Secure IP Camera Video Streaming Through Kurento Media Server

Kgothatso Ngako
Council for Scientific and Industrial Research, Pretoria, South Africa
kngako@csir.co.za

Abstract

With the rise of the IOT botnets which exploit vulnerable IOT devices, consumers with devices such as IP cameras accessible on the Internet might be involved in the next DDoS attack without even knowing it. Due to the nature of the vulnerabilities most mitigation lie outside the control of the consumer.

Most IP cameras allow for consumers to view the feed of their video via a RTSP or alternative streaming protocol. For local area networks this setup works fine. But if the consumer wants to access the video stream from outside the network they are left without an easy out of the box option for them to securely access the video stream without falling pray to the threats currently on the Internet.

This paper is a case study on how Kurento Media Server can be used to secure IP camera feeds through securing their network topology and creating a KMS gateway to securely relay Kurento protocol communications.

Keywords: webrtc, ip camera, streaming, media server, kurento

1 Introduction

Publicly accessible internet protocol (IP) cameras are widely used to provide live video streams to their users without the need of a computer. Users who don't take precautionary measures with regards to the security of their devices have become victims of exploitation (sometimes without knowing), as seen from the existence of sites that index live ip cameras that have weak or non-existant authorisation techniques (*Insecam - World biggest online cameras directory*, n.d.).

Initially many thought that competent users of these devices could secure them by following a number simple steps. Namely using strong passwords on the device, having the camera encrypt the video feed (through secure or password protected media transmission protocols like RTSP), keeping the firmware up to date, and having secure network communications on the device (in the case that the device allows a web accessible admin facility) (Information, 2013).

Unfortunately all of these steps do nothing to keep your device safe from being a slave in a botnet since most vulnerabilities lie outside the control of the user. A majority of users can do nothing to mitigate most of the vulnerabilities that exist on these devices due to how the manufactures of the devices shipped them. Some of these vulnerabilities are hard-coded credentials, inability of end-users to update

firmware, lack of support for https, undocumented points of entry points in network communication protocols used by devices(Spring, 2017).

A majority of these vulnerabilities can be mitigated by making sure the devices are not directly accessible through the Internet, and sending all data through a media server. IP camera users can then focus on securing the media server as it will be the main point of entry when they want to access video streams.

This paper sets out to define ways for users to securely access their video streams without exposing their devices to the many threats that exist on the internet.

The following sections in this paper are as follows. **Background**, covers Vulnerabilities, Exploits, and Mitigations that are related to IP camera security. **Methodology**, details the steps we followed to secure an IP camera in our sandbox. **Future work**, details the future work which can add to the security of the system. **Results and conclusion**, provides are short discussion on the work and reason behind the system.

2 Background

Many consumer grade IP cameras have been found to have vulnerabilities making them susceptible to attacks. The alarming aspects of a large portion of these vulnerabilities is that they can be exploited remotely (the attacker never requiring physical access to the IP cameras). Some exploits exist in protocol used by IP cameras, others are a problem with the configuration of the cameras subsystem/sub-modules. Once the vulnerabilities on an IP camera are exploited, the attacker can recruit the IP camera as to as a node in a Botnet (Tim Yeh and Lu, 2017) (McMillan, 2017).

2.1 Vulnerabilities

Researchers have shown that IP cameras (and IoT devices as a whole) face security concerns at either of the layers that make up the IoT network (Application, Perception, and Network layers) (Alaba et al., 2017*a*).

Some vulnerabilities exist in overlapping layers of the IOT network. Below is a breakdown of the most common vulnerabilities found to compromise IP camera security.

2.1.1 Weak default credentials

Having weak default or no credentials on internet connected devices is one of the many ways that the security of IP cameras get compromised. Authentication credentials give access to different uses of an IP camera, from the admin console, to root shell access. Usually consumers are giving the option to set their own unique authentication credentials for the admin console. But it is barely ever the case that users are giving the option to update the shell root credentials as that's where the IP

camera's firmware sits and menufactors don't make tend to keep these credentials away from the end user.

Even though manufacture don't make the credentials to the root shell of IP camera's public the recent Mirai Botnet attack has shown how widely is the use of weak authentication credentials on IP cameras by accumulating over 100 000 nodes in it's botnet through just try 60 different factory default username/password like admin/admin, or root/superadmin and effectively took down the internet(Cimpanu, 2017).

2.1.2 Low support for HTTPS

Most IP cameras and IoT devices run on Low Power and Lossy Networks (LLN). These networks are designed for energy efficiency to give them a rich set of applications. Unfortunately they obtained this rich set of applications at a trade off that leads them to perform poorly with regards to cryptography related operations (Alaba et al., 2017*b*). Orginally this was the case for why encryption based communication protocols like https aren't the default when it comes to accessing an IP camera's admin console.

2.1.3 Vulnerable Firmware

Firmware is the software that runs on read only memory (rom) that IP cameras require to function as expected. Due to the high rate at which vulnerabilities are being found in software it is best practise to keep a devices firmware up to date. This is a mitigation against zero days attacks being carried out on the device running vulnerable firmware.

Unfortunately some manufactures of IP cameras don't make updates available to the users of their devices. And even when the updates are made available most users don't have the technical capabilities to successfully execute the update.

2.1.4 Vulnerable Protocols

IP cameras use a vast number of protocols to give the end user value. Vulnerable consumer grade web cameras have been found to contain vulnerabilities such as running an Unencrypted Telnet servers, an old version of MiniUPnP, and enabled remote NAT-PMP detection(Williams et al., 2017) (Patton et al., 2014)

2.2 Exploits and Threats to IP camera

From the above vulnerabilities, exploiting menufacturers use of weak/common default authentication credentials has to the most notable exploit of the past few years. The DDoS attack on Dyn Name servers through the Mirai Botnet. A botnet which had compromised an estimated 100 000 internet connected IP cameras and routers which brought a vast chunk of the Internet to a standstill (Hilton, 2016).

2.2.1 Mirai Botnet

Through the static code analysis which was carried out on Mirai source code, which is publicly available, (Hamdija and Sasa, 2016) have found that the code is divided into three modules: 1) **Bot module**, 2) **Command and Control (C2) server** module, and 3) **Loader module**.

The **Bot module** (which carries out the scanner, attack, and killer submodules) is what executes on an infected IoT device and it communicates with the **C2 server** to carry out attacks when the need arises.

The **Loader module** receives information about vulnerable IoT devices and serves and infects the devices with the Bot module.

The botnet propagates through the scanner sub-module discovering vulnerable IoT devices through testing and randomly generated public IP addresses and passing the public IP address to the Loader module. A vulnerable device is seen as any device that allows for root authentication through either one of the 62 factory default username and password combinations. The propagation vector can be modified to exploit either one of the vulnerabilities which were listed above. This is how Persai got created.

For the sake of this paper we will only focus on the **Bot and Loader** modules. Because if we stiffle

Ever since the success of the Mirai Botnet attack variations of the botnet have been found. Persirai uses a zero-day vulnerability to infect devices and control them remotely (Tim Yeh and Lu, 2017). The damage that can be caused through exploiting vulnerable will impact organsations in ways they are currently incapable of combating (Mansfield-Devine, 2016).

2.3 Mitigation's

There exists a number of mitigation against the threats which an IP camera is facing. Most mitigations mirror their vulnerabilities. For weak/common authentication credentials, stronger authentication credentials/techniques could be applied. For vulnerable firmware software updates and securtiy patches can be applied. Vulnerable network protocols can be replaced with their more secure alternatives.

Unfortunately their is no way to ensure that any device that is accessible on the Internet is safe from all the threats which are consistently being produced. One of the most preferred mitigation is found in securing the network which devices are connected too.

2.3.1 Network Security Implementation

The are a number of ways for one to set up a network where connected devices are safe from Internet launched attacks. Due to the multitudes of options users can then

choice a method that suits their needs and resources (Oxenhandler, 2003). The following are a number of options that we can go for to prove secure video streaming.

Virtual Private Network (VPN) technology is usually used to extend a private network accross the public network and safeguard devices from public Internet traffic. Depending on technical capability a user could set up their own VPN or use one from a trusted service provider.

Alternatively the routers which the IP cameras are directly connected too should have the Universal Plug and Play protocol disabled to not allow the devices to listen for connections that come directly from the Internet (Miller et al., 2001).

A firewall can be setup on a network to block incoming Internet traffic that wants to connect or discover local IoT devices.

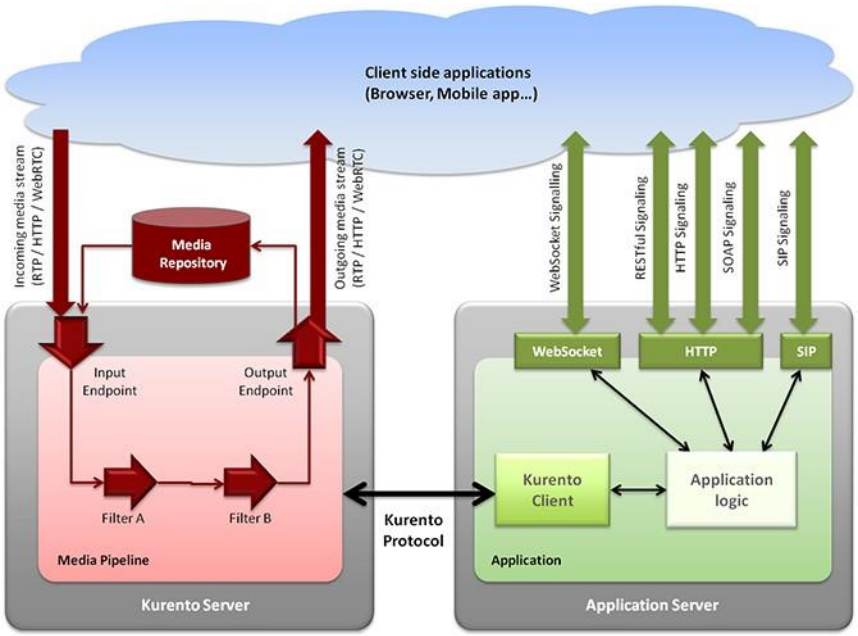
This wealth of options when it comes to securing is vital with IP cameras and other IoT devices as the network topologies they exist in are quite dynamic as compared to traditional computer networks (Alaba et al., 2017*b*). Once the network is secure we can focus on setting up the media server. The media server will then need to carry out the application layer functions while maintaining security.

2.4 Kurento Media Server

The Kurento Media Server (KMS) is a Web Real-Time Communication (WebRTC) based technology that handles media transmission, processing, loading, and recording and supports different network streaming protocls (like http, RTSP and WebRTC). KMS can be used to develop a web application to broadcast media from varying devices that serve and consume varying media protocols with automatic media transcodification between any of the codecs supports by GStreamer (low level implementation library). The Kurento Architecture can be seen on figure 1.

3 Methodology and Results

One of the surest ways to secure IP cameras from botnets like Mirai is to not allow for network communications originating from outside the network using one of the methods above. But unfortunately, for the case of IP cameras, this



Kurento Architecture. Kurento architecture follows the traditional separation between signaling and media planes Kurento (2015).

Figure 1: Kurento Architecture

will keep users/viewrs from accessing the media on the IP cameras whej they are outside the network.

The solution that we developed to counter this drawback adds a module to the machine running KMS which will allow the machine to **instantiate, persist and relay Kurento protocol communications** between the Application server and the KMS. By doing this an application server will need no knowledge of a machine running an instance of KMS until the **KMS-gateway module** on the machine running KMS becomes live and connects to the application server.

This **KMS-gateway** module will give us the option to deploy such a machine on a network where all the devices aren't accessible from the internet while allowing the media streams from IP cameras to be viewed through the interfaces provided by the **application server** and KMS. This allows users who have a need to access the media streams from their IP cameras to now do so securely.

Since the devices which viewers will use to access the **application server** will not be able to connect to the IP camera's media streams, the **KMSgateway module** and publicly accessible **application server** we use a secured turn (relay) server to handle

media streaming between different networks (Mahy et al., 2010) (Matthews et al., 2010).

3.1 Mitigating against Botnet discovery

The IP cameras are isolated from incoming Internet connections, and all media or data from IP cameras goes through the **KMS-gateway module** which securely feeds it to the publicly accessible **application server** that viewers can access through a browser or application that the **application server** makes available. Any **Bot or Loader modules** trying to discover or infect machines on this network (even the ones with uncorrected vulnerabilities) will not be able to connect to it. Like any other device not within the network not using the coupled **KMS-gateway** and **application server**.

3.2 Mitigating against the propagation of a Botnet

Since this network blocks all incoming connections and only allows outgoing connections, we effectively disable the ability for botnets like the Mirai botnet from discovering this network. And since the responsibility of connecting the application server and the Kurento Media Server is now placed in the new **KMS-gateway module** this network configuration succeeds in keeping the IP cameras safe from exploits.

Even in the case that one of the devices on the network becomes infected by the **Bot** module of the Mirai botnet, the **Loader module** will not be able to infect the other devices on the network unless the infected device re-configures the network to allow for incoming connections.

3.3 Monitoring for Compromised Network

The application server easily monitors network activity for each of the addresses of the IP cameras by trying to instantiate communications without the use of the **KMS-gateway module**. A flag is raised once a connection to the IP camera is established and adminstrators of the isolated network can be nofitified of possible compromise.

The monitoring tool is built on top of the nmap network exploration tool.

4 Future Work

The current system can be strengthened by developing it into a IDS which identifies the Mirai botnet behaviour of killing processes that are running listening on a predefined high port or through intercepting the unencrypted communication between the **C2 server** and the **bot** as described Hamdija and Sasa
(2016).

Other researchers have modified the Mirai C2 server to catalogue vulnerable IoT devices (Jerkins, 2017). This modified Mirai C2 server integrated with the media server

to monitoring IOT devices in our network will enrich the flag events raised and allow adminstrators to take more informed actions.

5 Conclusion

We have shown that we can secure the public video stream produced by an IP camera using the KMS and an additional gateway module while mitigating the risks which come with having an Internet connected IP camera.

This approach of serving IP camera content moves away from having IP cameras on an open network to a more limited and secure way of accessing a closed network that gives users and administrators generic access to their data and media in a more trusted manner. This is solution is by far not a perfect solution as it does require technical capabilities that the average user of IP cameras may not have but it could be beneficial for the advancement of more secure IP cameras and media servers.

References

Alaba, F. A., Othman, M., Hashem, I. A. T. and Alotaibi, F. (2017*a*), 'Internet of things security: A survey', *Journal of Network and Computer Applications* **88**(Supplement C), 10 – 28.
URL: *http://www.sciencedirect.com/science/article/pii/S1084804517301455*

Alaba, F. A., Othman, M., Hashem, I. A. T. and Alotaibi, F. (2017*b*), 'Internet of things security: A survey', *Journal of Network and Computer Applications*
.

Cimpanu, C. (2017), 'Someone published a list of telnet credentials for thousands of iot devices','"https://www.bleepingcomputer.com/news/security/ someone-published-a-list-of-telnet-credentials-for-thousands-of-iot-devices/". Accessed: 2017-09-18.

Hamdija, S. and Sasa, M. (2016), 'Analysis of mirai malicious software'.

Hilton, S. (2016), 'Dyn analysis summary of friday october 21 attack', https:// dyn.com/blog/dyn-analysis-summary-of-friday-october-21-attack/. Accessed: 2017-08-10.

Information, C. (2013), 'https://www.consumer.ftc.gov/articles/0382using-ip-cameras-safely', https://www.consumer.ftc.gov/articles/ 0382-using-ip-cameras-safely. Accessed: 2017-09-20.

Insecam - World biggest online cameras directory (n.d.), http://www.insecam. org/. Accessed: 2017-09-30.

Jerkins, J. A. (2017), Motivating a market or regulatory solution to iot insecurity with the mirai botnet code, *in* 'Computing and Communication Workshop and Conference (CCWC), 2017 IEEE 7th Annual', IEEE, pp. 1–5.

Kurento (2015), 'Architecture'. [Online; accessed December 05, 2017].
URL: *http://doc-kurento.readthedocs.io/en/latest/images/Architecture.png* Mahy,

R. et al. (2010), 'Rfc 5766-traversal using relays around nat'.

Mansfield-Devine, S. (2016), 'Ddos goes mainstream: how headline-grabbing attacks could make this threat an organisation's biggest nightmare', *Network Security* **2016**(11), 7–13.

Matthews, P., Mahy, R. and Rosenberg, J. (2010), 'Traversal using relays around nat (turn): Relay extensions to session traversal utilities for nat (stun)'.

McMillan, R. (2017), 'An unexpected security problem in the cloud: Misconfigured software and services are leading to accidental exposures of company data',

https://www.wsj.com/articles/ an-unexpected-security-problem-in-the-cloud-1505700061?mod= e2tw. Accessed: 2017-09-18.

Miller, B. A., Nixon, T., Tai, C. and Wood, M. D. (2001), 'Home networking with universal plug and play', *IEEE Communications Magazine* **39**(12), 104– 109.

Oxenhandler, D. (2003), 'Designing a secure local area network - sans institute', https://www.sans.org/reading-room/whitepapers/bestprac/ designing-secure-local-area-network-853. Accessed: 2017-10-4.

Patton, M., Gross, E., Chinn, R., Forbis, S., Walker, L. and Chen, H. (2014), Uninvited connections: a study of vulnerable devices on the internet of things (iot), *in* 'Intelligence and Security Informatics Conference (JISIC), 2014 IEEE Joint', IEEE, pp. 232–235.

Spring, T. (2017), 'Two popular ip cameras riddled with vulnerabilities', https://threatpost.com/ two-popular-ip-cameras-riddled-with-vulnerabilities/127172/. Accessed: 2017-09-20.

Tim Yeh, D. C. and Lu, K. (2017), 'Two popular ip cameras riddled with vulnerabilities', http:// blog.trendmicro.com/trendlabs-security-intelligence/ persirai-new-internet-things-iot-botnet-targets-ip-cameras/. Accessed: 2017-08-11.

Williams, R., McMahon, E., Samtani, S., Patton, M. and Chen, H. (2017), Identifying vulnerabilities of consumer internet of things (iot) devices: A scalable approach, *in* 'Intelligence and Security Informatics (ISI), 2017 IEEE International Conference on', IEEE, pp. 179–181.