Josh Gillman

CIT 480 - Senior Capstone

Cloud Based Security System

Research Topic – Encryption

For a security system, one of the biggest concerns is in the name itself – security, not only in the sense that the user is trying to secure a location, but in the actual security of the data being captured by the system. Because of this, we would like to include a secure way to transport the video captured by the camera using SSL encryption of the video data. According to its definition from info.ssl.com, SSL is defined as the standard security technology for establishing an encrypted link between a web server and a browser (What is SSL?). This link ensures that all data passed between the web server and browsers remain private and integral (What is SSL?). To be able to create an SSL connection a web server requires an SSL Certificate, and once a number of security questions are answered to securely identify the website and the company hosting it, the web server then creates two cryptographic keys - a Private Key and a Public Key (What is SSL?).

In order for us to do this, we will be installing OpenSSL on both our Raspberry Pi rig and our web based VM that will be hosting the website. According to Super User, OpenSSL is a powerful cryptography toolkit that can be used to benchmark computer speed and to encrypt files or messages – exactly what we need to do with the video captured (Super User). To do this, a number of steps will have to be taken on both the Pi and the VM in order to accomplish the installation and creation of encryption keys. The following commands can be entered into the Linux console to set up the keys:

$ sudo apt-get install openssl #to install openssl

Once installed, navigating to /usr/lib/ssl will bring up the configuration file where (as marked in the following example), each aspect of the setup is controlled from. A full example of this file is seen here:

####################################################################

[ CA\_default ]

dir = ./yourCA # Where everything is kept

certs = $dir/certs # Where the issued certs are kept

crl\_dir = $dir/crl # Where the issued crl are kept

database = $dir/index.txt # database index file.

#unique\_subject = no # Set to 'no' to allow creation of

# several ctificates with same subject.

new\_certs\_dir = $dir/newcerts # default place for new certs.

certificate = $dir/cacert.pem # The CA certificate

serial = $dir/serial # The current serial number

crlnumber = $dir/crlnumber # the current crl number

# must be commented out to leave a V1 CRL

crl = $dir/crl.pem # The current CRL

private\_key = $dir/private/cakey.pem# The private key

RANDFILE = $dir/private/.rand # private random number file

default\_days = 365 # how long to certify for

default\_crl\_days= 30 # how long before next CRL

default\_md = default # use public key default MD

preserve = no # keep passed DN ordering

####################################################################

[ req ]

default\_bits = 2048

default\_keyfile = privkey.pem

distinguished\_name = req\_distinguished\_name

attributes = req\_attributes

x509\_extensions = v3\_ca # The extentions to add to the self signed cert

####################################################################

[ req\_distinguished\_name ]

countryName = Country Name (2 letter code)

countryName\_default = DE

countryName\_min = 2

countryName\_max = 2

stateOrProvinceName = State or Province Name (full name)

stateOrProvinceName\_default = yourState

localityName = Locality Name (eg, city)

localityName\_default = yourCity

0.organizationName = Organization Name (eg, company)

0.organizationName\_default = yourCompany

organizationalUnitName = Organizational Unit Name (eg, section)

#organizationalUnitName\_default = -

commonName = Common Name (e.g. server FQDN or YOUR name)

commonName\_max = 64

emailAddress = yourEmail

emailAddress\_max = 64

From here, setting up a new certificate authority on the server is as simple as entering:

$ ./CA.sh -newca

$ ./CA.sh -newreq

$ ./CA.sh -sign

Similar steps will be subsequently be taken on the client, in this case our website VM, so that the encrypted video can be opened and viewed by the users.

Works Cited

"What Is SSL?". Info.ssl.com. N. p., 2017. Web. 17 Jan. 2017.

User, Super. "Using Openssl To Encrypt Messages And Files On Linux - Linuxconfig.Org". Linuxconfig.org. N. p., 2016. Web. 17 Jan. 2017.