Sem vložte zadání Vaší práce.

FAKULTA
INFORMAČNÍCH
TECHNOLOGIÍ
ČVUT V PRAZE

Diplomová práce

Detekce anomálií v provozu IoT sítí

Bc. Dominik Soukup

Katedra počítačových systémů
Vedoucí práce: Tomáš Čejka

7. března 2018

Poděkování

Doplňte, máte-li komu a za co děkovat. V opačném případě úplně odstraňte tento příkaz.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou, a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla, a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu), licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 7. března 2018 .....

Odkaz na tuto práci

Soukup, Dominik. *Detekce anomálií v provozu IoT sítí.* Diplomová práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2018.

Abstrakt

V několika větách shrňte obsah a přínos této práce v češtině. Po přečtení abstraktu by se čtenář měl mít čtenář dost informací pro rozhodnutí, zda chce Vaši práci číst.

Klíčová slova Nahraďte seznamem klíčových slov v češtině oddělených čárkou.

Abstract

Sem doplňte ekvivalent abstraktu Vaší práce v angličtině.

Keywords Nahraďte seznamem klíčových slov v angličtině oddělených čárkou.

Obsah

Seznam obrázků

Úvod

Koncept internetu existuje již několik desítek let a pro spoustu lidí se stal nedílnou součástí pracovního i osobního života. V poslední době je možné sledovat stále rostoucí počet zařízení, která jsou do něj zapojena. Tento trend by měl pokračovat i do budoucnosti, a dokonce v ještě větším měřítku. Odhadem je přes 30 miliard připojených zařízení do roku 2020 [1]. Důvodem zrychleného růstu je expanze síťového připojení na veškeré elektronické zařízení a senzory, které umožní vzdálené ovládání a monitorování. Pro označení tohoto trendu se používá termín Internet věcí (Internet of Things, IoT).

Cílem IoT je usnadnit, zlepšit a ušetřit lidskou činnost napříč všemi odvětvími. Uplatnění se nachází zejména ve výrobních podnicích, dopravě nebo běžných domácnostech. Příklad konkrétního nasazení popisuje článek [2], jehož cílem je měření kvality spánku a odhalení případných poruch. Monitorovací systém lze dále propojit například s ovládáním místnosti, které bude reagovat na aktuální úroveň spánku úpravou světel, oken nebo vzduchu.

Dále je upraven model komunikace, který již nevyžaduje zasílání zpráv centrálnímu serveru (north-south), ale podporuje přímou komunikaci mezi připojenými uzly (east-west). Oblast IoT nezahrnuje jen malá a nevýkonná zařízení, ale jeho součástí jsou i výkonná datová centra a pokročilé algoritmy, které vyhodnocují získané informace.

Hlavní hrozbou Internetu věcí je bezpečnost. Dochází zde k přenosu citlivých dat, která slouží k automatizovanému řízení dalších systémů, monitorování prostředí a zabezpečovacím účelům. Zároveň s masivním rozšířením nově připojených zařízení roste riziko vzniku nových útoků a možnosti způsobení větších škod. Příkladem bezpečnostního incidentu je útok na distribuční síť elektrického produ na Ukrajině, který měl dopad na 225 000 zákazníků [3].

Pro potlačení možného vzniku hrozeb musí být součástí každné dnešní IoT sítě sada procesů, které umožní důvěryhodné získávání validních dat, vzdálenou správu a možnost detekce anomálií jako v běžných IP (Internet Protocol) sítích.

Kapitola 1

Cíl práce

Cílem magisterské práce je analyzovat množinu aktuálně používaných protokolů pro komunikaci v IoT sítích a identifikovat jejich bezpečnostní zranitelnosti. Při analýze bude věnována pozornost zejména bezdrátovým senzorovým protokolům. Na základě získaných znalostí bude navržen, implementován a otestován algoritmus pro monitorování a automatickou detekci anomálního provozu v IoT sítích. Algoritmus bude možné spustit v prostředí nově vznikající opensource brány BeeeOn, čímž dojde k rozšíření dostupných bezpečnostních funkcí.

Kapitola 2

Analýza

Kapitola se zabývá analýzou celkové architektury IoT sítí a způsoby pro jejího zabezpečení. Postupně je prozkoumán komunikační model, možné bezpečnostní hrozby, a existující řešení pro obranu. Na základě analýzy jsou uvedeny funkční a nefunkční požadavky, které jsou kladeny na výsledný program. V závěru jsou vybrány konkrétní technologie pro realizaci.

2.1 Architektura IoT sítí

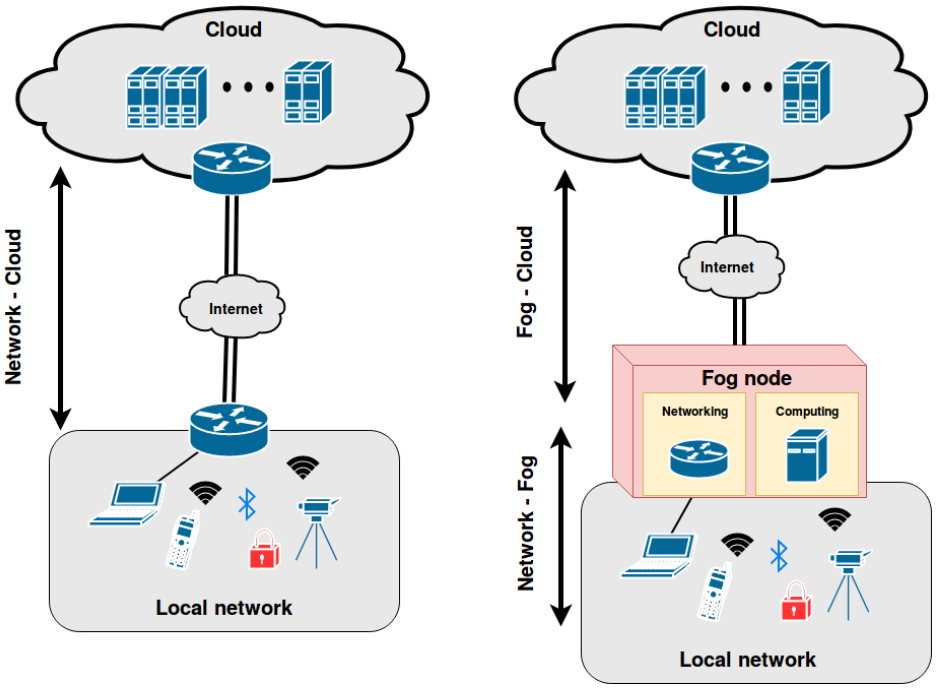
V blízké době se očekává stále větší nárůst zařízení, která jsou připojena k internetu. Dle odhadů by jejich počet měl v roce 2020 překročit 30 miliard [1]. Pro takové množství připojení už není možné, aby každé zařízení komunikovalo přímo se vzdáleným datovým centrem, protože nároky na potřebnou šířku pásma by byly obrovské [4]. Dalším problémem je často velmi omezený výkon připojených prvků, který je nezbytný pro použití bezpečnostních funkcí umožňujících kompletně zabezpezpečenou komunikaci.

Řešení těchto problémů je do probíhající komunikace přidat několik podvrstev, které umožní přesunout výpočetní výkon blíže ke koncovým zařízením, a tím celý proces zpracování dat provést efektivněji.

2.1.1 Fog computing

Fog computing je rozšíření Cloud computingu, které spočívá v přesunutí výpočetního výkonu blíže k okraji sítě. Rozšíření je umožněno pomocí přidání síťových zařízení, které kromě běžných funkcionalit poskytují i výpočetní výkon pro běh externích programů. Programy je často možné nasadit pomocí kontejnerů nebo samostatných virtuálních strojů, což velmi usnadňuje jejich distribuci [4].

Porovnání klasické a Fog architektury se nachází na obrázku 2.1. V reálném nasazení může být použito i více Fog vrstev, kde každá provádí určitý stupeň předzpracování a řízení dat. Zavedením principů Fog computingu vznikají pro



Obrázek 2.1: Porovnání klasické a Fog architektury

síť následující výhody:

- **Zlepšení bezpečnosti** [4]

  Síťové prvky jsou trvale napájené a připojené k internetu. Podporují pokročilé bezpečnostní funkce, a proto je možné například vytvářet šifrované tunelové spojení pro bezpečný přenos dat.

- **Nižší nároky na šířku pásma a latency** [4]

  Odeslaná data z koncových zařízení jsou zpracovávána a filtrována na okraji sítě. Tím je možné rychleji reagovat na přijaté zprávy a snížit nároky na latency a šířku pásma. Zároveň krátkodobá data mohou být uložené ve Fog vrstvě a centrální datové cetrum může být využito pro dlouhodobé údaje, které se zpracovávají pokročilými algoritmy pro analýzu dat.

- **Jednotná správa** [4]

  Při správě sítě už se nemusí přistupovat přímo na koncové prvky, které často komunikují různými protokoly, ale stačí pouze řídit síťová zařízení v jednotlivých Fog vrstvách, které odstiňují různorodost protokolů

a nabízí standardizovaný přístup. Díky této abstrakci je zároveň zjednodušeno zpracování získaných dat a je umožněno přímé zasílání zpráv mezi koncovými prvky, které používají odlišné komunikační protokoly.

2.1.2 IoT brána

IoT brána je síťové zařízení, které je umístěno velmi blízko koncových zařízení a představuje vstup do Fog vrstvy. Jejím hlavním cílem je získávat data z připojených zařízení a poskytovat je vyšším vrstvám. Pokud je brána reprezentována výkonnějším síťovým prvkem, tak v rámci brány může probíhat i základní zpracování dat.

Pro IoT sítě je typické, že obsahují velké množtví koncových prvků komunikujících různorodými způsoby. Zejména senzory používají protokoly, které nepodporují IP spojení. Důvodem použití této komunikace je často velký důraz na nízkou spotřebu a specifické požadavky na způsob zasílání zpráv. Příkladem protokolů pro senzorové sítě je například: Z-Wave, Bluetooth a Zigbee. Jejich detailní popis se nachází v kapitole 2.2. Tato různorodost vyžaduje, aby brána obsahovala dodatečná rozhranní, které umožní připojení nejrůznějších bezdrátových i drátových koncových prvků.

V současné době existuje mnoho různých bran jejichž parametry se liší dle způsobu nasazení a provozních nároků. Velkým problémem v této oblasti je malý důraz na bezpečnostní funkce, které umožní vzdálené řízení brány, kontrolu provozu a aktualizace programového vybavení. Z těchto důvodů vznikl opensource projekt BeeeOn [5] jehož cílem je vytvořit softwarou IoT bránu, kterou bude možné spustit na různých hardwarových platformách. BeeeOn brána je navržena modulárně tak, aby byla schopna zpracovávat více rozdílných senzorových protokolů, a tím je možné provozovat jedno univerzální zařízení namísto několika proprietárních. Zároveň je kladen důraz na bezpečnost a veškeré údaje, které je možné získat o provozu, jsou poskytovány pro analýzu. Nad těmito údaji je postaven detekční algoritmus, který je výsledkem této diplomové práce.

2.1.3 Komunikační model a jeho hrozby

Při použití principů popsaných v předchzích kapitolách lze model komunikace rozdělit do následujících vrstev [6]:

- Aplikační vrstva
- Síťová vrstva
- Senzorová vrstva

Jednotlivé vrstvy budou popsány v následujích podkapitolách.

2.1.3.1 Senzorová vrstva

Senzorová vrstva obsahuje veškerá koncová zařízení, které získávají informace ze svého okolí nebo vykonávají potřebnou službu [7]. Tato zařízení jsou připojena kabelově nebo bezdrátově k IoT bráně. K jedné bráně může být připojeno několik prvků, které komunikují odlišnými způsoby.

Velkým nebezpečím této vrstvy jsou zejména bezdrátové protokoly, protože při nepoužití zabezpečení může snadno dojít k odposlouchávání nebo úprávám provozu [6]. Dále se zde mohou vyskytovat zařízení, které jsou označeny jako zabezpečené, ale používají zastaralé bezpečnostní funkce nebo obsahují implementační chyby. Tento případ je velmi nebezpeční, protože vyvolává falešný pocit bezpečí.

2.1.3.2 Síťová vrstva

Po zpracování senzorových dat na bráně je nutné získané informace odeslat dalším službám. K tomuto účelu slouží síťová vrstva. Cílem této vrstvy je také umožnění vzdálené správy brány [7]. Pro výběr konkrétního protokolu je nutné znát rozhranní aplikační vrstvy. Nicméně spojení je většinou vytvořeno pomocí protokolu HTTPS (Hypertext Transfer Protocol Secure) nebo technologie VPN (Virtual Private Network). Nad tímto spojením je postaven další služba pro výměnu zpráv. Příkladem může být: MQTT (Message Queuing Telemetry Transport), COAP (Constrained Application Protocol), AMQP (Advanced Message Queuing Protocol).

Bezpečnostní hrozby této vrstvy jsou stejné jako v klasických sítích. Je potřeba dodržet principy důvěry, integrity a dostupnosti. Tímto přístupem je možné předejít útokům jako: DDoS (Distributed Denial Of Service), MITM (Man In The Middle) a podvržení informací. Zároveň je nutné nezapomenout, že se zde většinou vyskytuje M2M (Machine To Machine) komunikace a je důležité použít vhodná komunikační rozhranní [6].

2.1.3.3 Aplikační vrstva

Aplikační vrstva se stará o ukládání dlouhodobých dat a jejich finální zpracování. Zároveň zobrazuje uživateli zpracované informace a umožňuje provádět konfiguraci celé sítě. Z důvodu možné rozsáhlosti celé sítě je kritické, aby správa topologie podporovala automatizaci. [7].

Tato vrstva je umístěna vetšinou v datovém centru a umožňuje vzdálený přístup. Její bezpečnostní problémy lze přirovnat k problémům cloud computingu. Dle množtví požadovaných funkcí může být různě složitá a s rostoucí složitostí se také liší nároky na úroveň zabezpečení. Příkladem možných útoků může být: Buffer Overflow, SQL Injection nebo DDoS.

2.2 Používané komunikační protokoly

2.2.1 MQTT

MQTT je otevřený komunikační protokol typu pulish/subcribe, který byl navržen v roce 1999. Již od návrhu byl zaměřen na nízkou náročnost komunikace a jednoduchost implementace. Díky těmto vlastnostem je velmi vhodný pro IoT a M2M systémy.

2.2.1.1 Způsob komunikace

Protokol MQTT je postaven nad transportní vrstvou TCP (Transmission Control Protocol) a využívá model pulish/subcribe, který vychází z tradičního způsobu zasílání zpráv typu klient-server. Roli serveru zde plní speciální uzel, který se nazývá broker. Broker je známý všem ostatním klientům, kteří mohou zasílat zprávy pomocí operace publish nebo se přihlásit o příjem zpráv díky operaci subscribe. Na základě provedených operací broker přijímá zprávy a rozhoduje o jejich přeposlání. Způsob odeslání zprávy závisí na obsažených metadatech.

Nejčastěji o směru odesláni rozhoduje předmět (topic) zprávy. Předmět je tvořen jednoduchým UTF-8 řetězcem s hierarchickou strukturou, ve které jsou jednolivé vrstvy odděleny dopředným lomítkem (např. /domov/přízemí/světloŠatna). V předmětu zprávy mohou být některé vrstvy nahrazeny zástupnými symboly + a #. Symbol + dokáže nahradit pouze jednu úroveň předmětu libovolným řetězcem a symbol # umožňuje zastoupit více úrovní. Díky těmto symbolům mohou klienti odeslat nebo přijímat zprávy z více témat.

Další užitečnou položkou protokolu MQTT je QoS (Quality of Service), která může nabývat hodnot 0, 1 nebo 2.

- **QoS 0**

  Veškeré zprávy jsou odesílány bez potvrzení a žádným způsobem není zvýšena úroveň spolehlivosti, která je shodná se spolehlivostí protokolu TCP.

- **QoS 1**

  Pomocí potvrzování zajišťuje, že každá zpráva bude příjemci doručena alespoň jednou.

- **QoS 2**

  Umožňuje, aby každá zpráva byla spolehlivě doručena právě jednou.

Hodnota QoS se nastavuje vždy mezi dvěma uzly při navazování spojení. Z pohledu brokeru se může stát, že přijatá a odeslaná zpráva mají jiné QoS. Úrovně 1 a 2 dále umožňují perzistetní ukládání zpráv v případě, že příjemce je nedostupný. Zároveň platí, že s vyšší úrovní roste i režie komunikace.

2.2.1.2 Bezpečnost

Zabezpečení protokolu MQTT je možné rozdělit do následujících vrstev:

- **Síťová vrstva**

  Veškerá komunikace postavena nad TCP/IP. Z tohoto důvodu je možné komunikaci zapouzdřit pomocí VPN (Virtual Private Network) připojení jako v běžných počítačových sítích. Z důvodu větších nároků na výkon je toto řešení vhodnější pro výkonější zařízení jako jsou například IoT brány, které mohou s brokerem navázat site-to-site spojení.

- **Transportní vrstva**

  Na této úrovni se využívá šifrování provozu pomocí protokolu TLS (Transport Layer Security). Omezením této metody jsou požadavky na výkon, které mohou být poměrně vysoké pokud nastává časté navazování spojení.

- **Aplikační vrstva**

  Samotný protokol MQTT nedefinuje žádné šifrovací mechanismy na aplikační úrovni. Zabezpeční dat zde musí zajistit uživatel ještě před zapozdřením do MQTT zprávy. Ovšem tímto způsobem je možné šifrovat jen tělo zprávy a hlavička zůstává nezměněná.

  Pro autentizaci je možné využít ověření pomocí jména a hesla nebo x.509 certifikátu. Jméno a heslo je přenášeno nešifrovaně, a proto je vhodné tuto metodu doplnit se zabezpečením síťové nebo transportní vrstvy. Autentizaci pomocí certifikátů je možné využít v případě použití TLS. Tato metoda je vhodnější pokud všechna zařízení jsou po jednotnou správou a je možné automatizovat distribuci klientských certifikátů.

  Dále je na straně brokeru možné definovat pravidla pro autorizaci. Tyto pravidla přiřazují klientů oprávnění pro provedení operací publish a subscribe nad příslušnými tématy.

2.2.2 COAP

2.2.3 Z-Wave

2.2.4 Bluetooth

2.3 Bezpečnostní slabiny a možnosti detekce

- Vnější vs vnitřní útoky
- Signature based vs Anomaly based
- paketová vs flow vs extended flow analýza

- Senzorová data - provozní data z driverů vs dedikované zařízení (SDR, ...)

2.4 Existující řešení

- Existuje řešení pomocí Suricata, které používá Turris (centrální, signature based řešení)
- Neexistuje řešení, které by provádělo detekce z senzorových dat
- Neexistuje řešení, které by umožňovalo senzorové nebo pokročilé síťové detekce na bráně nebo mimo ni

2.5 Analýza požadavků

2.6 Zvolené řešení

Kapitola 3

Návrh

3.1 Možnosti detekce

- testbed, externí sonda, provozní data

3.2 Možnosti nasazení

- popis celkové architektury řešení
- lokální a externí detekce

3.3 Scénáře útoků

3.4 Kolektor

3.5 Detektor

3.6 Multiplexor a demultiplexor

Kapitola 4

Realizace

Kapitola 5

Testování

Závěr

Literatura

[1] Statista: Internet of Things (IoT) connected devices installed base worldwide from 2015 to 2025. Statista. [online], 2016 [cit. 2018-01-14]. Dostupné z: `https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/`

[2] Milici, S.; Lázaro, A.; Villarino, R.; aj.: Wireless Wearable Magnetometer-Based Sensor for Sleep Quality Monitoring. *IEEE Sensors Journal*, 2018: s. 2145–2152.

[3] Whitehead, D. E.; Owens, K.; Gammel, D.; aj.: Ukraine cyber-induced power outage: Analysis and practical mitigation strategies. In *2017 70th Annual Conference for Protective Relay Engineers (CPRE)*, 2017, s. 1–8.

[4] Statista: Fog Computing and the Internet of Things: Extend the Cloud to Where the Things Are. Cisco. [online], 2015 [cit. 2018-01-15]. Dostupné z: `https://www.cisco.com/c/dam/en_us/solutions/trends/iot/docs/computing-overview.pdf`

[5] BeeeOn: Main Page. beeeon.org. [online], 2017 [cit. 2018-02-06]. Dostupné z: `https://beeeon.org`

[6] Zhao, K.; Ge, L.: A Survey on the Internet of Things Security. In *2013 Ninth International Conference on Computational Intelligence and Security*, 2013, s. 663–667.

[7] Pacheco, J.; Hariri, S.: IoT Security Framework for Smart Cyber Infrastructures. In *2016 IEEE 1st International Workshops on Foundations and Applications of Self* Systems (FAS*W)*, 2016, s. 242–247.

Příloha **A**

Seznam použitých zkratek

GUI Graphical user interface

XML Extensible markup language

Příloha **B**

Obsah přiloženého CD

- `readme.txt` ..... stručný popis obsahu CD
- `exe` ..... adresář se spustitelnou formou implementace
- `src`
  - `impl` ..... zdrojové kódy implementace
  - `thesis` ..... zdrojová forma práce ve formátu LaTeX
- `text` ..... text práce
  - `thesis.pdf` ..... text práce ve formátu PDF
  - `thesis.ps` ..... text práce ve formátu PS