or the network to attacks.

Multiple factors affect the efficiency and accuracy of IDS systems, with the placement of the sensor and encryption being the two that severely affect the detection capabilities by preventing the flow from being inspected properly.

The NIDS can be used either in parallel or inline configurations as seen in Figure 13. If the sensor is placed inline, it could be employed as NIPS to block connections or flow and as an NIDS to log alarms when anomalies are detected. The sensor is placed in parallel when detection of anomalies, logging alarms and passive interaction is desired.

NIPS have advantages and disadvantages when compared to NIDS. More care should be taken for the configuration of NIPS as it will drop connections. One benefit of blocking anomalies is that it can mitigate the risk of a vulnerability being exploited while the saffected software is patched.

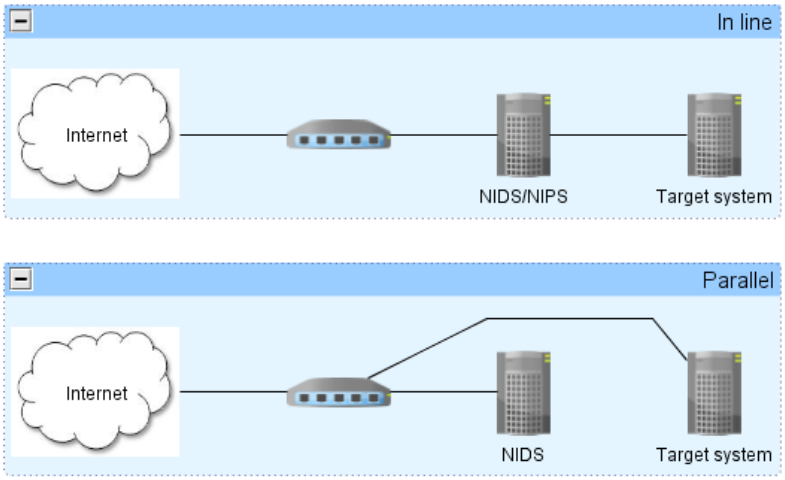


**Figure 13: IDPS placement in line and in parallel**

On outdated or vulnerable systems using host intrusion prevention systems (HIPS) and firewalls, strong network segmentation and NIPS can significantly reduce the exposure from vulnerabilities. NIPS rulesets sometimes also contain zero-day rules that block attacks to vulnerabilities that still have no patch released.

On the other side, the intrusion detection systems require close surveillance and follow-up; when an attack or anomaly is detected, an alarm is logged. Often the NIDS will log the attack but will not do anything to prevent it.

For both types of systems to be effective, the configuration needs to define at least what functions and ports are enabled on the protected systems, the addresses of the internal networks. A default configuration may produce thousands of alerts every day to just a few; thus, setting the right values is of utmost importance.

Bro has the possibility to integrate threat intelligence feeds and to create the dedicated intel.log when a malicious source is detected. The software can write the logs in JSON format to make integrating with other log aggregation or SIEM systems easier.

### 4.2.5. Proxies and Filtering

A proxy performs the requests on behalf of another system; it may work as forwarding proxies which are used by internal systems to access the Internet. Reverse proxies are the opposite; these give access to the internal systems from the web. Systems behind a reverse proxy are seen from the external networks as if it was the reverse proxy itself, which translates all the external calls and addresses to the internal names.

Using proxies in both ways provides advantages such as content filtering and cache. Filtering the incoming traffic at the reverse proxy prevents malicious requests from exploiting vulnerabilities or accessing interfaces that should not be accessible.

Deep inspection of every request permits much more granular filtering when the reverse proxy is the encryption endpoint. Installing a web application firewall (WAF) to the reverse proxy gives the intrusion detection and protection capabilities for web requests that would otherwise be impossible for the NIDS to be inspected.

ModSecurity is a very powerful WAF that is configured using rules. The filtering capabilities of the WAF come from the ruleset. Providers such as Trustwave (Trustwave SpiderLabs, 2004) and AtomiCorp (AtomiCorp, 2005) offer commercial rulesets, while OWASP (OWASP, n.d.) has an open source counterpart with high quality protection.

WAF rulesets can be used to detect and prevent attacks such as SQL injection, cross-site request forgery, and cross-site scripting. The WAF can also be used to fix faults in applications with virtual patches.

Virtual patch is the modification of the application behavior or contents by the WAF; the vulnerability still exists, but it is not exploitable as long as the WAF protects it.

### 4.2.6. MQTT

The IoT devices are usually designed to use little power and resources and may run on batteries for years. The overhead produced by HTTP and some other protocols can be significant. The HTTP headers usually use a few hundred bytes, which is a large amount when only a "yes" or "no" response is required. Message Queue Telemetry Transport (MQTT), is a lightweight protocol that requires only a few bytes and is ideal

for this kind of application. MQTT messages are often only a few bytes and can be transmitted with little bandwidth and power usage.

The MQTT protocol is based on publish and subscribe message exchanges using brokers, as shown in Figure 14. The broker serves as a relay to deliver the messages to its destination, and those messages are requested by the broker by subscribing to a given topic. One example is a smartphone application subscribing to the living room temperature, and the application subscribes to the topic, while the broker delivers the published messages.

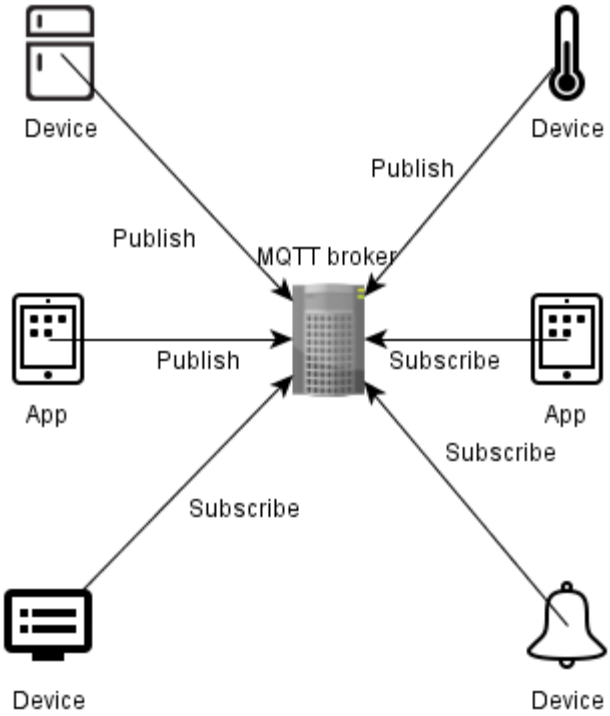


**Figure 14: Hub and spoke model**

However, not all MQTT brokers are equal in encryption and authentication features. Eclipse Mosquitto (Eclipse Mosquitto, 2010) is an open-source software that supports most official MQTT features, including TLS support for encrypted network connections and authentication.

The MQTT protocol supports three Quality of Service levels. QoS level 0, “deliver at most once," which does the best-effort delivery, level 1, “delivers at least

once," guaranteeing the delivery but may send the message more than once, and level 2, “delivers exactly once," which ensures that each message is delivered only one time.

Mosquitto MQTT broker by default accepts anonymous connections over clear text, and TLS encryption and username and password authentication, as shown in Figure 15, can be used as a minimum to protect confidentiality. Sending username and password over non-secure channels is prone to interception. Mosquitto MQTT supports client certificates for improved authentication.

```
pi@raspberrypi:/etc/mosquitto $ sleep 2 && mosquitto_pub --cafile /etc/mosquitto/ca_certificates/ca.cr
t -d -t test_mqtt -m "MQTT mosquitto test successful!" -u $mqttUser -P $mqttPass -h `hostname` -p 8883
 &
[1] 21781
pi@raspberrypi:/etc/mosquitto $ mosquitto_sub --cafile /etc/mosquitto/ca_certificates/ca.crt -d -t tes
t_mqtt -u $mqttUser -P $mqttPass -h `hostname` -p 8883
Client mosqsub/21793-raspberry sending CONNECT
Client mosqsub/21793-raspberry received CONNACK
Client mosqsub/21793-raspberry sending SUBSCRIBE (Mid: 1, Topic: test_mqtt, QoS: 0)
Client mosqsub/21793-raspberry received SUBACK
Subscribed (mid: 1): 0
Client mosqpub/21795-raspberry sending CONNECT
Client mosqpub/21795-raspberry received CONNACK
Client mosqpub/21795-raspberry sending PUBLISH (d0, q0, r0, m1, 'test_mqtt', ... (31 bytes))
Client mosqpub/21795-raspberry sending DISCONNECT
Client mosqsub/21793-raspberry received PUBLISH (d0, q0, r0, m0, 'test_mqtt', ... (31 bytes))
MQTT mosquitto test successful!
```

**Figure 15: Mosquitto publish and subscribe commands over TLS**

## 4.3. Inventory

An accurate and updated inventory of the systems running on the network ease monitoring, and the details retrieved from it reduce the number of false positives on the IDS and filter out the protocols, ports, URI, sources, destinations, and applications.

The list of systems connected to the network can be gathered passively using the DHCP service logs and the network flow activity. For performing the operating system identification, Nmap can be used for the active identification and P0f running at the router for the passive identification.

### 4.3.1. Active scanning

Active scanning is performed by sending network packets and comparing the responses looking for specific operating system characteristics. Typically, the active scan identifies in a few seconds all the devices connected to the network, as shown in Figure 16. It detects what services are listening on each device and interrogates the detected services to obtain protocol versions and other useful information. However, it has some negative consequences such as the fact that devices and software may be especially sensible to this kind of traffic and may crash during the scan.

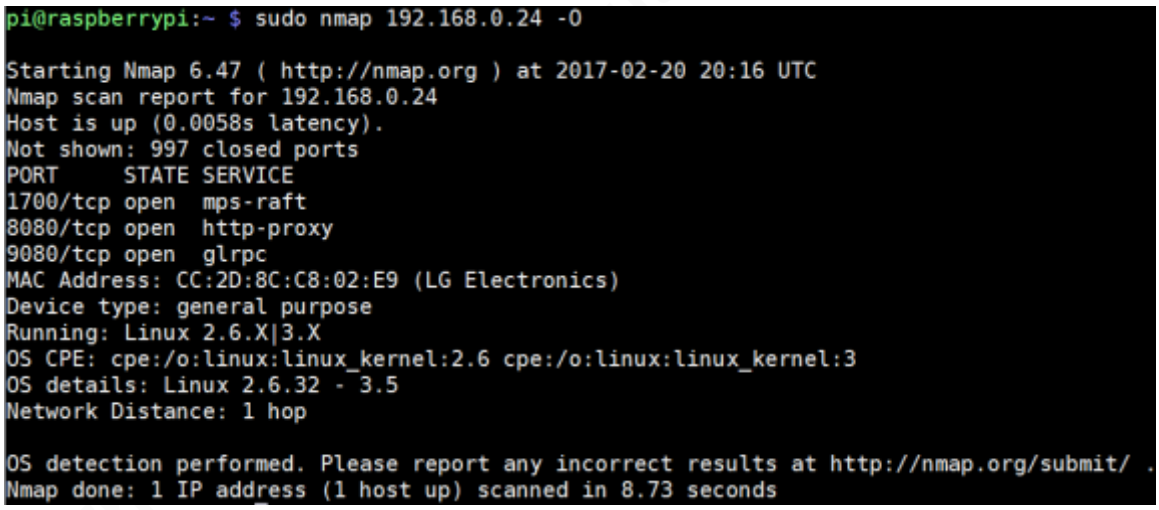


**Figure 16: Nmap active operating system finger printing**

The ping command, as shown in Figure 17, may also be used to look for responding IP addresses and the arp command to list the corresponding MAC addresses. Some devices block or do not respond at all to echo requests sent by ping.

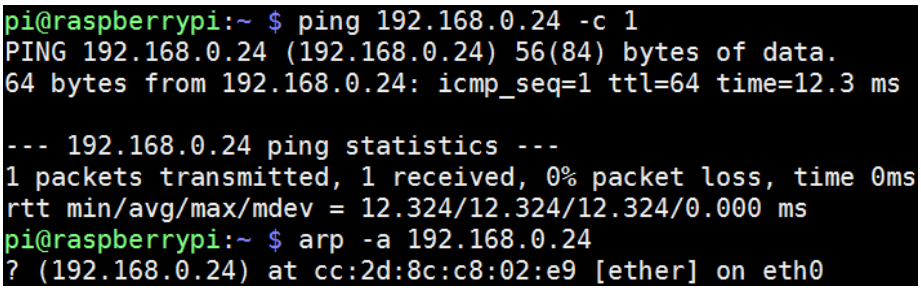


**Figure 17: Enumerating the network with ping and arp commands**

### 4.3.2. Passive scanning

Passive operating system fingerprinting, as shown in Figure 18, is made by analyzing the characteristics of network packets such as flags, window size, fragmentation, and TTL. P0f is an example of tool that can identify the OS types using network traffic.

```
pi@raspberrypi:~ $ sudo p0f
p0f - passive os fingerprinting utility, version 2.0.8
(C) M. Zalewski <lcamtuf@dione.cc>, W. Stearns <wstearns@pobox.com>
p0f: listening (SYN) on 'eth0', 262 sigs (14 generic, cksum 0F1F5CA2), rule: 'all'.
192.168.0.21:12023 - Windows XP/2000 (RFC1323+, w+, tstamp-) [GENERIC]
  Signature: [8192:128:1:52:M1460,N,W2,N,N,S:.:Windows:?]
  -> 192.168.0.14:22 (distance 0, link: ethernet/modem)
192.168.0.23:51067 - UNKNOWN [65535:64:1:60:M1460,S,T,N,W8:.:?:?] (up: 31 hrs)
  -> 192.168.0.14:80 (link: ethernet/modem)
```

**Figure 18: p0f passive OS fingerprint**

Tracking the DHCP leases provides the IP and MAC addresses in the network, as shown in Figure 19, depending on the package used to provide the DHCP service, dnsmasq, for example, use the "/var/lib/misc/dnsmasq.leases" file but other packages use either their lease file or "/var/log/messages" for logging.

```
pi@raspberrypi:~ $ tail -1 /var/lib/misc/dnsmasq.leases
1487678500 78:40:e4:e4:5d:38 10.3.2.150 android-a53fa03b3a066cdf 01:78:40:e4:e4:5d:38
```

**Figure 19: dnsmasq DHCP lease file**

Network intrusion detection systems and packet sniffers are useful to identify the connections, ports, applications, and protocols in the network. Examples of each of the previous are Bro IDS that have a set of log files that contain all that information separately and Wireshark that captures traffic on a given interface and has a set of statistics tools to quickly identify the same info. The first performs its analysis in the background, while the the latter requires user intervention.

## 4.4. Vulnerability detection

Several companies develop software to check the network and devices for known vulnerabilities and misconfigurations. For example, tenable Nessus and Rapid7 Nexpose

are commercial alternatives that offer a home or community version available for free with some limitations. On the open-source side, OpenVas provides this capability.

Internet of Things (IoT)- specific scans can also be done with IoT Seeker (Rapid7, 2016).IoT Seeker's main objective is to find devices that can be easily hijacked and used for malicious purposes, as shown in Figure 20:

```
/Users/rapid7/freetools>perl iotScanner.pl 1.23.123.431,
1.23.123.443,1.23.123.453,1.23.123.457,1.23.123.459,1.23.123.461,1.
23.123.462,1.23.123.463,1.23.123.465,1.23.123.466,1.23.123.467,1.23
.123.469,1.23.123.472,1.23.123.473,1.23.123.475,1.23.123.477,1.23.1
23.479,1.23.123.480,1.23.123.481
device 1.23.123.431 is of type Stardot still has default passwd
device 1.23.123.443 is of type Arecont has changed passwd
device 1.23.123.453 is of type American Dynamics has changed passwd
device 1.23.123.457 is of type W-Box has changed passwd
device 1.23.123.459 is of type Arecont has changed passwd
device 1.23.123.461 is of type American Dynamics has changed passwd
device 1.23.123.462 is of type W-Box has changed passwd
device 1.23.123.463 is of type Arecont has changed passwd
device 1.23.123.465 is of type American Dynamics has changed passwd
device 1.23.123.466 is of type W-Box has changed passwd
device 1.23.123.467 is of type Arecont has changed passwd
device 1.23.123.469 is of type American Dynamics has changed passwd
```

**Figure 20: IoT Seeker tool checking for default credentials**

Many of the checks performed are non-intrusive and are based solely on the version identification of the service. Due to this discovery method, if a device falsely reports the software or version, then using the results of the vulnerability assessment is incomplete or inaccurate.

The use of credentialed assessments reduces the number of false positives and allows the analyst to discover additional software or vulnerabilities compared to the use of a traditional network scan.

### 4.4.1. Vulnerability scanning considerations

Vulnerability scans can cause service disruptions if not configured properly, some items to consider are what plugins to enable, speed, and frequency. By using safe checks

only, vulnerability scanners restrict the use of plugins or checks that may be disruptive to the network.

In addition, by scanning often, the vulnerability assessments can perform network discovery, and detect vulnerabilities earlier so action can be taken to remediate the issues.

Vulnerability scans help to identify devices that do not get updates automatically and IoT devices that require human intervention to be updated, which often involve a flash update to load a new version of the software.

Even small computers or SBC can use hundreds of concurrent connections to many systems to finish the scanning faster. The speed of the scan is important if a network device or firewall have issues to handle many concurrent connections.

All vulnerability scanners can enable or disable checks that are not required. The number of resources used during the scan can be reduced by enable only those plugins for vulnerabilities associated with systems on the network. Most checks are classified depending on the the operating system or software they correspond to such as Linux, Windows, Oracle databases, and Apache web servers.

Supervising the first scans is especially important to gauge the resource consumption and check for service disruptions. The number of parallel checks combined with network congestion detection help to reduce the scan speed when network slowness is detected.

## 5. Building a home firewall

Vulnerabilities in commercial firewalls are frequent; recently, many different firewall devices have been exploited including several big name firewall companies, just to name a couple, Cisco and Fortinet had exploits made available in late 2016 after Shadow Brokers leaked some NSA zero-day exploits. There are many offerings in the market for a home or small office networks with all budgets, but buying a 600 dollar unit does not guarantee a no vulnerability environment and sometimes does not even guarentee performance.

## 5.1. Hardware selection

Building a firewall to protect networks with 100 Mbit bandwidths or lower has the advantage of customizing every component and knowing which software is being used. A raspberry pi 3 model b may handle the traffic between multiple network segments and provide the IDS, reverse proxy, WAF, and MQTT broker capabilities at the same time.

The raspberry 3b has ARM processor, two network interfaces and four USB ports, and run the Linux operating system. These four USB ports give the pi the flexibility to use dongles to add more network interfaces or attached storage. The embedded wireless interface can provide access point services with limited area coverage.

Tiny form factor PCs are almost as little as the Pi and most are x86- or amd64-processor based and offer more processing power and memory. Usually, the more powerful units are bigger and more expensive. When size, power consumption, and flexibility are important, the Pi would be a better option while the PC is a better candidate when high bandwidths are required.

Some additional hardware that may be useful are external USB network adapters from Alfa networks; their units have very powerful RF and support many different antennas for indoor and outdoor use, while some units support both the 2.4, and 5 GHz bands.

## 5.2. Operating system selection

Linux operating systems are usually free and can be customized to have any feature, from which for running on a Pi, there are two popular options, Raspbian and Arch Linux ARM. For the x86- or amd64-based units, Pfsense is a remarkable one for using a PC as a firewall.

Raspberry Pi Foundation maintains a Debian Jessie based Linux called Raspbian (Raspberry Pi foundation, 2015) and is available to download from their website free of charge. The operating system comes in two flavors, raspbian Jessie lite and raspbian with "pixel." Firewalls must be hardened and all non-essential components must be disabled, so if this operating system is going to be used, it is better to use the lite version.

Arch Linux ARM (Arch Linux ARM, 2009) is an alternative operating system which "aims for simplicity and full control to the end user" for the Pi.

Pfsense is the perfect alternative for x86 or amd64 based units. It is designed specifically to be a firewall and includes all features of any high-grade firewall. This operating system is open source and can be downloaded for free similar to the others.

## 5.3. Network design

The firewall must be the only entry point for all sensitive networks. Attaching more network interfaces is useful when there are devices that have different sensitivity levels, and using a dedicated interface per network segment for each different group provides more granularity on the access controls and reduces the impact when an element is compromised.

For example, supposing there are security surveillance cameras, alarm sensors, and home appliances that require network and internet connectivity, four different network adapters would be optimal. A cabled-network interface is desirable for connecting to the uplink router or firewall, which is similar to alarm sensors, while wireless cameras and DVR are very popular. Additionally, they are better if left alone in their segment and the additional segment for all other devices.

Communication encryption is mandatory for any data exchanges traveling through insecure channels such as the Internet or wireless LAN, and they are highly desirable for local networks when there are sensitive data or session credentials that have to be protected.

## 5.4. Intrusion detection and prevention

Preventing intrusions and detecting anomalies is easier when a defense in depth model is followed and the surface of attack is reduced. Thus removing non-essential elements, harden, patch frequently and monitoring reduce the risk of intrusion.

Protocol encryption prevents third parties from knowing the content that is useful for confidentiality, but often also prevents or makes the analysis for detecting the

intrusion difficult. HTTP and MQTT are popular protocols that use TLS for encrypting the network flows, which is an easy way for inspecting those flows for anomalies and attacks by decrypting, analyzing and then encrypting if they are necessary.

A reverse proxy using Apache with ModSecurity and owasp crs as WAF can receive the incoming requests, perform the full analysis, and proxy the request to the end system for either encrypted or clear text. This can be useful to provide TLS support to interfaces that otherwise use only clear text HTTP.

MQTT brokers can send and receive messages using TLS on the exposed interface and use clear text on the protected network to allow snort and bro inspection.

At the host level, IDS tools as ossec trigger alarms and may run scripts to actively block attacks, anomalies, and abusive usage.

Knowing the network and its components is key to spotting anomalies. Some anomalies can be identified due to communications between systems that are not supposed to exchange any data, unusual volumes of data, flow direction, configurations changes, new users, and changed passwords.

# 6. Conclusion

Defending the home network is an act of responsible ownership; it is no longer acceptable to connect devices to the network and forget them. Unfortunately, not all vendors are doing their part to secure the devices they sell, and due to the high internet bandwidth, the increasing number of devices, and the lack of proper security practices from both vendors and owners, actual and future large-scale attacks are capable to collapse the Internet access for entire countries and global services.

Vendors and researchers must join forces and ease the adoption of secure practices that make the network a safe place; in a world where everything will be connected, no device is safe by itself when all devices and networks are interdependent. Attackers are doing their share to look for vulnerabilities and diligently find bad practices. Systems such as DNS are fundamental for the internet, and a failure on them may cause global-service disruptions.

The traditional network and system models need to be refreshed for this new era. We are heading towards a model where there are no boundaries and no system owners. However, Most of the current protocols and applications are not ready for such an open environment with shared assets and "pay as you use" billing schemas.

There are no hacker proof systems yet, but the more effort required to break in and maintain undercover and the reduced benefit received, the less likely those systems are to be targeted.

## References

Arch Linux ARM. (2009). *Archi Linux Raspberry Pi 3 installation*. Retrieved from Arch Linux ARM: https://archlinuxarm.org/platforms/armv8/broadcom/raspberry-pi-3

AtomiCorp. (2005). *Atomic ModSecurity Rules*. Retrieved from AtomiCorp: https://wiki.atomicorp.com/wiki/index.php/Atomic_ModSecurity_Rules

Banana-pi. (2016/7, Jul/Jan 22/9). *BPI-R1 Manual*. Retrieved from banana pi BPI-R1 A20 dual core open source smart router: https://bananapi.gitbooks.io/bpi-r1/content/en/bpi-r15gbeethernetports.html

Cisco. (2017). *OpenDNS*. Retrieved from OpenDNS: https://www.opendns.com/

Comodo. (2001). *Secure DNS*. Retrieved from DNS by Comodo: http://securedns.dnsbycomodo.com/

Consult, S. (2016, Dec 6). *Backdoor in Sony IPELA Engine IP Cameras*. Retrieved from SEC Consult Security firm: http://blog.sec-consult.com/

Delaney, J. R. (2017, Jan 4). *The Best Wireless Routers of 2017*. Retrieved from PCmag: http://www.pcmag.com/article2/0,2817,2398080,00.asp

Duck DNS. (n.d.). *Duck DNS*. Retrieved from Duck DNS: https://www.duckdns.org/

Dyn. (1998). *Internet Guide*. Retrieved from Dyn: http://dyn.com/labs/dyn-internet-guide/

Eclipse Mosquitto. (2010, May 31). *Mosquitto Open Source MQTT broker*. Retrieved from Eclipse Mosquitto: https://mosquitto.org/

embeDD GmbH. (2017). *About DD-WRT*. Retrieved from dd-wrt: https://www.dd-wrt.com/site/content/about

Farrington, D (2005, Sep). SCORE: Checklists and step-by-step guides. Retrieved from SANS: http://www.sans.org/score/checklists/wireless

Goodin, D. (2016, Nov 28). *Newly discovered router flaw being hammered by in-the-wild attacks*. Retrieved from Arstechnica: http://arstechnica.com/security/2016/11/notorious-iot-botnets-weaponize-new-flaw-found-in-millions-of-home-routers/

Google. (2017). *Transparency report*. Retrieved from Google: https://www.google.com/transparencyreport/https/metrics/?hl=en

Hilton, S. (2016, Oct 26). *Dyn Analysis Summary Of Friday October 21 Attack*. Retrieved from Dyn: http://dyn.com/blog/dyn-analysis-summary-of-friday-october-21-attack/

Jelic, F. (2016, Nov 6). *Analysis: Record DDoS Attacks by Mirai – IoT Botnet*. Retrieved from Deep.dot.web: https://www.deepdotweb.com/2016/11/06/analysis-record-ddos-attacks-mirai-iot-botnet/

Keeper. (2017, Jan 13). *What the Most Common Passwords of 2016 List Reveals [Research Study]*. Retrieved from Keeper Security Blog: https://blog.keepersecurity.com/2017/01/13/most-common-passwords-of-2016-research-study/

Krebs, B. (2016, Oct 3). *Who Makes the IoT Things Under Attack?* Retrieved from Krebs on Security: https://krebsonsecurity.com/2016/10/who-makes-the-iot-things-under-attack/

MalwareMustDie. (2016, Oct 14). *MMD-0058-2016 - Linux/NyaDrop - a Linux MIPS IoT bad news*. Retrieved from Malware must die: http://blog.malwaremustdie.org/2016/10/mmd-0058-2016-elf-linuxnyadrop.html

McKeay, M. e. (2016). *Akamai's [state of the internet] / security Q3 2016.* Akamai Technologies.

Mitchell, B. (2016, Mar 26). *Functions and Features of Routers for Home Computer Networks*. Retrieved from Lifewire: https://www.lifewire.com/functions-and-features-of-routers-3986215

Manuel Leos Rivas, MLeosRivas@mastersprogram.sans.edu

Mitchell, B. (2016, Oct 19). *Gallery of Home Network Diagrams*. Retrieved from Lifewire: https://www.lifewire.com/home-network-diagrams-4064053

Nielsen, J. (1998, Apr 5). *Nielsen's Law of Internet Bandwidth*. Retrieved from Nielse Norman Group: https://www.nngroup.com/articles/law-of-bandwidth/

OWASP. (n.d.). *OWASP ModSecurity Core Rule Set Project*. Retrieved from OWASP: https://www.owasp.org/index.php/Category:OWASP_ModSecurity_Core_Rule_Set_Project

Patrick Eha, B. (2013, Sep 25). *An Accelerated History of Internet Speed*. Retrieved from Entrepreneur: https://www.entrepreneur.com/article/228489

Rapid7. (2016). *IoT Seeker*. Retrieved from Rapid7 IoT Seeker: https://information.rapid7.com/iotseeker.html

Raspberry Pi Foundation. (2015). *Raspberry Pi*. Retrieved from Raspberry Pi: https://www.raspberrypi.org/products/raspberry-pi-3-model-b/

Rubicon Communications, LLC (Netgate). (2014). *Getting Started*. Retrieved from pfSense: https://www.pfsense.org/getting-started/

Schulzrinne, e. a. (1998, Apr). *Real Time Streaming Protocol*. Retrieved from IETF: https://www.ietf.org/rfc/rfc2326.txt

Smith, T. (2015, Jul 29). *Sweet Security: Deploying a Defensive Raspberry PI*. Retrieved from Tripwire: https://www.tripwire.com/state-of-security/security-data-protection/cyber-security/sweet-security-deploying-a-defensive-raspberry-pi/

Symantec. (1995). *Norton Connectsafe DNS*. Retrieved from Norton Connectsafe: https://dns.norton.com/

TP-link. (2016, Jul 7). *What is a DMZ and how to configure DMZ host*. Retrieved from TP-link: http://www.tp-link.com/us/FAQ-28.html

Trustwave SpiderLabs. (2004). *Trustwave ModSecurity Rules*. Retrieved from ModSecurity: http://www.modsecurity.org/rules.html

Manuel Leos Rivas, MLeosRivas@mastersprogram.sans.edu

# Appendix

Scenario: Building a network using a Raspberry Pi model 3b, as shown in Figure 21 as firewall, IDS, and reverse proxy to protect a single network segment using NAT. The network includes some sensors using IoT; the reverse proxy feature is protected using ModSecurity WAF to give access to home automation administration software and IDS and active response capabilities. The Pi also hosts the MQTT broker.

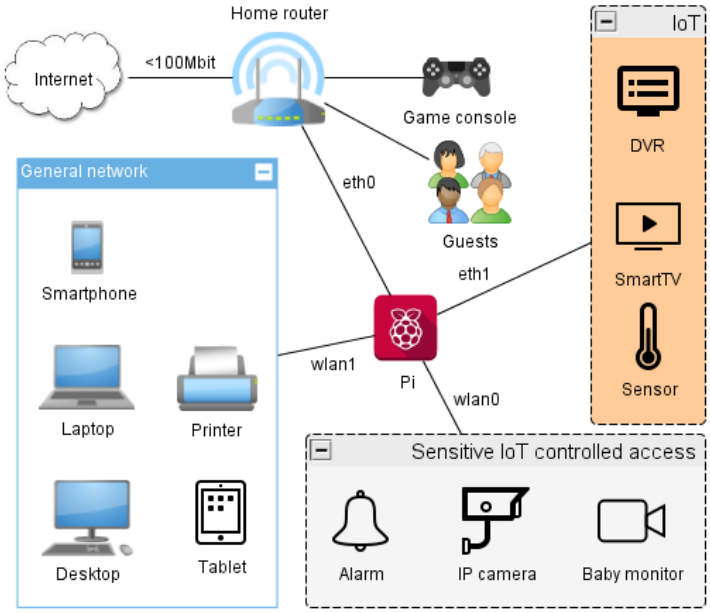


**Figure 21: My home network project.**

To do the headless operating system installation, the image from the raspberry pi website was downloaded, and the win32diskimager was used to write it on the micro sd card and then create an empty file in the root directory of the micro sd card called “ssh,” which is shown in Figure 22.

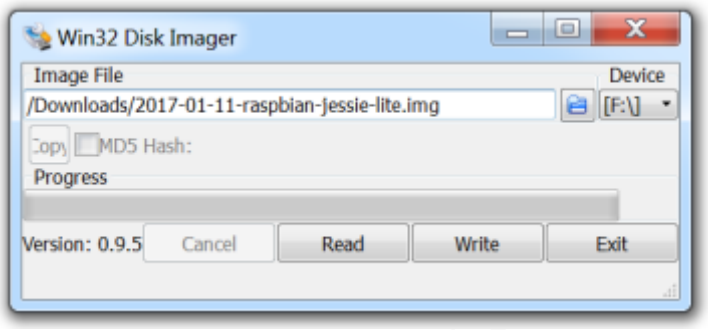


**Figure 22: Win32 Disk Imager writing to micro sd card (5min).**

The raspberry uses the latest raspbian lite, and the installation and configuration process of most components is semi-automated by Sweet Security (Smith, 2015).

Once the unit is running, nmap can find the IP address by logging in with the username "pi" and password "raspberry." The first thing to do, change the pi user password, runs "raspi-config" to extend the file system, reduces the amount of memory for the video to 16 MB, and enables ssh by default, as shown in Figure 23. This forces the Pi to reboot.

```
> ./nmap -p 22 192.168.0.0/24

Starting Nmap 6.49BETA1 ( http://nmap.org ) at 2017-02-26 21:32 Romance Standard Time
Nmap scan report for raspberrypi (192.168.0.14)
Host is up (0.046s latency).
PORT   STATE SERVICE
22/tcp open  ssh
MAC Address: B8:27:EB:6B:D2:17 (Raspberry Pi Foundation)
```

**Figure 23: Using nmap to find the raspberry pi on the local network (2 min).**

Then the user can download from https://github.com/spartantri/SweetSecurity the Sweet Security script and run it, which it installs ossec HIDS, bro IDS, critical stack threat intelligence plugin, snort, ELK stack and all prerequisites, as shown in Figure 24. The script also performs the base configuration. The installation process takes around two hours because some components are built from source, and bro in particular will take most of that time to compile.

```
pi@raspberrypi:~ $ cd /tmp
pi@raspberrypi:/tmp $ git clone https://github.com/spartantri/SweetSecurity.git
Cloning into 'SweetSecurity'...
remote: Counting objects: 194, done.
remote: Compressing objects: 100% (106/106), done.
remote: Total 194 (delta 64), reused 0 (delta 0), pack-reused 88
Receiving objects: 100% (194/194), 3.47 MiB | 305.00 KiB/s, done.
Resolving deltas: 100% (107/107), done.
Checking connectivity... done.
pi@raspberrypi:/tmp $ ./SweetSecurity/SweetSecurity.sh
Please enter your Critical Stack API Key:
12345-67890-12345-67890<--Get one from http://intel.criticalstack.com/
Enter SMTP Host (smtp.gmail.com): mysmtpserver.com
Enter SMTP Port (587):
Enter Email Address (email@gmail.com): myemail@emailprovider.whatever
Enter Email Password (P@55word): SomethingLongAndSafeIguess!
Installing Pre-Requisites...
```

**Figure 24: SweetSecurity installing all IDS stuff (~2 hours).**

After the core components are in place, the MQTT broker, local certificate authority, Apache, ModSecurity, OWASP CRS can be installed using the scripts at https://github.com/spartantri/rpi-nsm, as shown in Figure 25.

```
pi@raspberrypi:/tmp $ git clone https://github.com/spartantri/rpi-nsm.git
Cloning into 'rpi-nsm'...
remote: Counting objects: 44, done.
remote: Compressing objects: 100% (40/40), done.
remote: Total 44 (delta 18), reused 16 (delta 2), pack-reused 0
Unpacking objects: 100% (44/44), done.
Checking connectivity... done.
pi@raspberrypi:/tmp $ chmod +x ./rpi-nsm/rPi3-ap-setup.sh
pi@raspberrypi:/tmp $ ./rpi-nsm/rPi3-ap-setup.sh
Must be root
pi@raspberrypi:/tmp $ sudo ./rpi-nsm/rPi3-ap-setup.sh
Will proceed with default rPi3 SSID!
Usage:
sudo ./rpi-nsm/rPi3-ap-setup.sh [apName]
Installation will start in 30 seconds, press <CTRL+C> to cancel...
```

**Figure 25: Configure and install hotspot software (10–15 min).**

Having the Apache, WAF, and all certificates ready, the "webappprofiler" tool from https://github.com/spartantri/webappprofiler can be used to generate a full positive security configuration that allows only known good requests over the reverse proxy to reach the end applications. Webappprofiler uses ZAP to gather all the web-application

elements and then build and store a profile in XML format, which XML is transformed using XSLT into modsecurity rules that are ready to be deployed, as shown in Figure 26.

```
#
# Session-Handling
#
<xsl:choose>
<xsl:when test="./@ratio &gt; 0 and ./@score &gt; 0">
<xsl:if test="count(@required) > 0">    SecRule &amp;REQUEST_COOKIES:<xsl:value-of select="./@name"/> "@eq 0" "id:9931733,
    SecRule REQUEST_COOKIES:<xsl:value-of select="./@name" /> &quot;!@rx <xsl:value-of select="./@regexp"/>&quot; "<xsl:i
</xsl:when>
<xsl:otherwise>
<xsl:if test="count(@required) > 0">    SecRule &amp;REQUEST_COOKIES:<xsl:value-of select="./@name"/> "@eq 0" "id:9931733,
    SecRule REQUEST_COOKIES:<xsl:value-of select="./@name" /> &quot;!@rx <xsl:value-of select="./@regexp"/>&quot; "<xsl:i
</xsl:otherwise>
</xsl:choose>

SecRule REQUEST_COOKIES:<xsl:value-of select="./@name" /> &quot;@rx <xsl:value-of select="./@regexp"/>&quot; "id:9931733,
```

**Figure 26: Sample webappprofiler XSLT section.**

Once the XML profile is connected to the python script that creates the rules, only a few seconds are needed to generate thousands of ModSecurity rules, and the number of rules depends on the complexity of the web application, as shown in Figures 27 and 28.

```
$ time python simplerules.py
Writing modsecurity rules file... (modsec.rules)
Generated 2858 rules

real    0m0.272s
user    0m0.000s
sys     0m0.015s
```

**Figure 27: Transformation script writing ModSecurity rules.**

```
#
# Session-Handling
#
    SecRule &REQUEST_COOKIES:wiki_session "@eq 0" "id:9980019,phase:2,setvar:tx.score=+10,pass,msg:'M
    SecRule REQUEST_COOKIES:wiki_session "!@rx ^[a-fA-F0-9]{32}$" "id:9990012,phase:2,t:none,t:urlDec


SecRule REQUEST_COOKIES:wiki_session "@rx ^[a-fA-F0-9]{32}$" "id:9980020,phase:2,t:none,pass,setsid:%
```

**Figure 28: ModSecurity rules ready to be deployed.**