Sigurnosni izazovi aplikativnog okruženja internet stvari

Sovilj, Nikola

Undergraduate thesis / Završni rad

2017

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Algebra University College / Visoko učilište Algebra**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:225:503859

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-03-28**

ALGEBRA VISOKO UČILIŠTE

Repository / Repozitorij:

Algebra Univerity College - Repository of Algebra Univerity College

zir.nsk.hr

dabar DIGITALNI AKADEMSKI ARHIVI I REPOZITORIJI

VISOKO UČILIŠTE ALGEBRA

ZAVRŠNI RAD

SIGURNOSNI IZAZOVI APLIKATIVNOG OKRUŽENJA INTERNET STVARI

Nikola Sovilj

Zagreb, svibanj 2017.

Student vlastoručno potpisuje Završni rad na prvoj stranici ispred Predgovora s datumom i oznakom mjesta završetka rada te naznakom:

„Pod punom odgovornošću pismeno potvrđujem da je ovo moj autorski rad čiji niti jedan dio nije nastao kopiranjem ili plagiranjem tuđeg sadržaja. Prilikom izrade rada koristio sam tuđe materijale navedene u popisu literature, ali nisam kopirao niti jedan njihov dio, osim citata za koje sam naveo autora i izvor, te ih jasno označio znakovima navodnika. U slučaju da se u bilo kojem trenutku dokaže suprotno, spreman sam snositi sve posljedice uključivo i poništenje javne isprave stečene dijelom i na temelju ovoga rada“.

U Zagrebu, datum.

Ime Prezime

Predgovor

Dajem zahvalnost svome mentoru dipl. ing. Silviu Papiću na strpljenju i pomoći tijekom izrade ovoga završnog rada.

Posebnu zahvalnost odajem svojoj obitelji koja mi je bila podrška tijekom studiranja.

Prilikom uvezivanja rada, Umjesto ove stranice ne zaboravite umetnuti original potvrde o prihvaćanju teme završnog rada kojeg ste preuzeli u studentskoj referadi

Sažetak

Ovaj završni rad dati će uvid u što Internet stvari je i kako se razvijao s vremenom. Biti će opisano što je to što ga čini kompleksnim konceptom, kakvi su izazovi koji se pojavljuju i kakva su rješenja razvijena koja su olakšala živote ljudima. Glavni cilj je pokazati kakve vrste napada su se dogodile u prošlosti i pružiti primjer rješenja koje može osigurati različite vrste IoT aplikacija.

Ključne riječi: Internet stvari, sigurnost.

ABSTRACT

This final paper will give an insight into what Internet of Things is and how it has developed over time. It will be described what is it that makes it a complex concept, what are the challenges that occur and what solutions have been developed that have made people's lives easier. Main goal is to show what kind of attacks have been done in the past and to provide an example solution that can secure various kinds of IoT applications.

Keywords: Internet of Things, security

Sadržaj

1. Uvod 1

2. Internet stvari 2

2.1. Povijesni pregled razvoja Interneta stvari 2

3. Komponente cjelovitog IoT pristupa 4

3.1. Senzori 4

3.2. Komunikacija 4

3.2.1. Komunikacija specifična za Internet stvari 6

3.3. Podaci 7

4. Područja primjene Interneta stvari 8

4.1. Pametna industrija 8

4.2. Automatizacija stambenih objekata 8

4.3. Pametna infrastruktura 9

4.4. Medicina 9

4.5. Automobilska industrija 10

5. Glavni izazovi u razvoju koncepata Internet stvari 11

5.1. Standardizacija 11

5.2. Upravljačke mogućnosti 11

5.3. Sigurnost 12

6. Primjeri napada koji su se ostvarili i posljedice 13

6.1.1. DDOS napadi 13

6.1.2. Napadi na medicinske uređaje 14

6.1.3. Napadi na sustave za kontrolu industrijskih postrojenja 14

6.1.4. Napadi u automobilskoj industriji 15

7. Testno okruženje za privremene mehanizme zaštite 16
7.1.1. Autorizacija jednim paketom 16
7.1.2. SSH Tuneliranje 18
7.1.3. Implementacija vatrozida 20
8. Opažanja i analiza rezultata 27
9. Preporuke za dizajn sigurnog okruženja Internet stvari 28
Zaključak 29
Popis kratica 30
Popis slika 31
Literatura 32

1. Uvod

Internet je već dugi niz godina služio kao medij za razmjenu informacija, omogućio je ljudima da postanu povezaniji tako što su se formirale zajednice putem različitih foruma. Ovime je Internet ostvario revoluciju koja dosada nije bila viđena, a to je nekontrolirani protok informacija tj. znanja koji je prije njegova postaojanja ljudima bio nezamisliv. Kompanijama je poslužio kao sredstvo pomoću kojeg prilaze korisnicima diljem svijeta i prate koliko su zadovoljni njihovim proizvodima. Osim velikih prilika koje su se pojavile kako običnim ljudima tako i kompanijama, Internet je uveo mnoge opasnosti. Pojavili su se računalni virusi koji su nastali iz radoznalosti i zabave kako bi se dokazalo da računala i mreže nisu sigurne, kasnije su se tokom godina su se razvijali u svrhu nanošenja sve veće štete kompanijama, ali i za ucjenjivanje pojedinaca tražeći otkupninu. Od tada se u IT sektoru pojavljuje potreba za sigurnosnim rješenjima koja će omogućiti neometan i siguran rad pojedinaca i kompanija. Međutim, s razvojem novih tehnologija sigurnost nikada nije na prvom mjestu već to mjesto zauzele su funkcionalnosti kako bi se tvrtka svojim proizvodom istaknula od svojih konkurenata. Stari pristup sigurnosti više ne funkcionira iz razloga što nove tehnologije uvode nove kompleksnosti koje dosada nisu istražene zbog čega je nemoguće tvrditi da je neki proizvod ili usluga bazirana na informacijskim tehnologijama u potpunosti sigurna. Koncept Internet stvari dosada se nije pokazao drugačijim i svakim danom sve više ukazuje da je potrebno primjeniti pristup prilikom samog dizajna.

Praktični dio ovog završnog rada u fokusu će imati zaštitu aplikacija kojima se pristupa s mreže što ne obuhvaća sve moguće vektore napada. Biti će prikazano kako primjeniti ideju o autorizaciji prema vatrozidu unutar Linux operacijskog sustava prije nego što je moguće ostvariti konekciju sa samom aplikacijom, što će hakerima otežati identifikaciju aplikacija tj. pokušaje skeniranja koji otkrivaju postojanje aplikacija. Osim napada na samu aplikaciju hakeri često dolaze do neovlaštenog pristupa uređajima kroz neke druge aplikacije ili servise koji nisu ažurirani i sadrže ranjivosti u sebi, a ovakvom zaštitom je moguće obuhvatiti i njih.

2. Internet stvari

Internet stvari (engl. *Internet of Things*) je koncept u kojem uređaji različitih namjena mogu komunicirati koristeći određeni protokol kako bi razmjenjivali podatke. Internet stvari primjenjiv je unuar više područja kao što su automobilska industrija, pametna industrija, pametni gradovi, automatizacija stambenih objekata i slično. Tradicionalno, ljudi su bili u interakciji s objektima i strojevima koji ih okružuju. Ovakva interakcija između ljudi i strojeva svakim se danom sve više smanjuje. Predmeti u fizičkom svijetu dobivaju logiku integriranu u sebe na temelju koje će se ponašati što smanjuje potrebu ljudskog djelovanja. Potreba za uvođenjem ovakvog koncepta svakim je danom sve veća jer omogućava da stvari koje grade određeni tip infrastrukture izmjenjuju podatke i informacije, što će rezultirati povećanim mogućnostima automatizacije u monogobrojnim industrijama. Osim u komercijalne svrhe, Internet stvari ima primjenu i u privatnim životima.

2.1. Povijesni pregled razvoja Interneta stvari

Kroz povijest viđene su 3 industrijske revolucije. Prva industrijska revolucija koja je obilježena korištenjem vodene pare za obavljanje mehaničkih zadataka u tvornicama. Otkriće i primjena električne energije u proizvodnim procesima pojavili su se u drugoj industrijskoj revoluciji početkom 20. stoljeća. Za vrijeme treće industrijske revolucije ekspanzija IT sektora omogućila je tranziciju s analogno mehaničkih tehnologija na digitalne tehnologije te automatizaciju proizvodnih procesa. Zadnjih par godina ekspanzijom Interneta stvari svijet je ušao u četvrtu industrijsku koja povezuje fizički i virtualni svijet. Međutim, Internet stvari nastao je puno ranije nego li je dobio pažnu i veliki interes kompanija koje proizvode pametne uređaje.

Uređaj koji se smatra prvim uređajem Interneta stvari bio je automat za prodaju pića. 1982. godine na sveučilištu Carnegie Mellon, automat je spojen na računalnu mrežu i imao je mogućnost generiranja izvještaja s popisom inventara.[1] U lipnju 2000. godine LG Electronics proizveo je prvi hladnjak koji je imao LAN port i mogućnost spajanja na IP

[1] https://en.wikipedia.org/wiki/Internet_of_things#History, 20.05.2017.

(engl. *Internet Protocol*) mrežu kojeg je razvijao od 1997. godine. [2] Kevin Ashton, koji je stvorio izraz Internet stvari, došao je na ideju razmjenjivanja podataka između stvari i uređaja pomoću RFID tehnologije koju je zamislio koristiti pri upravljanju lanca opskrbe tvrtke Procter & Gamble. Na Tehnološkom institutu Massachusetts pokrenuo je RFID istraživačku grupu pod imenom Auto-ID Center čiji je cilj stvoriti otvoreni standard RFID tehnologije i omogućiti širu primjenu. [3] Razvoj bar kodova i QR kodova pridonio je razvoju koncepta Interneta stvari jer omogućava da se podatak o nekom predmetu prenese očitanjem koda koji predmet nosi, bez potrebe za kompliciranim tehnološkim izvedbama na predmetu.

[2] http://talkincloud.com/cloud-computing/iot-past-and-present-history-iot-and-where-its-headed-today, 20.05.2017

[3] https://en.wikipedia.org/wiki/Kevin_Ashton, 20.05.2017.

3. Komponente cjelovitog IoT pristupa

Internet stvari je kompleksan koncept koji se sastoji od različitih hardverskih komponenti koje služe za prikupljanje podataka iz okoline, interakciju s okolinom, operacijskih sustava koji upravljaju čitavim hardverom i programskih jezika u kojima su aplikacije napisane. Komunikacija može biti ostvarena na više načina, postoje mnogobroji protokoli koji su definirani standardom, opće prihvaćeni i koji se smatraju zrelima budući da se već dulje vremena koriste. Osim navedenog, usluge u oblaku koje IoT (engl. *Internet Of Things*) aplikacije mogu koristiti su mnogobrojne.

3.1. Senzori

Senzori su komponente koje služe za prikupljanje podataka iz okoline kako bi se podaci kasnije obradili i kako bi se na temelju njih izvela neka radnja. Senzori mogu prikupljati podatke kao što su slike, zvuk, temperatura, vibracije, detektirati vlažnost zraka i plinove, osjetiti promjene pri ubrzanju i nagibu objekta. Količina dostupnih senzora osigurava veliki broj aplikacija i primjena kao npr. regulacija CO_2 emisija u tvornicama.

Najpopularnije platforme za brzo prototipiranje i testni razvoj su Raspberry Pi, Arduino, Beaglebone i brojne druge. Cijena ovih računala veličine kreditne kartice je relativno niska što ih čini pristupačnima te su postali popularni u školama u svrhu edukacije budućih generacija u eri Internet stvari.

3.2. Komunikacija

Komunikacija između uređaja može biti ostvarena na više načina. Prva podjela komunikacije je na žičanu i bežičnu komunikaciju. Žičana komunikacija pogodnija je za povezivanje većih stvari i stvari koje grade infrastrukturu, dok je za manje stvari pogodnije koristi bežične tenologije koje koriste vrlo malo snage kako bi ostvarili komunikaciju zbog manje potrošnje energije. Danas je Ethernet najkorišteniji standard za žičanu komunikaciju u lokalnim mrežama budući da je relativno jeftin, a postiže brzine prijenosa od 100 Mbps sve do 40 Gbps. Osim Etherneta koji koristi bakar kao prijenosni medij, korištenje optičkih niti koje prenose svjetlost umjesto napona koriste se u telekom industriji zbog još većih brzina prijenosa koje postižu. Bežične mreže imati će veću primjenu zbog jednostavnosti

povezivanja uređaja. Količina bežičnih tehnologija je velika i prilikom odabira odgovarajuće treba imati na umu kolika je potreba za napajanjem, udaljenost koju pokriva i sigurnost protokola.

Veliki je broj tehnologija i protokola koji se koriste u komunikaciji između uređaja. Najpopularniji je TCP/IP set protokola koji se koristi zadnjih 30-ak godina u računalnim mrežama koji omogućava komunikaciju pametnih telefona, računala, tableta i ostalih uređaja s mrežnim servisima poput Web poslužitelja, NTP poslužitelja, DNS poslužitelja i drugih. IP radi na mrežnom sloju koji osigurava da svaka poruka tj. paket koji treba biti poslan dospije do odredišta. Svaki uređaj u mreži ima jedinstvenu IP adresu na temelju koje se razlikuje od ostalih uređaja. Budući da IPv4 koristi samo 32 bita u adresiranju, ukupno je moguće imati 2^{32} tj. oko 4,3 milijardi jedinstvenih adresa dostupnih na Internetu. IPv6 je novija verzija Internet protkola koja koristi 128 bita za adresiranje što znači da je ukupan broj jedinstvenih adresa jednak 2^{128} te će zbog svoga kapaciteta omogućiti spajanje većeg broja uređaja na Internet. Svaka aplikacija koja želi ostvariti mrežnu konekciju mora od operacijskog sustava zatražiti slobodan TCP (engl. *Transmission Control Protocol*) ili UDP (engl. *User Datagram Protocol*) port uz koji će se vezati kako bi operacijski sustav, jednom kada primi podatke koje mu je proslijedila mrežna kartica, mogao odlučiti kojoj aplikaciji proslijediti podatke. TCP je protokol koji radi na transportnom sloju OSI modela kao i UDP. Za razliku od UDP-a, TCP je pouzdan protokol tj. garantira isporuku svih podataka između aplikacija koje komuniciraju, dok se UDP većinom koristi za komunikaciju koja zahtjeva obradu podataka u realnom vremenu, bez ponovnog slanja u slučaju da neki paket nije stigao na odredište.

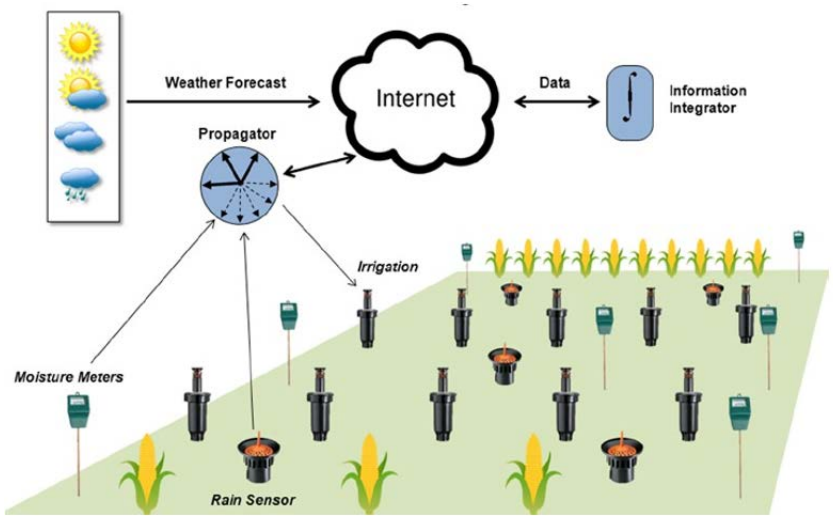
Slika 1 - Prikaz TCP/IP modela

Sloj podatkovne veze se kroz sve čvorove koji čine Internet izmjenjuje i specifičan je za fizičku izvedbu same konekcije. Jedan dio može biti ostvaren optičkom povezanošću, drugi dio bakreno, a ostatak bežično. Iz ovog razloga Internet protokol, koji se radi iznad sloja podatkovne veze, potpuno je neovisan o mediju koji se koristi za prijenos.

Aplikacijski sloj određuje kakva je komunikacija između samih aplikacija koje razmjenjuju podatke i u kontekstu Interneta stvari može biti vrlo specifičan i nestandardiziran.

3.2.1. Komunikacija specifična za Internet stvari

Arhitektura mreža koja se koristi za komunikaciju uređaja koji čine Internet stvari ne zahtjeva uvijek povezanost Internet protokolom budući da neki od uređaja imaju ograničenu procesorsku moć pa je zbog toga implementacija cijelog seta TCP/IP protokola izostavljena. U ovakvim slučajevima koriste se uređaji koji imaju ulogu propagatora. Propagator funkcionira tako što ima povezanost Internet protokolom prema ostatku mreže s jedne strane , a s druge strane povezanost sa senzorima. Nakon što su podaci prikupljeni sa skupine senzora pomoću nekog manje zahtjevnog protokola koji osigurava manju potrošnju energije, propagator u njihovo ime šalje te podatke u informacijski integrator tj. bazu podataka.[4]



Slika 2 – Prikaz IoT mrežne arhitekture[5]

WSN (engl. *Wireless Sensor Network*) je naziv za ovakvu mrežnu arhitekturu sačinjenu od više bežičnih senzora koji prikupljaju podatke i šalju ih pomoću propagatora.

[4] DACOSTA, F. Rethinking the Internet of Things, Apress, 2014, str. 17-21
[5] DACOSTA, F. Rethinking the Internet of Things, Apress, 2014, str. 38

3.3. Podaci

Podatak je jednostavna neobrađena misaona činjenica koja ima neko značenje. Podaci su znakovni prikaz činjenica i pojmova koji opisuju svojstva objekta i njihovih odnosa. Obrada podataka je proces pretvorbe podataka u informacije. Da bi podatak postao informacija mora zadovoljiti dva uvjeta. Prvi uvjet je da podatak mora biti unutar nekoga konteksta kako bi se smatrao informacijom, a drugi uvjet je taj da za primatelja mora nositi novost.[6]

Internet stvari će svojom primjenom generirati ogromnu količinu podataka. Obradom podataka biti će moguće otkriti kako unaprijediti postojeće usluge ili pružiti nove usluge koje će postati personaliziranije tj. biti će prilagodljive i optimizirane za potrebe pojedinca. Poduzeća će imati veliku korist od Interneta stvari kao generatora velikih podataka. Analitika postaje svakodnevnica svake ozbiljnije tvrtke kako bi se lakše prilagodila okruženju unutar kojeg se nalazi. Karakteristika podataka koje generiraju senzori je da su vrlo mali, međutim količina senzora koja se koristi za prikupljanje je velika što dovodi do potrebe da se podacima pristupa organizirano. Uloga integratora je prikupljati, filtrirati, spremati i analizirati podatke koje su senzori generirali i omogućiti pristup tim podacima da ih koriste treće strane u svrhu izgradnje novih usluga što znači da dobiva ulogu posrednika podataka unutar IoT ekosustava. Integratori se stoga mogu podijeliti u dvije glavne skupine, javne kako bi podaci bili dostupni svima i privatne gdje se mogu pojaviti novi poslovni modeli.

[6] https://hr.wikipedia.org/wiki/Podatak,_informacija,_znanje,_mudrost, 27.08.2017.

4. Područja primjene Interneta stvari

Internet stvari može biti široko primjenjen i nije usko vezan uz određenu industriju. Tehnologija koja danas postoji može se iskoristiti u svrhu izgradnje novih proizvoda koji imaju namjenu ubrzati razmjenu informacija između različitih procesa, kako proizvodnih tako i onih koji uključuju ljudsku interakciju s fizičkim svijetom. Glavni cilj je pomoću senzora prikupiti što veći uzorak određenih podataka koji će poslužiti u obradi određenim aplikacijama.

4.1. Pametna industrija

Internet stvari će svojom primjenom dodati nove mogućnosti u tvornice čime će olakšati procese proizvodnje i nadzora. Primjena Interneta stvari u novim proizvodima i uslugama tek će isploviti u nadolazećim godinama i neće izostaviti niti jednu industrijsku granu. Neki od razloga za primjenom su optimizacija internih troškova, lakše upravljanje proizvodnim procesima, kontrole kvalitete i slično.

4.2. Automatizacija stambenih objekata

Od 1900. do 1920. pojavili su se prvi kućni uređaji koji nisu bili „pametni“, ali za svoje doba predstavljali su veliko dostignuće. Prvi usisavač s motorom pojavio se 1901. godine, a šest godina kasnije proizveden je električni usisavač. Osim usisavača u ovom razdoblju proizvedeni su frižideri, slušilice veša, glačala, mašine za pranje i ostalo. ECHO IV je prvi pametni uređaj koji se pojavio 1966, iako nije tako prepoznat. Mogao je računati, regulirati kućnu temperaturu te paliti i gasiti uređaje. Godinu dana kasnije pojavio se i „The Kitchen Computer“. [7]

Izraz „Internet stvari“ zaživio je početkom 21. stoljeća kada se popularnost pametnih kuća i automatizacije pametnih domova povećala. Pametni domovi većinom su imali funkcionalnosti kojima se mogla regulirati rasvjeta, termostati, udaljeno pregledavati sigurnosne kamere i slično. Cijena takvih uređaja bila je relativno pristupačna, a količina konkurencije velika. Veliki broj manjih kompanija u području pametnih domova akviziran

[7] http://www.iotevolutionworld.com/m2m/articles/376816-history-smart-homes.htm, 23.05.2017.

je od strane internetskih divova. Nest Labs, kao jedan od poznatijih proizvođača termostata koji imaju mogućnost učenja tj. pametnog prilagođavanja temperature, akvizirao je Google za 3,2 milijarde dolara 2014. godine.

4.3. Pametna infrastruktura

Uvođenjem tehnologije u svrhu povezivanja infrastrukture unutar grada građani će imati povećani standard života. Primjena M2M (engl. *machine to machine*) komunikacije može se ostvariti u poboljšanju transportnih sustava, povećanju energetske učinkovitosti, smanjenju onečišćenja okoliša i slično. Pametne gradove je moguće unaprijediti pomoću dobre organizacije prometnica i mehanizama za detekciju i upravljanje zagušenjima.

U Singapuru je 1970-tih godina pokrenuta inicijativa koja je riješila problem zagušenosti u prometu čime je smanjeno prosječno vrijeme kretanja automobila po glavnim cestama. Krajem 2014. godine premijer Singapura, Lee Hsien Loong pokrenuo je program za postavljanje dodatnih senzora i kamera u svrhu izgradnje sustava koji prate sve od čistoće grada do prometnica. [8]

4.4. Medicina

U medicini je već dugi niz godina popularna primjena uređaja koji mogu pratiti razna stanja pacijenata poput srčanih aktivnosti, moždanih aktivnosti i slično. Uz nove tehnološke inovacije zdravstvo očekuje udaljeno praćenje stanja pacijenata i telemedicina će kroz svoju primjenu olakšati pacijentima obavljanje pretraga tako što neće morati stalno fizički biti prisutni u bolnici. Ovakav pristup također će olakšati zdravstvene usluge u područjima koja su slabije naseljena i nemaju bolnice u blizini. Izdavanje recepata za lijekove elektroničkim putem jedan je od načina na koji se pacijenta uvodi u zdravstvene ustanove u elektroničkom obliku što se može iskoristiti za ostali niz usluga koje su bazirane na takvom elektroničkom identitetu, a što također olakšava praćenje pacijenata između različitih bolnica.

Podaci generirani pomoću senzora za praćenje stanja pacijenata omogućiti će bolnicama da pohranjuju informacije o različitim bolestima, uspoređuju sličnosti među pacijentima s

[8] http://www.ioti.com/smart-cities/world-s-5-smartest-cities, 23.05.2017.

istim bolestima i uvesti će primjenu strojnog učenja kako bi se bolesti uz veću preciznost dijagnosticirale.

4.5. Automobilska industrija

Automobilska industrija je jedna od industrija koja trenutno najviše i najbrže adaptira nove tehnologije kako bi omogućila nove funkcionalnosti. Autonomna vozila postaju sve popularnija s ciljem smanjenja prometnih nesreća budući da automobili budućnosti neće trebati ljudsku interakciju kako bi se njima upravljalo već će samostalno voziti što primjena kombinacije tehnologija poput GPS-a, radara, lidara, računalnog vida i strojnog učenja omogućava.

Postoje 3 vrste komunikacije automobila:

- V2V - vozilo s vozilom
- V2C - vozilo s oblakom
- V2I - vozilo s infrastrukturom

Vozilo s vozilom komunikacija može se koristiti u svrhu izbjegavanja kolizije, obavještavanju o kretanju u prometu i slično.

Komunikacija vozilo s oblakom najčešće podrazumjeva konekcije koje se koriste za preuzimanje proizvođačevih ažuriranja ili za prijenost podataka koje proizvođač koristi kako bi uveo dodatne usluge, npr. prikupljanje podataka o trenutnoj lokaciji i vremenskim uvjetima kako bi se obavijestila ostala vozila koja se kreću prema toj lokaciji u slučaju vremenskih nepogoda.

Komunikacija između vozila i infrastrukture odnosi se na interakciju između vozila i prometa, pojedinih dijelova grada, parkinga i slično.

5. Glavni izazovi u razvoju koncepata Internet stvari

Internet stvari je relativno novi koncept i ne postoje odgovori kako točno obaviti dizajn nekog rješenja ili proizvoda budući da je specifičan za svoju primjenu. Većina nedostataka i propusta uočava se tek s vremenom nakon čega slijede poboljšanja koja kasnije postaju novi referentni modeli.

5.1. Standardizacija

Jedan od glavnih izazova Interneta stvari je standardizacija. Svaki proizvođač nekog proizvoda može koristiti različite standarde čime narušava kompatibilnost s ostalim proizvodima. Međutim ovo je pitanje uporabljivosti i jedan od načina na koji proizvođači osiguravaju interoperabilnost samo između svojih proizvoda.

Implementacija starih standarada, koji imaju sigurnosne propuste, unutar proizvoda ostavlja ranjivosti na dug period pri čemu nije moguće obaviti ažuriranje kako bi se ranjivost popravila. Standardi koje Internet stvari koristi trebaju omogućiti jednostavna sučelja za komunikaciju među uređajima uzimajući sigurnost u obzir.

Osim standarada, regulacija će biti potrebna kako bi se osigurala ponuda proizvoda na tržištu koji garantiraju sukladnost sa zahtjevima i određenu razinu sigurnosti.

5.2. Upravljačke mogućnosti

Broj uređaja koji čine Internet stvari je velik i jedan od izazova je kako svima njima upravljati. Postojeći protokoli koji se koriste za upravljanje mrežnom konfiguracijom su dovoljni u ograničenim slučajevima. DHCP (engl. *Dynamic Host Configuration Protocol*) je jedan od takvih protokola koji služi automatsku dodjelu IP adresa uređajima koji se pojave kao novi sudionici unutar mreže. Ovako je izbjegnuta potreba za ručnim podešavanjem mrežne konfiguracije svakog novog uređaja koji se pojavi u mreži. Upravljanje i nadzor potrebno je odraditi kroz zasebne centralizirane aplikacije koje su razvijane za uređaje čije mogućnosti prepoznaju budući da su namjene i mogućnosti tih uređaja različite.

5.3. Sigurnost

Postoji više aspekata sigurnosti kod Interneta stvari koje treba uzeti u obzir i ne postoji jedno rješenje i pristup koji vrijedi za sve uzevši u obzir različite tehnologije i način na koji su implementacije ostvarene. U obzir treba uzeti sigurnosti mreže preko koje senzori komuniciraju, fizičku sigurnost samih senzora i uređaja, sigurnost infrastrukture ili usluga u oblaku trećih strana koje iznajmljuju infrastrukturu. Osim navedenog u pitanje dolazi i privatnost podataka samih korisnika.

Jedan od razloga zašto je sigurnost uređaja i mreža ugrožena je sve veća dostupnost alata za penetracijsko testiranje kao što je Metasploit koji sadrži bazu poznatih ranjivosti i maliciozan kod koji ih iskorištava. Kali Linux je jedna od Linux distribucija koja sadrži gotov set alata za prikupljanje informacija, analizu ranjivosti, napade na bežične mreže i slično.

Kako bi proizvodi u budućnosti bili sigurni, tvrtke koje ih proizvode trebaju postati softverske kompanije tako što će ulagati u obrazovanje vezanim uz informacijske tehnologije budući da nemaju dovoljno iskustva s razvojem softvera i implementacije ostatka informacijskih tehnologija koliko imaju veliki tehnološki divovi kao što je Google.

Tvrtke očekuju novi izazovi poput BYOD (engl. *Bring Your Own Device*) čime se narušava sigurnost budući da zaposlenici donose i koriste vlastite uređaje u korporativnim mrežama koji mogu biti zaraženi i poslužiti kao novi vektor napadačima u maliciozne svrhe. Zbog ovakvog izazova pojavile su se tvrtke koje proizvode mrežne uređaje u svrhu kontrole pristupa mreži. Takvi uređaji mogu odobriti ili odbiti pristup mreži bazirano na politikama tj. pravilima koje zahtjevaju određeno stanje uređaja, npr. u slučaju da uređaj ima preuzeta zadnja ažuriranja može pristupiti mreži.

Ponašanje uređaja u nedefiniranim uvjetima može biti opasno po život. Jedan od primjera su autonomna vozila koja sama donose odluke o vožnji i upravljaju sami sobom, što podrazumjeva da su sve situacije u prometu unutar kojih se vozilo može naći predviđene.

6. Primjeri napada koji su se ostvarili i posljedice

Količina napada svakim je danom sve veća, a sigurnosne mjere koje se trebaju poduzimati su kompliciranе i veliki broj korisnika ne posjeduje potrebna znanja. Proizvođači ne poduzimaju dovoljno mjera kako bi zaštiti svoje proizvode i time spriječili da korisnici postanu žrtve malicioznih napada. Svaki od napada može nositi velike posljedice na infrastrukturu i ono što je najbitnije, ljudske živote. Kroz iduće primjere biti će navedeni i pojašneni različiti tipovi napada koji su se dosada ostvarili.

6.1.1. DDOS napadi

DDOS (engl. *Distributed Denial of Service*) je vrsta napada koja koristi veliki broj zaraženih računala (engl. *botnet*) kako bi uskratil pristup nekom Internet servisu učestalim ponavljanjem zahtjeva za neki od resursa koji servis nudi. Ovakva vrsta napada bila je ostvarena pomoću računala, a zadnjih par godina napadi su se počeli izvršavati pomoću ostalih uređaja povezanih na Internet kojih je svakim danom sve više.

Mirai je jedan od primjera koji je koristio IP kamere i usmjernike (engl. *router*) kako bi onemogućio pristup stranici novinara Brian Krebs-a koji se bavi računalnom sigurnošću. Maliciozni kod je funkcionirao tako da je skenirao Internet kako bi pronašao uređaje koji su ranjivi tj. uređaje koji su koristili predefinirana korisnička imena i lozinke za udaljeni pristup. Mrežni resursi zaraženih uređaja bili su korišteni čime je bilo generirano 1 Tbit/s prometa prema francuskoj web hosting kompaniji u drugom napadu. [9]

DDOS napadi jedni su od najučestalijih napada. Osim napada na web stranice, DDOS napadi mogu imati izrazito negativan utjecaj u slučaju napada na dio mrežne infrastrukture koji je zadužen za neki proizvod, npr. usluge u oblaku koje služe za pohranu podataka. Uređaji koji koriste takve usluge postaju neupotrebljivi jednom kada je neka od usluga u oblaku napadnuta.

[9] https://en.wikipedia.org/wiki/Mirai_(malware), 12.06.2017.

6.1.2. Napadi na medicinske uređaje

Medicina kao grana koja će biti pod rastućim utjecajem razvoja IoT koncepta je daleko najugroženija. Razina sigurnosti postojećih sustava u medicini treba doseći veću razinu zbog činjenice da su podaci koji takvi sustavi čuvaju bitni, ne smiju biti izgubljeni u slučaju napada koji zahtjevaju otkupninu (engl. *ransomware*) niti smiju biti otkriveni budući da mogu biti zlouporabljeni.

Najčešći oblici kompromitiranja informacijskih sustava je bio pomoću spear phishing napada. Takva vrsta napada izvodi se tako što žrtva otvara elektroničku poštu koja zavarava žrtvu budući da elektronička pošta izgleda kao da dolazi s poznatog izvora. Sadržaj može izgledati originalno i često koristi oslovljavanje osobe koja prima poštu tako da žrtva povjeruje da je elektronička pošta legitimna, a zapravo sadrži maliciozni privitak. Maliciozni kod koji se nalazi unutar privitka može poslužiti u otvaranju stražnjeg ulaza i tako dalje omogućio napadaču širanje po ostatku mreže.

Uređaji koji su u samom kontaktu s ljudima također mogu biti ugroženi. Pejsmejkeri se smatraju nesigurnima jer uređaji koji služe njihovom podešavaju (pejsmejker programeri) ne koriste lozinke kako bi se zaštitili od neovlaštene uporabe niti ne koriste nikakvu formu autentikacije samome pejsmejkeru, većina pejsmejker programera će raditi s ostalim pejsmejkerima ukoliko je i jedan i drugi uređaj proizveo isti priozvođač. Ovo je problem jer pejsmejker programeri više ne zahtjevaju malu udaljenost od pejsmejkera kako bi komunicirali, a napadači mogu čitati podatke koje pejsmejker sadržava ili ga reprogramirati.[10]

6.1.3. Napadi na sustave za kontrolu industrijskih postrojenja

Stuxnet je maliciozni program koji je napadao Siemens PLC uređaje. Tražio bi PLC uređaje nakon što bi se nastanio na računala s Windows operacijskim sustavom. Računalna mreža preko kojih su računala i PLC uređaji komunicirali bila je fizički odvojena od ostatka Interneta i drugih računalnih mreža kako bi bila zaštićena. Usprkos tome napad je proveden pomoću USB uređaja za pohranu podataka koji je sadržavao maliciozni kod. Maliciozni program napadao je PLC uređaje kako bi pomoću njih povećao broj okretaja ventilatora što je dovelo do raspada konstrukcije koja se nalazila oko ventilatora zbog jakih vibracija.

[10] http://blog.whitescope.io/2017/05/understanding-pacemaker-systems.html, 15.08.2017.

6.1.4. Napadi u automobilskoj industriji

U posljednjih nekoliko godina automobilska industrija bila je istraživana kako bi se otkrilo koliko su novi automobili nesigurni. U tim istraživanjima otkrivene su razne ranjivosti koje su napadačima omogućile udaljeno upravljanje automobilima kao što je aktiviranje kočnica, upravljanje volanom i preuzimanje ostalih kontrola koje omogućavaju upravljanje automobilom.

Jedan od najpopularnijih električnih automobila danas, Tesla Model S, pokazao se nesigurnim. Istraživači iz područja sigurnosti uspjeli su preuzeti glavne upravljačke funkcije nad automobilom poput zaključavanja i otključavanja vrata i pokretanje u zaustavljanje automobila. Dvije od tri komponente auta supješno su napadnute, ploča s instrumentima i ploča središnjeg prikaza informacija. Obje komponente koristile su Linux operacijski sustav. Fizički pristup omogućio je pronalazak memorijskih kartica koje su sadržavale ključeve i certifikate koji su se koristili za konfiguraciju VPN tunela koristeći OpenVPN, 4 pinski konektor na koji se moglo spojiti modificirajući CAT 5 kako bi se omogućio pristup lokalnoj mreži. Dva servisa dostupna s lokalne mreže bila su zastarjela te su sadržavali ranjivosti, dnsmasq i mini httpd. Koristeći pronađene ključeve za konfiguraciju VPN tunela istraživači su došli u mogućnost preuzimanja nadogradnji i sigurnosnih tokena koji su se preuzimali svakih 24 sata. Korisnički račun tesla1 na ploči središnjeg prikaza informacija koristio je sigurnosni token kao lozinku pri čemu se podatak o tokenu spremao nekriptiran i na ploču s instrumentima. Lokalna datoteka shadow u koju se spremaju kriptirane lozinke lokalnih korisnika sadržavala je kratke i jednostavne lozinke koje je lako pogoditi. X11, sustav za iscrtavanje prozora kao osnovni grafički element, moguće je napasti tako da prikazuje proizvoljne slike ili onemogućiti njegov rad za vrijeme vožnje. Promatrajući mrežni promet koji služi za signalizaciju i upravljanje među uređajima unutar automobila istraživači su otkrili koji paketi sadrže kontrolne pozive. Jednom kada je kontrolni poziv uspješno izveden uređaj koji je posrednik će nastaviti komunikaciju na CAN mreži s komponentama koje trebaju obaviti neku funkciju, primjerice zaključavanje automobila. Ranjivosti su zakrpane prilikom OTA (engl. *Over-The-Air)* ažuriranja, koji je jedan od mehanizama koji se preporuča da ga slijede svi proizvođači u automobliskoj industriji kako bi mogli relativno brzo primijeniti sigurnosna ažuriranja.[11]

[11] https://blog.lookout.com/hacking-a-tesla

7. Testno okruženje za privremene mehanizme zaštite

U sljedećem dijelu biti će pojašnjeni neki od sigurnosnih mehanizama koji imaju ulogu u zaštiti komunikacija između uređaja. Mogu biti implementirani na razini mrežne infrastrukture ili s razine samog uređaja.

7.1.1. Autorizacija jednim paketom

Prije objašnjena autorizacije jednim paketom biti će pojašnjeno što je vatrozid i preteće mehanizmu autorizacije jednim paketom.

Vatrozid (engl. *firewall*) je sigurnosni sustav čija je namjena štititi uređaje na mreži od ostvarivanja neželjenih konekcija. U mrežnim komunikacijama implementiran je kao posebni mrežni uređaj ili programski. Većina današnih operacijskih sustava ima svoju programsku implementaciju vatrozida dok se zasebni mrežni uređaji za tu svrhu koriste u većim okruženjima i štite više uređaja. Vatrozid funkcionira tako što koristi niz pravila koja definiraju koji uređaji smiju komunicirati međusobno i po kojim portovima. Portovi su dodjeljeni aplikacijama čime broj porta predstavlja aplikaciju, npr. web server koristi port 80 koji spada u skupinu dobro poznatih portova koji seže od 0 do 1024. Ostale aplikacije mogu koristiti neki drugi port koji može spadati u skupinu dobro poznatih portova ili neke ostale od 1025 do 65535. Vatrozid ima ulogu propuštati pakete prema određenim portovima ili ih odbacivati uz različiti uvjete. Odluku o tome da li propustiti promet može donositi temeljem izvorišne IP adrese ili izvorišne mreže unutar koje je uređaj koji pokušava ostvariti komunikaciju.

```
Chain INPUT (policy DROP)
target     prot opt source               destination
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:http

Chain FORWARD (policy DROP)
target     prot opt source               destination

Chain OUTPUT (policy DROP)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere             ctstate NEW,RELATED,ESTABLISHED
```

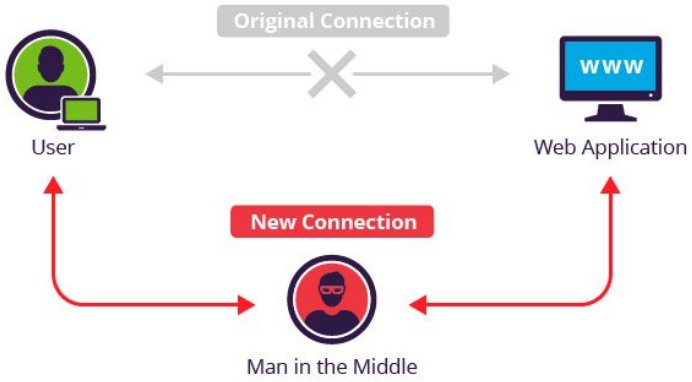
Slika 3 - Primjer pravila iptables vatrozida za Linux operacijskom sustavu

Slika 1 prikazuje jednostavan primjer pravila vatrozida na Linux operacijskom sustavu. Vatrozid će u ovom slučaju prihvaćati ulazni promet na portu 80 tj. sav HTTP (engl. *Hyper*

Text Transfer Protocol) promet, prosljeđivanje se neće obavljati i sve izlazne konekcije i konekcije koje su inicirane po portu 80 biti će propuštene.

Vatrozid je jedan od mehanizama zaštite, međutim skeniranje portova kao ključni korak prilikom pokušaja hakiranja služi za otkrivanje servisa i aplikacija s kojima je moguće ostvariti komunikaciju. Najpoznatiji alat je nmap koji se koristi u svrhu skeniranja portova. Pravila vatrozida podešena su tako da bilo tko može ostvariti komunikaciju s određenim aplikacijama ili servisima čime je moguće vršiti skeniranje portova. Jednom kada napadač obavi skeniranje može dalje izvršiti napad koristeći objavljene programe koji iskorištavaju poznate ranjivosti ili sam može otkriti postoji li ranjivost i iskoristiti ju.

Kako bi se aplikacije i servisi dodatno zaštitili administratori koriste metodu kucanja na portove (engl. *port knocking*). Port knocking je implementiran tako da se eksterno s nekog uređaja šalje određena sekvenca spajanjem na portove, npr. 80, 8080 i 65 koji se ne koriste nakon čega bi program na strani kojoj se pokušava pristupiti detektirajući takvu sekvencu dodao pravilo u vatrozid koje će omogućiti pristup portu npr. 1491. Sve do trenutka dok se točna sekvenca ne pošalje port 1491 nije dostupan čime je aplikacija ili servis koji radi na tom portu siguran od pokušaja otkrivanja jer je vatrozid zatvoren. Port knocking kao dodatni mehanizam zaštite napadačima otežava pronalazak otvorenih portova i aplikacija ali nije najsigurnije rješenje. Port knocking je ranjiv iz razloga što prilikom izvršavanja MITM (engl. *Man in the Middle*) napada, tj. napada u kojem maliciozni korisnik presreće sav promet tako što se postavlja kao posrednik između dvije strane koje komuniciraju. Napadač tako može vidjeti portove koji se koriste za autentikaciju vatrozidu i ponovno ih generirati.



Slika 4 - Prikaz MITM napada[12]

[12] https://www.incapsula.com/images/illustrations/web-app-security-mini-site/man-in-the-middle-mitm.jpg, 02.08.2017.

Autorizacija jednim paketom (engl. *Single Packet Authorization*) koristi sličan princip u kojem je dovoljan jedan paket kojim se obavlja autentikacija i nakon uspješne autentikacije odvija se autorizacija tako da se dodavanjem novog pravila u konfiguraciji vatrozida odobrava komunikacija. Paket nije moguće ponoviti niti usred MITM napada jer glavna ideja autorizacije jednim paketom je korištenje MAC (engl. *Message Authentication Code*) i enkripcije. MAC osigurava autentičnost i integritet poruke dok enkripcija omogućava povjerljivost tj. tajnost sadržaja.

Korištenje autorizacije jednim paketom kao dodatnog mehanizma zaštite osigurava mrežnu komunikaciju čime napadača tjera na eksploataciju nekih od implementacija autorizacije jednim paketom kako bi kasnije mogao vršiti skeniranje portova i otkrivanje servisa i aplikacija koji su dostupni.

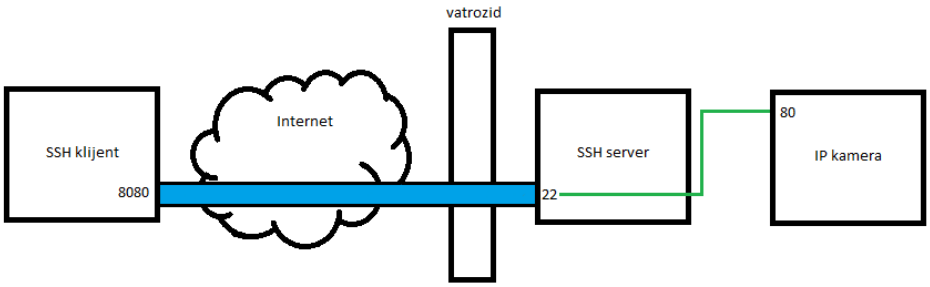
7.1.2. SSH Tuneliranje

SSH je mrežni protokol koji koristeći kriptografiju osigurava mrežni pristup u mrežama koje nemaju nikakve sigurnosne mehanizme poput enkripcije. Trenutna verzija SSH protokola je 2.0. Zadani port koji koristi je 22.

Najčešći način korištenja SSH protokola je u svrhu administriranja udaljenih računala izvršavajući naredbe iz ljuske operacijskog sustava čime su Telnet i rlogin protokoli postali zastarjeli budući da nisu bili toliko sigurni tj. nisu omogućavali tajnost podataka koji su se prenosili mrežom. SSH može se koristiti za siguran prijenos podataka, montiranje direktorija s udaljenih računala koristeći SSHFS čime se dobiva udaljen pristup spremištu podataka, prijenos grafičkog prikaza aplikacija pokrenutih na udaljenim poslužiteljima, tuneliranje i ostale napredne mogućnosti.

SSH tuneliranje je mogućnost koju OpenSSH, najpopularnija implementacija SSH protokola danas, pruža. U osnovi tuneliranje znači obavljanje enkapsulacije još jednom na postojeću konekciju čime se dodaje još jedan sloj koji obavlja enkripciju. SSH tunel je uspostavljen između klijenta i servera i svaka konekcija koja je prosljeđena kroz tunel je dodatno enkapsulirana unutar samog SSH protokola čime dobiva zaštitu budući da SSH obavlja enkripciju, provjeru integriteta podataka tj. provjeru da li su podaci ostali u stanju u kakvom ih je pošiljatelj poslao i autentičnost koja potvrđuje da su podaci izmjenjeni između izvora koji stvarno je onakav kakvim se predstavlja.

Tuneliranje je korisno iz razloga što dio konekcije koji je ostvaren preko Interneta ima zaštitu od MITM napada i ostalih napada kao npr. napad u kojem je određena sekvenca ponovljena (engl. *replay attack*) koja sadrži neki bitan parametar kao što je lozinka ili informacije o korisničkom računu. SSH tuneliranje također se može koristiti kako bi osigurao pristup bitnim resursima koji nemaju ugrađenu enkripciju preko javnih i nesigurnih mreža koje se mogu koristiti u hotelima, kafićima, aerodromima i sl.



Slika 5 – Tuneliranje koristeći SSH protokol

Naredba izvršena na ssh klijentu:

ssh -L 8080:ipkamera.exampledomain.com:80 sshserver.exampledomain.com

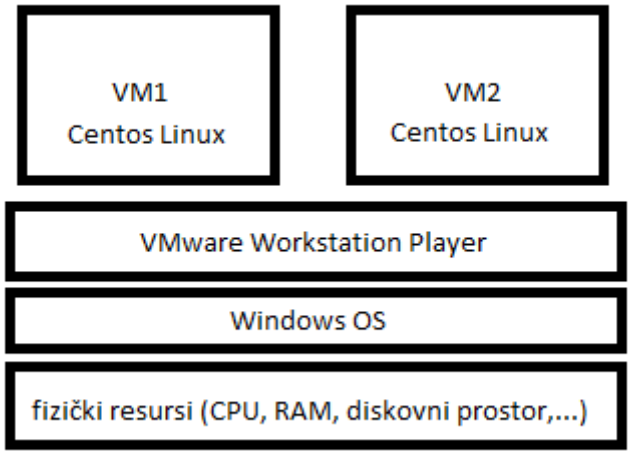
uspostaviti će tunel sa SSH serverom tako da svaki put kada se na računalu nazvano SSH klijent pokuša otvoriti lokalni port 8080 će prije nego dođe do destinacije koja je IP kamera, konekcija biti kriptirana između SSH klijenta i SSH servera, a ostatak konekcije koji je dio lokalne mreže na kojoj se nalaze SSH server i IP kamera biti će prosljeđen sa SSH servera. Kako bi navedeni primjer funkcionirao na vatrozidu treba biti konfigurirano prosljeđivanje portova (engl. *port forwarding*) kako bi sav promet koji dolazi na vatrozid na port 22 bio prosljeđen SSH serveru na port 22.

SSH tuneliranje moguće je koristiti u kombinaciji s autorizacijom jednim paketom kako bi se port koji je na strani SSH servera dodatno zaštitio

7.1.3. Implementacija vatrozida

7.1.3.1 Uvod i priprema okruženja

Implementacija vatrozida biti će prikazana kroz primjer u virtualiziranom okruženju s dvije virtualne mašine koje koriste Linux operacijski sustav. Jedna virtualna mašina pokušati će pristupiti mrežnim resursima druge virtualne mašine koristeći autorizaciju jednim paketom.



Slika 6 - Virtualna okolina

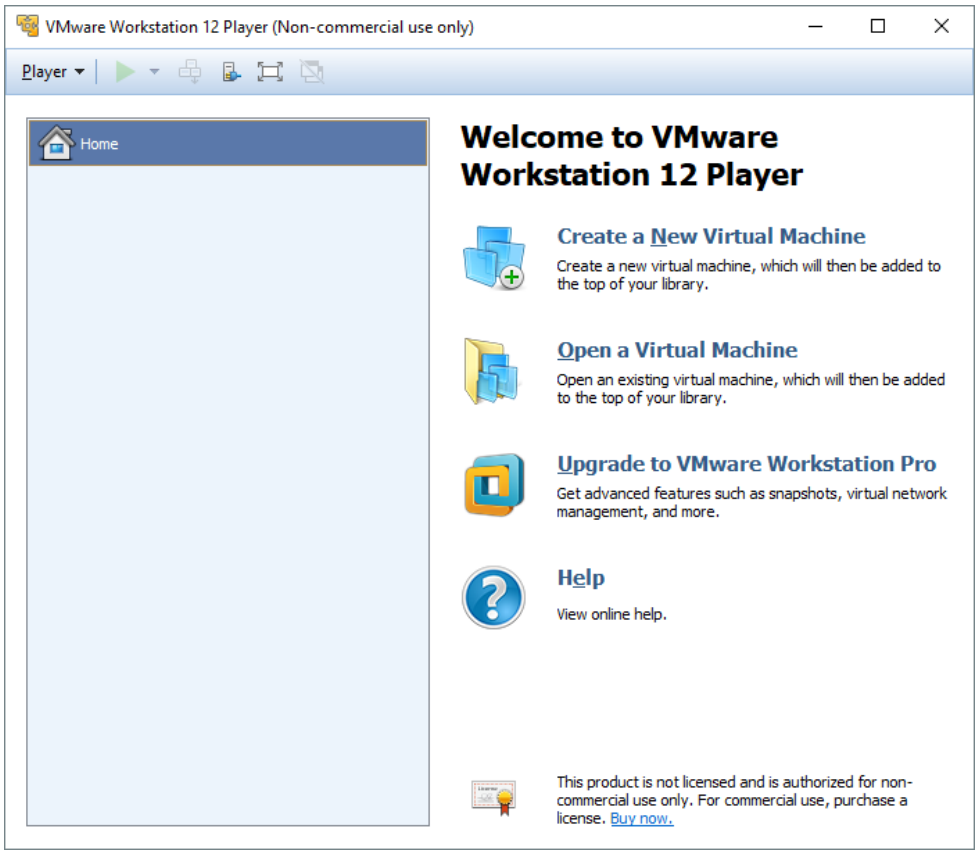
Slika virtualna okolina prikazuje okružje koje će biti pripremljeno u nastavku.

7.1.3.2 Instalacija hipervizora

Virtualizacija je tehnologija koja omogućava da više operacijskih sustava koristi jedno fizičko računalo. Programska podrška koja ovo omogućava zove se hipervizor. U primjeru će se koristiti VMware Workstation Player koji je besplatan i čije su mogućnosti za potrebe ovog završnog rada dovoljne.

Na https://www.vmware.com/products/player/playerpro-evaluation.html je moguće preuzeti instalaciju za VMware Workstation Player.

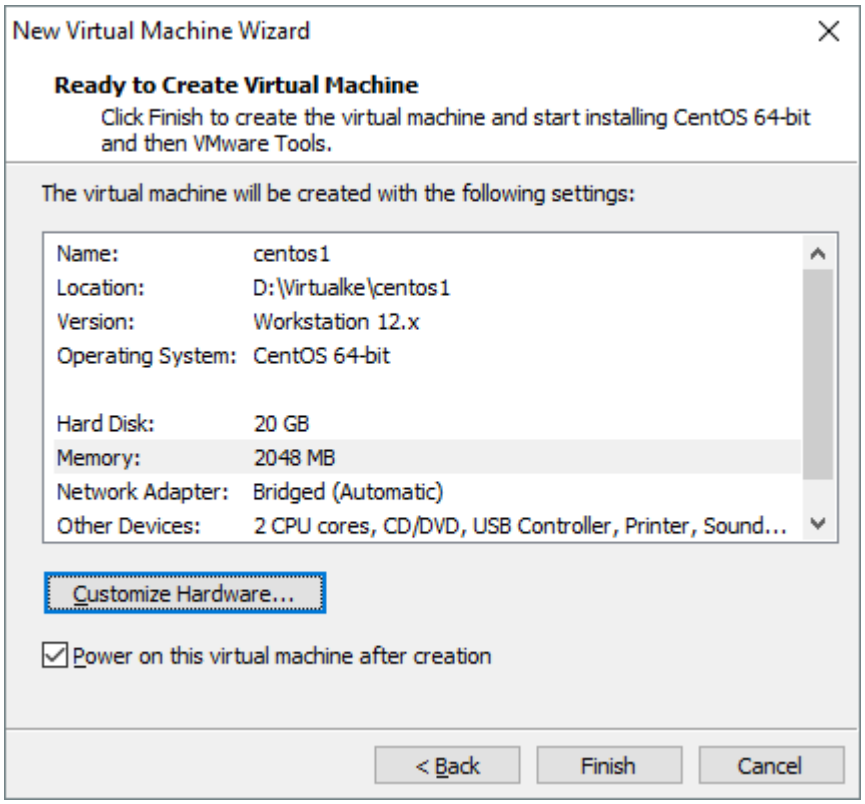
Instalaciju je potrebno pokrenuti s administratorskom razinom prava kako bi proces instalacije počeo. Prilikom instalacije potrebno je prihvatiti uvjete korištenja, odabrati instalacijski direktorij, moguće je instalirati upravljačke programe koji unaprjeđuju funkcionalnosti tipkovnice i ostale uobičajene parametre koji se pojavljuju kod većine instalacija. Nakon uspješne instalacije potrebno je ponovno pokrenuti računalo.



Slika 7 - VMware Workstation Player

Za nastavak pripreme virtualnih mašina potrebno je skinuti jednu od Linux distribucija. U ovom primjeru koristiti će se Centos distribucija. Na https://www.centos.org/download/ nalazi se iso datoteku koja sadrži instalaciju Centos distribucije.

Nakon preuzimanja iso datoteke instalacija se izvršava kreiranjem virtualne mašine (Create a New Virtual Machine) i potom se izabire preuzeta iso datoteka. Sljedeći korak je postavljanje podataka o korisničkom računu na operacijskom sustavu, što je integrirano u proces izrade virtalne mašine prije instalacije operacijskog sustava. Predzadnji korak je odabir veličine virtualnog diska koji će biti raspoloživ virtualnoj mašini za pohranu operacijskog sustava i podataka. U zadnjem koraku moguće je odraditi dodatnu konfiguraciju parametara virtualne mašine kao što je povećanje RAM memorije ili promjena mrežne konfiguracije.

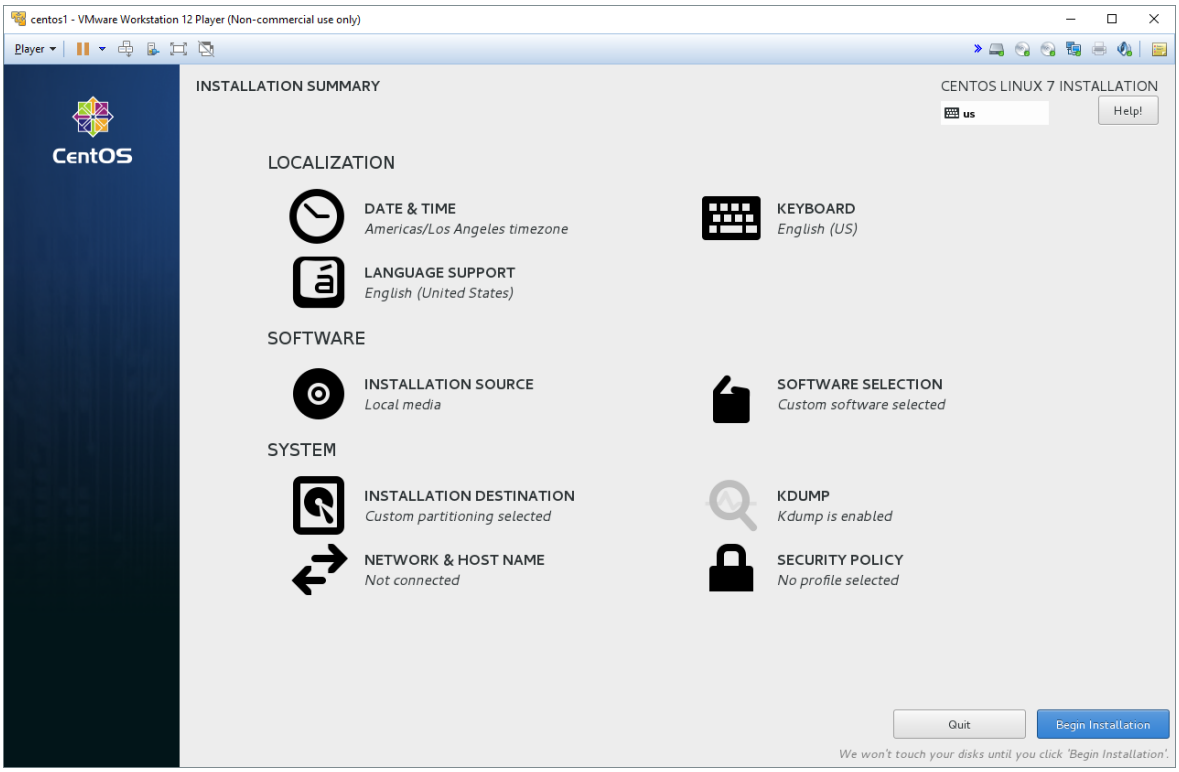


Slika 8 - Postavke virtualne mašine

Nakon završetka definiranja parametara virtualne mašine ona se automatski pokreće s iso instalacijskog medija.

7.1.3.3 Instalacija Centos Linux distribucije

Linux je operacijski sustav koji je nastao 90-ih godina prošloga stoljeća. Linus Torvalds, programer koji je napisao Linux, napisao je taj operacijski sustav kako bi omogućio ostatku svijeta njegovo besplatno korištenje. Linux je kasnije razvijan tako da programeri iz čitavog svijeta doprinose njegovom razvoju. Svoju primjenu našao je u podrumima enzuzijasta koji su ga razvijali, vrlo brzo je preuzeo ulogu posluživanja brojnih web stranica. Razvojem grafičkog sučelja postao je jednostavniji za korištenje i time konkurencija Windows operacijskim sustavima. Android je temeljen na Linux kernelu što ga čini vodećim po broju koji okupira pametne telefone. U zadnje vrijeme Linux nalazi veliku primjenu od uređaja kao što je Raspberry Pi sve do pametnih automobila. Za očekivati je još veći porast primjene Linux operacijskog sustava na IoT tržištu.



Slika 9 - Instalacija Centos Linux distribucije

S ovog ekrana potrebno je konfigurirati mrežne postavke i particioniranje diskova kako bi proces instalacije započeo.

Kako bi mrežni adapter dobio adresu od lokalnog DHCP servera potrebno ga je upaliti unutar NETWORK & HOST NAME izbornika.

Radi jednostavnosti instalacije particioniranje diska može biti zadano kao automatsko.

Postupak instalacije je isti za izradu druge virtualne mašine.

7.1.3.4 Konfiguracija vatrozida i auzorizacija jednim paketom

Implementacija autorizacije jednim paketom koja će se koristiti zove se knockknock. Knockknock je aplikacija napisana u Python programskom jeziku za starije Linux distribucije stoga će biti potrebno napraviti neke od izmjena kako bi ona radila na Centos 7 distribuciji.

Git je sustav za upravljanjem izvornim kodom i potrebno ga je instalirati kako bi se pomoću njega preuzeo knockknock. Centos koristi yum kao paketni sustav pomoću kojega će se instalirati Git. Potrebno je imati root ovlasti ili biti korisnik koji je dio wheel grupe kako bi se mogla obavljati instalacija programa na Linux sustavu.

Kao root korisnik tj. korisnik s najvećim pravima na sustavu dovoljno je upisati naredbu

yum -y install git

i git će biti instaliran. Svi ostali korisnici koji su dio wheel grupe evaluaciju prava dobivaju korištenjem sudo naredbe stoga će naredba koju trebaju upisati u terminal biti

sudo yum -y install git.

Nakon što je Git instaliran pomoću njega je moguće preuzeti izvorni kod korištenjem naredbe

git clone https://github.com/moxie0/knockknock.git

čime će se pojaviti nova mapa imena knockknock unutar trenutnog direktorija iz kojeg je naredba izvršena.

Postupak instalacije gita i preuzimanja knockknock programa obavljen je na obje virtualne mašine. Virtualna mašina centos1 će biti ta koju će štiti vatrozid i kojoj će centos2 virtualna mašina pristupati preko mreže tako da se autorizira slanjem jednog paketa.

Centos 7 distribucija koristi noviji program za upravljanje vatrozidom koji će biti onemogućen pomoću naredbi

systemctl stop firewalld i systemctl disable firewalld

budući da komplicira konfiguraciju vatrozida. Stariji servis koji se još dan danas koristi biti će instaliran pomoću naredbe

yum -y install iptables-services

i pokrenut pomoću naredbe

systemctl start iptables.service i systemctl enable iptables.service.

Unutar direktorija knockknock nalazi se skripta minimal-firewall.sh. Prije izvršavanje skripte potrebno je izbrisati trenutnu konfiguraciju pomoću naredbe

iptables -F

Skripta minimal-firewall.sh ne može se pokretati tj. potrebno je promjeniti prava kako bi se ona mogla izvršavati. Naredba

chmod +x minimal-firewall.sh

će omogućiti njezino izvršavanje svim korisnicima.

Kako bi knockknock program radio potrebno je pokrenuti skriptu minimal-firewall.sh koja će konfigurirati vatrozid tako da sve izlazne konekcije dopušta (npr. moguće je vršiti preuzimanje ažuriranja), a sve ulazne odbija i radi logiranje tj. popis pokušaja ostvarivanja mrežnog prometa. Program knockknock kasnije čita log datoteku i na temelju zaglavlja IP paketa koje je kriptirano dešifrira i određuje da li uređaj koji je poslao poseban paket dobiva pristup nekoj aplikaciji koja sluša na nekom portu kojeg vatrozid štiti.

Centos 7 i ostale novije distribucije koriste systemd inicijalizacijski sustav i time njegov sustav logiranja s kojom knockknock program ne zna raditi. Kako bi knockknock program radio na svim novijim distribucijama neophodno je izvršiti izmjenu konfiguracije rsyslog servisa. Koristeći jedan od uređivača teksta potrebno je otvoriti /etc/rsyslog.conf datoteku i obrisati komentar kojeg čini znak # ispred riječi kern i zadati putanju do datoteke /var/log/kern.log te spremiti izmjene. Nakon ovog koraka restart rsyslog servisa je neophodan kako bi se izmjene primjenile. Centos 7 ne dolazi s instaliranim paketnim sustavom za python koji služi za instalaciju ostalih modula. Jedan od modula koji knockknock koristi je pycrypto i da bi se mogao instalirati potrebne su dodatne datoteke koje sadrži paket python-devel. Iduća naredba instalirati će sve navedeno

easy_install pip && yum -y install python-devel && pip install pycrypto

Knockknock zahtjeva hping3 alat kako bi mogao slati modificirane parametre unutar IP zaglavlja stoga ga je potrebno instalirati naredbom:

yum -y install epel-release && yum -y install hping3

Datoteka setup.py u knockknock mapi služi za instalaciju i distribuciju modula. Naredba

python setup.py install

mora biti izvršena prije daljnjeg korištenja knockknock programa.

Knockknock se sastoji od više programa:

- knockknock-daemon.py služi za izmjenu postavki vatrozida tako što se konstantno izvršava u pozadini i traži pakete s određenim kriptografskim potpisom unutar IP zaglavlja
- knockknock-genprofile.py služi za generiranje profila koje se izvršava na uređaju kojeg knockknock-daemon štiti i spremaju u /etc/knockknock.d/profiles/<imeProfila>/, a datoteke koje definiraju profil trebaju biti kopirane na uređaj koji pokušava ostvariti komunikaciju u ~/.knockknock/<IP_adresa_uređaja_kojem_pristupa ili DNS_ime >/
- knockknock.py koristi se na uređaju koji pokušava pristupiti tj. na klijentu koristeći informacije o profilu kako bi poslao paket kojeg će knockknock-deamon.py obraditi i dodati pravilo u konfiguraciju vatrozida

Kao što je već pojašnjeno knockknock-genprofile.py služi za generiranje profila. Program se izvršava iz ljuske operacijskog sustava prosljeđujući dva parametra. Prvi parametar je ime profila, drugi parametar je broj porta koji će se koristiti za identifikaciju profila i ne smije ga koristiti neki servis ili aplikacija.

Izvršavanje naredbe

knockknock-genprofile.py centos2 1943

generirati će profil kojem će nakon slanja paketa biti dozvoljeno ostvarivanje SSH (*engl.* Secure Shell) konekcije.

Profil se sastoji od 4 datoteke:

- cipher.key - sadrži ključ koji se koristi za enkripciju
- config - sadrži port koji se koristi kao parametar prilikom identifikacije profila
- counter - sadrži broj poslanih paketa prilikom pokužaja autentikacije
- mac.key - sadrži Message Authentication Code koji osigurava autentičnost paketa

Generirane datoteke trebaju biti kopirane u odgovarajuću mapu kako je navedeno. SCP koristi SSH protokol za kopiranje datoteka između uređaja koji imaju pokrenut SSH servis i moguće ga je iskoristiti za kopiranje profila.

scp /etc/knockknock.d/profiles/centos2/* root@192.168.1.6:/root/.knockknock/192.168.1.5/

Za pokretanje knockknock-daemon programa potrebno je dati prava čitanja svim ostalim korisnicima tj. korisnicima koji nisu niti vlasnik niti dio grupe vlasnika kern.log datoteke zato što knockknock daemon koristi sigurnosni mehanizam otpuštanja prava, tako da čak ako je program i pokrenut s pravima root korisnika, nakon otpuštanja prava tijekom svog izvršavanja program više nema pristup kern.log datoteci i ne može čitati iz nje.

Naredba

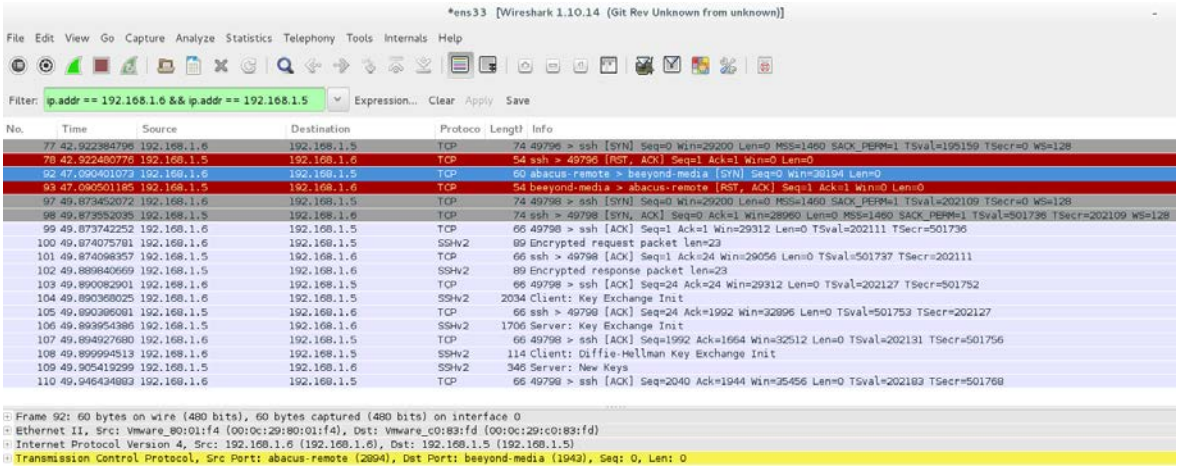
chmod 604 /var/log/kern.log

omogućiti će normalan rad programa.

Program knockknock prima 2 parametra. Prvi parametar je broj porta za koji se zahtjeva da pristup bude odobren kroz vatrozid, a drugi je IP adresa uređaja kojem se pokušava pristupiti po tom portu.

8. Opažanja i analiza rezultata

Uz pomoć alata Wireshark koji služi za analizu mrežnog prometa, biti će prikazan pokušaj spajanja s centos2 na centos1 virtualnu mašinu. Alat je korišten tako da prikazuje samo pakete izmjenjene između virtualnih mašina centos1 i centos2 koristeći filter polje koje koristi njihove IP adrese.



Slika 10 – Prikaz mrežnog prometa pomoću alata Wireshark

Iz paketa broja 77 vidljivo je da centos2 virtualna mašina IP adrese 192.168.1.6 pokušava ostvariti SSH konekciju s centos1 virtualnom mašinom IP adrese 192.168.1.5. Konekcija nije ostvarena budući da su pravila postavljena minimal-firewall.sh skriptom na način da centos1 virtualna mašina odbacije sav dolazni promet i šalje obavjest tako što postavlja TCP-reset parametar u zaglavlje. U 78. paketu centos1 je odbio konekciju.

92. paket je poslan izvršavanjem knockknock.py programa. Paket je koristio port 1943 koji je asociran uz profil koji centos2 virtualna mašina koristi nakon čega je centos1 odbio paket. Iako je paket odbijen vatrozidom, njega je obradio glavni knockknock-daemon.py čitajući kern.log datoteku i zatim dodao novo pravilo u konfiguraciju vatrozida koje traje 15 sekundi i omogućava samo centos2 virtualnoj mašini spajanje na port 22 koji koristi SSH protokol.

Trajanje pravila koje se dodaje u konfiguraciju vatrozida može biti promjenjeno na proizvoljno vrijeme. Unutar datoteke /etc/knockknock.d/config nalazi se delay parametar kojim se određuje trajanje pravila koje autorizira konekciju.

Ostali paketi predstavljaju uspješno spajanje centos2 na centos1 virtualnu mašinu.

9. Preporuke za dizajn sigurnog okruženja Internet stvari

Samom dizajnu treba pristupiti minimalistički tj. treba osigurati korištenje isključivo onih komponenti i protokola koji su nužni kako bi neko rješenje moglo obavljati svoju primarnu funkciju. Fizički pristup napadaču ne bi trebao omogućiti otkrivanje nikakvih informacija pomoću dostupnih sučelja.

Dobar mehanizam automatskog preuzimanja ažuriranja jedna je od ključnih preporuka koja bi se trebala provesti iz više razloga. Sve ranjivosti koje u budućnosti budu otkrivene na ovaj način mogu biti mogu biti uklonjene bez potrebe da korisnik sam vrši ažuriranja jer obično korisnik nije niti svjestan da je ranjivost prisutna, a samim time je ažuriranje obavljeno čim su sigurnosne zakrpe izdane. Prilikom ažiriranja je bitno koristiti kriptografiju u svrhu osiguranja mrežne veze i digitalnog potpisivanja samog ažuriranja jer su se u prošlosti pojavili napadi koji su iskorištavali mehanizam ažuriranja kako bi napadač dostavio lažne nadogradnje koje su mu kasnije ostvarile neautorizirani pristup.

S razine mrežne infrastrukture unutar koje se uređaj nalazi moguće je odraditi većinu sigurnosnih kontrola koje povećavaju sigurnost koje ne ovise o samom proizvođaču uređaja. Korištenje sigurnosnih ekstenzija za postojeće protokole koje garantiraju povjerljivost, integritet i autentičnost podataka. Nadzor uređaja s razine mrežne infrastrukture može indicirati izvršava li se napad s tog uređaja, koristi li vrstu prometa koju takav uređaj uobičajeno ne koristi i slično.

Većina tehnologija koja stoji iza Internet stvari bazirana je na web aplikacijama koje se izvršavaju unutar web preglednika kao što je HTML i Javascript kako bi se generirale web stranice za pristup i upravljanje uređajima kao što su IP kamere. OWASP (engl. *Open Web Application Security Project*) je projekt koji je nastao kao rezultat rada sigurnosnih stručnjaka kako bi otkrili postojeće ranjivosti pružili smjernice u osiguravanju web aplikacija.[13]

[13] https://www.owasp.org/index.php/Main_Page, 01.09.2017.

Zaključak

Internet stvari još uvijek je u samom procesu nastajanja i njegova široka primjena očekuje se tek u godinama koje slijede. S vremenom koje prolazi vidljivo je da je Internet stvari još uvijek nezrelo područje po pitanju sigurnosti. Ovo je većinom krivica samih proizvođača koji ne razmišljaju o posljedicama koje mogu proizaći prilikom kompromitiranja sustava koji imaju dodir sa stvarnim svijetom koji nas okružuje. Bitno je napomenuti kako treba osigurati smjernice koje će poslužiti svim sudionicima koji grade Internet stvari i povećati razinu svijesti kako bi se spriječili incidenti u budućnosti. Ovo predstavlja veliki izazov jer je potrebna velika količina znanja te nove tehnologije koje se uvode i njezine primjene još su uvijek neistražene pa se sigurnosni propusti obično otkriju nakon nekog vremena.

Veliki problem koji se pojavljuje uključuje sva dosada razvijena rješenja koja nemaju sigurnosne značajke omogućene i pitanje je vremena kada će se otkriti njihove ranjivosti, kako će se manifestirati te hoće li ugroziti ljudske živote. Uvođenjem regulacija koje zahtjevaju povećanu razinu sigurnosti moguće je mitigirati dosada poznate vrste napada kojih je mnogo. Međutim, provjeru sigurnosti Internet stvari treba provoditi redovito jer su nove načini napada otkriveni svakim novim danom.

Praktični dio ovog rada koji opisuje primjenu SSH tuneliranja i autorizacije jednim paketom primjenjiv je na veliki dio Internet stvari budući da je Linux operacijski sustav jedan od najčešćih odabira. Prednost je ta što ne zahtjeva izmjenu u postojećim aplikacijama i primjenjiv je na sve aplikacije koje koriste Internet protokol za komunikaciju.

Popis kratica

IT	Information Technology	informacijske tehnologije
LAN	Local Area Network	lokalna mreža
IP	Internet Protocol	Internet protokol
RFID	Radio Frequency Identification	identifikacija radio frekvencije
IoT	Internet of Things	Internet stvari
NTP	Network Time Protocol	protkol mrežnog vremena
DNS	Domain Name System	domenski sustav imena
TCP	Transmission Control Protocol	protokol kontrole prijenosa
UDP	User Datagram Protocol	protokol korisničkih podataka
WSN	Wireless Sensor Network	mreža bežičnih senzora
DDOS	Distributed Denial of Service	distribuirano uskraćivanje usluge
MAC	Message Authentication Code	kod za autentikaciju poruke
SSH	Secure Shell	sigurna ljuska
RAM	Random Access Memory	memorija nasumičnog pristupa

Popis slika

Slika 1 - Prikaz TCP/IP modela 5

Slika 2 – Prikaz IoT mrežne arhitekture 6

Slika 3 - Primjer pravila iptables vatrozida za Linux operacijskom sustavu 16

Slika 4 - Prikaz MITM napada 17

Slika 5 – Tuneliranje koristeći SSH protokol 19

Slika 6 - Virtualna okolina 20

Slika 7 - VMware Workstation Player 21

Slika 8 - Postavke virtualne mašine 22

Slika 9 - Instalacija Centos Linux distribucije 23

Slika 10 – Prikaz mrežnog prometa pomoću alata Wireshark 27

Literatura

Svaki autor piše popis literature na kraju rada. Popis literature se piše stilom literatura.

[1] https://en.wikipedia.org/wiki/Internet_of_things#History, 20.05.2017.

[2] http://talkincloud.com/cloud-computing/iot-past-and-present-history-iot-and-where-its-headed-today, 20.05.2017

[3] https://en.wikipedia.org/wiki/Kevin_Ashton, 20.05.2017.

[4] DACOSTA, F. Rethinking the Internet of Things, Apress, 2014

[5] https://hr.wikipedia.org/wiki/Podatak,_informacija,_znanje,_mudrost, 27.08.2017.

[6] http://www.iotevolutionworld.com/m2m/articles/376816-history-smart-homes.htm, 23.05.2017.

[7] http://www.ioti.com/smart-cities/world-s-5-smartest-cities, 23.05.2017.

[8] https://en.wikipedia.org/wiki/Mirai_(malware), 12.06.2017.

[9] http://blog.whitescope.io/2017/05/understanding-pacemaker-systems.html, 15.08.2017.

[10] https://blog.lookout.com/hacking-a-tesla

[11] https://www.owasp.org/index.php/Main_Page, 01.09.2017.