Makalah Tugas Kuliah Keamanan Perangkat Lunak EL5215

OWASP INTERNET OF THINGS TOP TEN

Oleh:

GALIH GIANTARA

23215034

Dosen:

Dr. Ir. Budi Rahardjo

Magister Teknik Elektro

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung

2016

ABSTRAK

Istilah Internet of Things dapat dijumpai di berbagai macam hal akan tetapi sampai saat ini belum ada definisi standar mengenai istilah Internet of Things. Secara mudah Internet of Things dapat dikatakan dimana benda-benda disekitar kita dapat berkomunikasi satu sama lain dengan memanfaatkan jaringan internet. Ide awal internet of things dikemukakan oleh Kevin Ashton di tahun 1999. Banyak pihak yang mencoba mendalami tentang Internet of Things. Banyak yang memperkirakan bahwa Internet of Things akan menjadi sebuah terobosan di dunia teknologi informasi. Banyak potensi yang dapat digali dari Internet of Things.

OWASP (*The Open Web Application Security Projek*) merupakan organisasi/komunitas terbuka yang fokus di bidang Keamanan Aplikasi. OWASP melakukan penelitian dan mensosialisasikan hasilnya untuk meningkatkan kesadaran akan keamanan aplikasi. OWASP memiliki beberapa projek diantaranya *Web*Goat, *Web*scarab dan OWASP top 10. OWASP Top10 merupakan dokumen yang merangkum 10 celah keamanan pada aplikasi yang paling berbahaya saat ini. Dengan adanya dokumen ini diharapkan pengembang aplikasi dapat memahami 10 celah keamanan ini dan mencegah timbulnya 10 masalah ini pada aplikasinya. OWASP Internet of Things sendiri merupakan rangkuman 10 celah keamanan yang dapat dikategorikan berbahaya menurut OWASP.

Dalam makalah ini dibahas mengenai Internet of Things serta 10 celah keamanan pada Internet of Things berdasarkan OWASP.

Kata Kunci : OWASP, Internet of Things, Keamanan, Aplikasi

DAFTAR ISI

1. Pendahuluan

Dewasa ini, seiring dengan perkembangan jaman, telah dirasakan kemajuan teknologi yang sangat pesat, khususnya dalam bidang teknologi nirkabel. Teknologi nirkabel ini memberikan akses tanpa batas untuk mempermudah keberlangsungan hidup manusia. Internet merupakan salah satu dari bentuk teknologi nirkabel yang masih berkembang. Saat ini hampir semua hal terhubung dengan internet. Berbagai macam hal yang merupakan bagian dalam sebuah sistem dan antara mereka dapat berkomunikasi dengan memanfaatkan koneksi internet, saat ini hal itu dikenal dengan istilah Internet of Things. Internet of Things dapat dijumpai di berbagai macam hal meskipun sampai saat ini belum ada definisi standar mengenai istilah Internet of Things.

Ide awal internet of things dikemukakan oleh Kevin Ashton di tahun 1999. Banyak pihak yang mencoba mendalami tentang Internet of Things. Banyak pihak yang mencoba menggali potensi yang dapat diberikan oleh dari Internet of Things karena menurut pihak-pihak tersebut Internet of Things dapat menjadi terobosan bagi kehidupan manusia di masa depan. Internet of Things bisa direalisasikan dengan tiga paradigma, *internet oriented*, *things oriented* serta *semantic oriented*[5].

Seperti halnya sebuah sistem, Internet of Things tidak lepas dari berbagai ancaman yang dapat merugikan dari para pengguna Internet of Things. Ancaman-ancaman tersebut datang dari para pelaku kejahatan *cyber*. Berkembangnya Internet of Things tidak hanya memberikan keuntungan untuk para penggunanya, tapi serta memperbesar kemungkinan kerugian yang akan dialami oleh pengguna Internet of Things itu sendiri. Kerugian yang mungkin dialami oleh para pengguna Internet of Things dapat berupa bocornya informasi dimana informasi tersebut seharusnya bersifat pribadi.

Untuk itu OWASP, sebuah organisasi non profit menerbitkan OWASP Internet of Things Top 10 demi memeberikan informasi kepada publik mengenai celah-celah keamanan yang dapat dimanfaatkan oleh para pelaku kejahatan *cyber* untuk dimanfaatkan demi keuntungan pribadi. Celah-celah keamanan yang dibahas adalah rangkuman dari berbagai celah keamanan yang dapat terjadi menjadi 10 buah celah keamanan yang menurut OWASP hal tersebut sangat berbahaya. OWASP Internet of Things Top 10 dapat membantu para *user* dan *developer* sistem yang memanfaatkan konsep Internet of Things untuk lebih memahami isu keamanan yang ada pada

Internet of Things, sehingga dapat lebih berhati-hati dalam menggunakan mafaat dari Internet of Things.

2. Internet of Things

Internet of Things merupakan sebuah pengembangan dari teknologi internet dimana semua objek memiliki konektivitas yang memungkinkan objek-objek tersebut untuk mengirim serta menerima data. Terminologi dari Internet of Things merujuk kepada pemikiran untuk masa depan, dimana setiap hari berbagai *Physical object* terkoneksi dengan internet dalam satu bentuk atau lainnya akan tetapi diluar ranah dekstop traditional[1]. Dengan Internet of Things mampu membuat rumah dan segala kehidupan di sekeliling kita menjadi cerdas. Memberi kemampuan inteligensi ke dalam berbagai macam lingkungan seperti halnya rumah, dapat memberikan peningkatan kualitas hidup, seperti dapat membantu kehidupan dari orang-orang yang sudah lanjut usia serta untuk orang yang sedang sakit[2].

Internet of Things melibatkan cakupan yang sangat luas, tidak hanya mengenai perangkat, jaringan atau serta *client*. Ada beberapa elemen yang harus diperhatikan saat bersinggungan dengan Internet of Things. Diantaranya adalah perangkat IoT, cloud, aplikasi *mobile*, *interface* dari jaringan, *software*, penggunaan enkripsi, penggunaan *otentikasi*, keamanan secara fisik dan terakhir *port* USB[3]. Prinsip utama dari Internet of Things adalah Internet of Things merupakan perpanjangan dari internet untuk masuk ke dalam dunia fisik dimana melibatkan interaksi dengan kesatuan fisik pada lingkungan sekitar[4].

3. OWASP (*The Open Web Application Security Projek*)

OWASP (*The Open Web Application Security Projek*) merupakan organisasi/komunitas terbuka yang fokus di bidang Keamanan Aplikasi. OWASP melakukan penelitian dan mensosialisasikan hasilnya untuk meningkatkan kesadaran akan keamanan aplikasi. Komunitas dari OWASP meliputi perusahaan, berbagai macam organisasi khususnya dari bidang pendidikan serta individu dari seluruh dunia. OWASP membuat berbagai macam artikel yang berisikan mengenai metodologi, dokumentasi, peralatan serta teknologi.

Meskipun OWASP memiliki *concern* di bidang keamanan teknologi informasi, OWASP sendiri tidak melakukan afiliasi dengan perusahaan teknologi manapun. Hal ini agar OWASP bebas dari

tekanan dan mampu bersifat objektif dalam memberikan informasi mengenai kemanan yang erat kaitannya dengan dunia teknologi informasi.

3.1 Sejarah

OWASP dimulai pada 9 September 2001 Oleh Mark Curphey dan Dennis Groves dan mulai *online* pada tanggal 1 Desember 2001. Sejak akhir 2003, Jeff Williams telah menjabat sebagai Ketua relawan dari OWASP. Para Pemimpin OWASP bertanggung jawab untuk membuat keputusan tentang arah teknis, prioritas proyek, jadwal, dan melepaskan. Secara kolektif, para pemimpin OWASP dapat dianggap sebagai pengurus Yayasan OWASP. OWASP memiliki beberapa nilai inti yakni:

a. *Open*, OWASP bersifat transparan dalam segala hal, baik dari sisi finansial maupun dari sisi *source code*.
b. *Innovation*, OWASP mendorong serta mendukung inovasi dan eksperimen demi solusi solusi untuk pemecahan tantangan dari segi keamanan perangkat lunak.
c. Global, semua lapisan masyarakat didukung untuk berpartisipasi dalam komunitas OWASP.
d. *Integrity*, OWASP merupakan komunitas global yang bersifat netral, jujur dan dapat dipercaya.

3.2 Publikasi OWASP

Ada beberapa publikasi yang telah dilakukan oleh OWASP, yaitu:

1. OWASP Top Ten: OWASP Top 10 pertama diterbitkan pada tahun 2003 dan diperbaharui secara berkala. Tujuan dari OWASP Top 10 adalah untuk meningkatkan kewaspadaan akan hal-hal yang berkaitan dengan keamanan perangkat lunak dengan melakukan identifikasi risiko keamanan yang paling kritikal. Projek Top 10 ini sendiri mengambil referensi dari berbagai sumber yakni standard-standar yang ada, buku, *tools* serta organisasi-organisasi seperti MITRE, PCI DSS, Defense Information Systems Agency, FTC dan masih banyak lagi
2. OWASP Software Assurance Maturity Model: Projek SAMM berkomitmen untuk membangun *framework* yang dapat digunakan untuk membantu organisasi-organisasi

memformulasikan serta mengimplementasikan sebuah strategi untuk keamanan perangkat lunak yang berdasar kepada kebutuhan yang berkaitan erat dengan resiko yang dihadapi oleh organisasi-organisasi tersebut.

3. OWASP Development Guide: menawarkan bimbingan praktikal yang berisikan contoh dari kode-kode J2EE, ASP.NET, and PHP. OWASP Development Guide meliputi sebuah *extensive array* dari isu-isu keamanan pada level aplikasi dari SQL *injection* maupun hingga permasalahan *modern* seperti *phising*, penanganan kartu kredit serta isu-isu privasi.
4. OWASP *Testing* Guide: meliputi pelatihan terbaik untuk *penetration testing framework* dimana pengguna dapat mengimplementasikannya di organisasi mereka serta bimbingan pengujian sebuah *low-level penetration* yang menjelaskan teknik untuk pengujian untuk sebagian besar aplikasi *web* serta isu-isu keamanan *web*-servis.
5. OWASP Application *Security* Verification Standard (ASVS): sebuah standar untuk menampilkan verifikasi pada keamanan di level menampilkan *application-level Security verifications*.
6. OWASP Top 10 Incident Response Guidance. Projek ini menawarkan sebuah penawaran proaktif terhadap rancangan respon dari sebuah insiden. Sasaran dari dokumen ini adalah dari pebisnis hingga *Security engineers*, *developer*, *auditor*, program, serta penegak hukum.
7. OWASP ZAP Projek: Projek Zed Attack Proxy (ZAP) mudah digunakan dengan *tools* untuk *penetration testing* yang sudah terintegrasi untuk menemukan kerentanan dari aplikasi *web*. Didesain untuk sasaran pengguna yang lebih luas dalam pengalaman akan keamanan termasuk *developer* dan *functional tester* yang baru dalam melakukan *penetration testing*.
8. *Web*goat: Sebuah aplikasi *web* yang sengaja dibuat tidak aman dan dibuat oleh OWASP sebagai bimbingan untuk pelatihan *Security programming*. Sekali dilakukan proses download, aplikasi hadir dengann sebuah tutorial serta sebuah set pelatihan yang berbeda-beda yang menginstruksikan bagaimana mengeksploitasi kerentanan dengan maksud untuk memberi pelatihan membuat kode program yang aman.

4. OWASP Top 10 pada IoT

Salah satu publikasi yang telah dilakukan oleh OWASP adalah OWASP Top 10. OWASP Top 10 adalah sebuah dokumen yang merangkum 10 celah keamanan paling berbahaya pada suatu aplikasi. Tujuan dari adanya publikasi dokumen ini adalah demi meningkatkan kewaspadaan akan keamanan dari suatu perangkat lunak. OWASP Internet of Things sendiri merupakan rangkuman 10 celah keamanan yang dapat dikategorikan berbahaya menurut OWASP.

4.1 *Insecure Web Interface*

Salah satu hal yang dapat menjadi celah keamanan dari sistem Internet of Things yaitu dari segi sisi *web interface*. Ada beberapa hal yang dapat diketahui mengenai celah keamanan dari sisi *web interface*. Pihak-pihak yang dapat menjadi pemberi ancaman pada celah keamanan ini adalah para pengguna yang dapat mengakses *web interface* baik pengguna dalam maupun pengguna luar. Biasanya penyerang memanfaatkan *weak credential*, mengambil *plain-text credential* atau mencacah akun-akun yang ada untuk mengakses *web interface*. Penyerang dapat berasal dari pengguna luar maupun dalam. Kelemahan dari keamanan pada sisi *web interface* dapat muncul saat timbul isu-isu, seperti pencacahan akun, lemahnya *account lockout* atau hadirnya *weak credential*. *Web Interface* yang tidak aman adalah suatu hal yang lazim. Inteface dari *web* muncul hanya untuk jaringan internal, bagaimanapun ancaman dari pengguna internal sama signifikannya dengan ancaman dari pengguna eksternal. Isu-isu pada *web interface* mudah ditemukan saat memeriksa *interface* secara manual beserta dengan *tools* pengujian otomatis untuk mengidentifikasi isu-isu lainnya seperti *cross-site scripting*. Dampak dari *web interface* yang tidak aman adalah data *loss* atau *corruption*, kurangnya akuntabilitas, atau penolakan akses serta dapat membawa kepada pengambil alihan perangkat secara total. Hal ini dapat berdampak pada segi bisnis. Tidak hanya *customer* yang dapat dirugikan, merk dagang dari suatu bisnispun dapat rusak.

Untuk mengetahui apakah *web interface* aman, dapat melakukan beberapa pengecekan. Pengecekan pertama adalah pastikan bahwa *username* dan *password default* dapat diganti selama mempersiapkan produk awal. Yang kedua adalah dengan memastikan bahwa akun akan terkunci jika 3-5 kali terjadi kegagalan saat proses *login*. Kemudian pastikan bahwa akun yang *valid* dapat diidentifikasi dengan menggunakan mekanisme *password recovery* atau halaman *new-user*. Langkah terakhir adalah dengan melakukan *review* terhadap isu-isu seperti *cross-site*

scripting, pemalsuan *cross-site request* serta sql *injection*. Berikut 2 buah contoh dari skenario penyerangan yang dapat terjadi pada sisi *web interface*

1. Skenario 1

   Web interface menampilkan fungsi dari "*forgot Password*" yang mana saat dimasukkan *invalid account* akan dimunculkan informasi bahwa akun tidak ada. Saat akun yang *valid* berhasil dimasukkan oleh penyerang, maka *password* dapat dimasukkan secara tidak terbatas hingga ditemukkan *password* yang benar. Hal ini dapat terjadi jika tidak adanya fitur *account lockout*.

   ```
   Account john@doe.com doesn't exist
   ```

2. Skenario 2

   Web interface rentan akan *cross-site scripting*

   ```
   http://xyz.com/index.php?user=<script>alert(123)
   </script>...Response from browser is an alert popup.
   ```

Pada kedua kasus, penyerang akan dengan mudah menentukan apakan akun *valid* atau tidak serta dapat menentukan apakah situs tersebut rentan terhadap *cross-site scripting* (XSS)

Ada beberapa cara agar *web interface* menjadi lebih aman, yaitu:

1. *Password* dan *username default* dapat diubah selama *initial setup*.
2. Pastikan bahwa mekanisme *password recovery* kuat dan tidak memberi penyerang informasi yang mengindikasikan akun yang *valid*.
3. Pastikan bahwa *web interface* tidak rentan terhadap XSS, SQLi atau CSRF.
4. Pastikan bahwa *credential* tidak diekspos baik pada *traffic* jaringan internal maupun jaringan eksternal.
5. Pastikan bahwa *password* yang lemah tidak diperbolehkan.
6. Pastikan *account lockout* setelah terjadi kegagalan untuk *login* sebanyak 3-5 kali.

4.2 *Insufficient Authentication/Authorization*

Celah keamanan yang kedua adalah *Insufficient Authentication/Authorization*. Ada beberapa hal yang dapat diketahui dari celah keamanan ini. Pihak-pihak yang dapat menjadi ancaman yaitu siapapun yang memiliki akses terhadap *web interface*, *mobile interface* atau cloud *interface* termasuk pengguna internal maupun pengguna luar. Penyerang memanfaatkan *password* yang lemah, mekanisme *password recovery* yang tidak aman, *credential*s yang tidak terproteksi

dengan baik atau kurangnya kontrol akses *granular* untuk mengakses sebuah *interface* tertentu. Penyerang dapat berasal dari internal maupun pengguna luar. Kelemahan keamanan yang dapat terjadi saat proses adalah auntentifikasi tidak cukup saat terjadi penggunaan *password* yang lemah atau kurang terproteksi. Ketidak cukupan *otentikasi* merupakan suatu hal yang lazim dengan asumsi bahwa *interface* hanya dapat diekspos kepada pengguna dengan jaringan internal dan tidak untuk pengguna luar denga jaringan yang lain. Kekurangan sering ditemukan pada semua *interface*. Berbagai isu yang berhubungan dengan *otentikasi* sangat mudah diketahui saat memeriksa *interface* secara manual serta dapat ditemukan dengan *automated testing*. Ketidak cukupan *otentikasi* dapat berdampak pada data *loss* dan *corruption*, kurangnya akuntabilitas atau penolakan akses dan dapat menuntuk terhadap pengambil alihan perangkat atau akun dari *user*. Dampak pada segi bisnis adalah *customer* dapat menjadi pihak yang dirugikan.

Untuk mengetahui apakan proses *otentikasi* yang dibangun sudah cukup, ada beberapa poin yang menjadi pertimbangan.

1. Hindari untuk menggunakan *password* yang mudah, seperti "1234" serta tentukan apakah kebijakan mengenai *password* sudah cukup untuk semua *interface*.
2. Meninjau kembali *traffic* pada jaringan untuk menentukan apakah *credential* ditransmisikan dalam *clear text*.
3. Meninjau kembali persyaratan mengenai *password*, seperti kompleksitas *password*, pengecekan *password* history, waktu berakhirnya *password* serta forced *password* reset untuk pengguna baru.
4. Meninjau kembali berbagai *interface* yang ada untuk menentukan apakah *interface* diperbolehkan untuk memiliki peran yang berbeda. Sebagai contoh, semua fitur akan dapat diakses untuk administrator akan tetapi untuk *user* hanya mendapat fitur yang terbatas.
5. Meninjau kembali kontrol akses dan pengujian untuk peningkatan privillage.

Berikut 2 buah contoh dari skenario penyerangan yang dapat terjadi pada sisi *Insufficient Authentication/Authorization*

1. Skenario 1

   Interface hanya membutuhkan simple *password*.

   ```
   Username = Bob; Password - 1234
   ```

2. Skenario 2

Username dan *password* kurang terproteksi saat ditransimsikan melalui jaringan.

```
Authorization: Basic YWRtaW46MTIzNA==
```

Pada kasus kedua, penyerang akan dengan mudah menebak *password* atau dapat mengambil *credential* saat ditransmisikan melalui jaringan dan didecode karena *credential* hanya diproteksi menggunakan *encoding* Base64

Untuk membuat *Otentikasi* menjadi lebih baik, maka diperlukan beberapa hal agar *otentikasi* menjadi cukup, yaitu:

1. Pastikan bahwa *password* yang digunakan haruslah sulit ditebak.
2. Pastikan bahwa kontrol akses *granular* ada pada tempatnya saat dibutuhkan.
3. Pastikan bahwa *credential* benar-benar terproteksi.
4. Implementasikan 2 faktor *otentikasi* jika mungkin.
5. Pastikan bahwa mekanisme *password recovery* benar-benar aman.
6. Pastikan bahwa *otentikasi* ulang dibutuhkan untuk fitur-fitur sensitif.
7. Pastikan bahwa terdapat opsi untuk mengkonfigurasi kontrol *password*

4.3 *Insecure Network Services*

Celah keamanan yang ketiga adalah *Insecure Network Services*. Ada beberapa hal yang dapat diketahui dari celah keamanan ini. Pihak-pihak yang dapat menjadi ancaman yaitu siapapun yang memiliki akses terhadap perangkat melalui jaringan, termasuk internal dan eksternal *user*. Penyerang memanfaatkan *Network Services* yang mudah diserang untuk menyerang perangkat itu sendiri atau memanfaatkan perangkat tersebut untuk menyerang. Penyerangan dapat berasal dari internal atau eksternal *user*. *Network Services* yang tidak aman, rentan akan serangan buffer overflow atau serangan yang membuat terjadi penolakan terhadap *service* condition dan membuat perangkat tidak dapat diakses oleh *user*. Serangan yang membuat terjadi penolakan terhadap servis dapat difasilitasi dengan tersedianya jaringan yang tidak aman. Jaringan yang tidak aman seringkali dapat terdeteksi oleh *automated tools* seperti *port* scanners dan fuzzers. Dampak dari permasalahan tersebut adalah terjadinya data *loss* atau *corruption*, penolakan terhadap *service* atau memfasilitasi penyerang untuk menyerang perangkat lainnya. Untuk melakukan pengecekan mengenai kemanan dari jaringan yang digunakan ada 3 hal yang dapat menjadi perhatian.

1. Tentukan apakah *Network Services* yang tidak aman ada diperangkat yang kita gunakan dengan melakukan *review* untuk *open port*s menggunakan *port* scanner.
2. Jika semua *port* teridentifikasi, maka setiap *port* tersebut dapat diuji dengan menggunakan *automated tools* yang mampu melihat kerentanan dari DoS berhubungan dengan buffer overflow dan penyerangan fuzzing.
3. Melakukan *review* terhadap *port* jaringan untuk memastikan bahwa *port-port* tersebut benar-benar dibutuhkan dan jika ada *port-port* yang dimunculkan ke internet dengan menggunakan UpnP.

Ada beberapa skenario penyerangan yang mungkin terjadi pada celah *Insceure Network Services*.

1. Skenario yang pertama adalah Fuzzing Attack. Fuzzing Attack menyebabkan *Network service* atau perangkat menjadi *crash*.

   ```
   Get %s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s HTTP/1.0
   ```

2. Skenario yang kedua adalah *port* yang terbuka ke internet tanpa sepengetahuan *user* melalui UpnP.

   Port 80 dan 443 exposed to the internet via a home router.

Berdasarkan 2 skenario di atas, penyerang mampu membuat perangkat benar-benar tidak berjalan dengan memanfaatkan HTTP GET atau melakukan akses ke perangkat dengan memanfaatkan jaringan internet dengan *port* 80 dan *port* 443.

Ada beberapa cara yang dapat dilakukan untuk mengamankan *Network services*.

1. Pastikan bahwa *port-port* yang benar-benar dibutuhkan yang dimunculkan dan tersedia.
2. Pastikan bahwa servis tidak rentan terhadap buffer overflow dan fuzzing attack.
3. Pastikan bahwa servis tidak rentan terhadap DoS attack yang dapat berdampak pada perangkat yang digunakan atau perangkat lainnya serta *user* pada jaringan lokal atau jaringan lainnya.
4. Pastikan bahwa *port* atau servis jaringan tidak dimunculkan ke internet melalui UpnP sebagai contohnya.

4.4 *Lack of Transport Encryption*

Celah keamanan yang keempat adalah *Lack of Transport Encryption*. Ada beberapa hal yang dapat diketahui dari celah keamanan ini. Pihak-pihak yang dapat menjadi ancaman adalah siapapun yang mempunyai akses ke jaringan yang terhubung dengan perangkat termasuk internal

dan eksternal *user*. Penyerang memanfaatkan *lack of transport* untuk melihat data tersebut melalui jaringan. Penyerangan dapat berasal dari eksternal dan internal *user*. *Lack of Transport Encryption* memungkinkan penyerang untuk melihat data saat data tersebut sedang dalam proses pengiriman melalui *local network* atau jaringan internet. *Lack of Transport Encryption* umum terjadi pada jaringan lokal dengan asumsi *network traffic* tidak dapat dilihat secara luas, akan tetapi pada kasus jaringan lokal, salah konfigurasi pada wireless *network* membuat *traffic network* dapat dilihat siapapun yang berada dalam kawasan jaringan tersebut. Berbagai macam isu yang berkaitan dengan *transport encryption* mudah untuk diketahui dengan melihat *network traffic* dan mencari data yang dapat dibaca. *Automated tools* serta dapat digunakan untuk melihat implementasi yang sesuai untuk lazimnya *transport encryption* seperti SSL dan TLS. *Lack of Transport Encryption* dapat berdampak pada data *loss* bergantung pada data yang tersingkap serta dapat membahayakan perangkat serta akun dari *user*.

Untuk mengetahui apakah sistem yang dirancang digunakan *transport encryption* dapat melakukan *review* terhadap *network traffic* dari perangkat, aplikasi *mobile* serta cloud connection untuk mengetahui apa informasi tersampaikan dalam *clear text*. Selain itu perlu serta untuk melakukan *review* penggunaan SSL dan TLS yang *up to date* dan terimplementasi dengan seharusnya. Terakhir perlu dilakukan *review* pada penggunaan berbagai protokol enkripsi untuk memastikan bahwa protokol-protokol tersebut terekomendasi dan dapat diterima.

Ada berbagi skenario penyerangan yang mungkin terjadi pada kasus ini.

1. Skenario pertama adalah penggunaan cloud *interface* yang hanya menggunakan HTTP.
   `http://www.xyzcloudsite.com`
2. Skenario yang dapat terjadi berikutnya adalah *username* dan *password* di transmisikan ddalam *cleartext* melalui jaringan.
   `http://www.xyzcloudsite.com/login.php?userid=3&password=1234`

pada skenario di atas, penyerang mampu melihat informasi yang sifatnya sensitif disebabkan kurangnya enkripsi.

untuk menggunakan *transport encryption*, setidaknya ada 3 poin utama yang dapat dilakukan.

1. Pastikan bahwa data terenskripsi menggunakan protokol seperti SSL dan TLS saat melakukan transit melalui jaringan.

2. Pastikan sistem menggunakan teknik enkripsi dimana yang digunakan adalah teknik enkripsi yang menjadi standar oleh industri lain selama proses pengiriman data jika SSL dan TLS tidak tersedia.
3. Pastikan bahwa hanya enkripsi yang dapat diterima dan sesuai standar yang digunakan dan hindari menggunakan protokol enkripsi yang terdapat kepemilikian di dalamnya.

4.5 *Privacy Concerns*

Celah keamanan yang kelima adalah *Privacy Concerns*. Ada beberapa hal yang dapat diketahui dari celah keamanan ini. Pihak-pihak yang dapat menjadi ancaman adalah siapapun termasuk intenal dan eksternal *user* yang memiliki akses ke perangkat yang digunakan, jaringan yang terhubung dengan perangkat tersebut, aplikasi *mobile* pada perangkat, serta koneksi cloud yang ada. Penyerang menggunakan lebih dari satu *vector* seperti *insufficient* authentication, *Lack of Transport Encryption* atau *insecure network* untuk melihat data personal yang tidak terproteksi dengan baik atau dikumpulkan tanpa ada kegunaan apa-apa. Penyerangan dapat berasal dari eksternal atau internal *user*. *Privacy concern* seperti pengumpulan data personal yang terjadi karena kurangnya proteksi terhadap data tersebut lazim terjadi. *Privacy Concerns* mudah diketahui, dengan melakukan *review* terhadap data personal yang dikumpulkan. *Automated tools* dapat digunakan untuk melihat *pattern* yang spesifik dari data personal atau data sensitif lainnya yang diindikasikan terkumpul. Kumpulan dari data personal bersamaan dengan kurangnya proteksi terhadap data tersebut dapat membahayakan *user*.

Untuk mengetahui apakah perangkat yang digunakan memiliki celah pada *Privacy Concerns* setidaknya ada 4 poin pengecekan yang dapat dilakukan.

1. Indetifikasi semua tipe data yang dikumpukan oleh perangkat, aplikasi *mobile* dari perangkat serta setiap cloud *interface*.
2. Perangkat beserta kompone-komponen yang ada harusnya hanya mengumpulkan data yang diperlukan untuk menunjang fungsi yang ada.
3. Melakukan pengecekan apakah informasi personal yang dapat teridentifikasi dapat disingkap saat data tidak trenkripsi dengan benar baik selama pada media penyimpanan atau selama transit pada jaringan.
4. Melakukan *review* tentang siapa saja yang memiliki akses ke informasi personal yang dikumpulkan.

Ada beberapa skenario penyerangan yang mungkin terjadi pada kasus *Privacy Concerns*. Berikut 2 skenario yang mungkin terjadi.

1. Skenario pertama adalah pengumpulan dari data personal data.
   `Date of birth, home address, phone number, etc.`
2. Skenario kedua adalah pengumpulan informasi mengenai finansial atau kesehatan.
   `credit card data and bank account information`

kedua skenario tersebut dapat dimanfaatkan oleh pencuri atau akun yang berbahaya.

setidaknya ada 4 poin pencegahan yang dapat dilakukan dalam kasus celah dari *Privacy Concerns*.

1. Pastikan bahwa data-data yang berkaitan dengan fungsionalitas perangkat yang dikumpulkan.
2. Pastikan bahwa data yang dikumpulkan benar-benar terproteksi dengan enkripsi.
3. Pastikan bahwa perangkat dan komponen-komponennya benar-benar memproteksi informsi personal yang ada.
4. Pastikan bahwa benar-benar individu terautentikasi yang memiliki akses ke kumpulan informasi personal.

4.6 *Insecure* Cloud *Interface*

Celah keamanan yang keenam adalah *insecure* cloud *interface*. Ada beberapa hal yang dapat diketahui dari celah keamanan ini. Pihak-pihak yang dapat menjadi ancaman adalah siapapun yang memiliki akses ke internet. Penyerang menggunakan lebih dari satu *vector* seperti *insufficient* authentication, *Lack of Transport Encryption* atau *account enumeration* untuk mengakses data atau mengontrol melalui *web*site cloud. Penyerangan sebagian besar datang melalui internet. Isu *Insecure* cloud *interface* ada saat penggunaan *credential* yang mudah ditebak atau dapat terjadinya *account enumeration*. Isu ini dapat mudah diketahui dengan melakukan *review* terhadap koneksi untuk mengakses cloud *interface* dan melakukan identifikasi jika SSL digunakan atau menggunakan mekanisme reset *password* untuk mengindentifikasi akun *valid* yang akan menuntun ke *account enumeration*. *Insecure* cloud dapat membahayakan data *user* dan dapat melakukan kontrol terhadap perangkat.

Untuk mengetahui seberapa aman cloud yang digunakan, dapat dilakukan pengecekan terhadap beberapa hal.

1. Tentukan bahwa *username default* dan *password default* dapat diganti selama *initial* product *setup*.
2. Tentukan bahwa akun dari *user* akan terkunci apabila terjadi kegagalan dalam proses *login* sebanyak 3-5 kali.
3. Tentukan bahwa akun yang *valid* dapat diidentifikasi menggunakan mekanisme *password recovery* atau halaman baru *new user*.
4. Melakukan *review* terhadap *interface* untuk berbagai isu, seperti *cross-site scripting*, *cross-site request* foregery dan sql *injection*.
5. Melakukan *review* terhadap semua kerentanan dari cloud *interface*s (API *interface* dan cloud-based *web interface*)

Ada beberapa skenario penyerangan yang dapat terjadi pada isu *insecure* cloud *interface*.

1. Skenario satu adalah *password* reset menunjukkan ke*valid*an dari sebuah akun.

   ```
   Password Reset "That account does not exist"
   ```

2. *Username* dan *password* tidak terproteksi dengan baik saat ditransmisikan melalui sebah jaringan.

   ```
   Authorization: Basic S2ZjSDFzYkF4ZzOXMjM0NTY3
   ```

Pada kedua kasus tersebut, penyerang mampu mengetahui ada atau tidaknya dari sebuah akun. Serta penyerang dapat mengetahui *credential* karena hanya dilindungi oleh Base64 *encoding*.

Ada beberapa hal yang dapat dilakukan untuk melakukan pengamanan terhadap sisi cloud *interface*.

1. *Default password* dan *default username* dapat diganti selama *initial setup*.
2. Pastikan bahwa akun *user* tidak dapat dienumerasikan menggunakan fungsionalitas sistem seperti mekanisme reset *password*.
3. Pastikan bahwa akun akan terkunci setelah 3-5 kali gagal dalam percobaan *login*.
4. Pastikan bahwa cloud-based *web interface* tidak rentan terhadap XSS, SQLi atau CSRF.
5. Pastikan bahwa *credential* tidak tersingkap melalui internet.
6. Implementasikan 2 faktor *otentikasi* jika memungkinkan.

4.7 *Insecure Mobile Interface*

Celah keamanan yang ketujuh adalah *insecure mobile interface*. Ada beberapa hal yang dapat diketahui dari celah keamanan ini. Pihak-pihak yang dapat menjadi ancaman adalah siapapun

yang memiliki akses ke aplikasi *mobile*. Penyerang menggunakan lebih dari satu *vector* seperti *insufficient* authentication, *Lack of Transport Encryption* atau *account enumeration* untuk mengakses data atau melakukan kontrol melalui *mobile interface*. Adanya celah *insecure mobile interface* muncul saat *credential* yang mudah ditebak digunakan atau *account enumeration* yang mungkin terjadi. *Insecure mobile interface* mudah untuk diketahui dengan melakukan *review* koneksi terhadap wireless *network* dan mengidentifikasi jika SSL digunakan atau dengan menggunakan mekanisme reset *password* untuk mengidentifikasi akun *valid* uyang akan menuntun terhadap *account enumeration*. Hal ini akan membahayakan data *user* dan melakukan kontrol terhadap perangkat.

Untuk mengetahui apakah *mobile interface* yang digunakan aman, dapat dilakukan beberapa pengecekan.

1. Tentukan jika *default username* dan *password* dapat diganti selama proses *initial* product *setup*.
2. Tentukan jika akun *user* akan terkunci apabila terjadi kegagalan 3-5 kali dalam proses *login*.
3. Tentukan jika akun yang *valid* bisa diidentifikasi menggunakan mekanisme *password recovery* atau halaman *new user*.
4. Melakukan *review* apakah *credential* tersingkap saat terhubung dengan wireless *network*s.
5. Melakukan *review* apakah 2 faktor *otentikasi* memungkinkan.

Ada beberapa seknario penyerangan yang dapat terjadi pada isu *insecure mobile interface*.

1. Skenario satu adalah *password* reset menunjukkan ke*valid*an dari sebuah akun.
   ```
   Password Reset "That account does not exist"
   ```
2. *Username* dan *password* tidak terproteksi dengan baik saat ditransmisikan melalui sebah jaringan.
   ```
   Authorization: Basic S2ZjSDFzYkF4ZzOXMjM0NTY3
   ```

Pada kedua kasus tersebut, penyerang mampu mengetahui ada atau tidaknya dari sebuah akun. Serta penyerang dapat mengetahui *credential* karena hanya dilindungi oleh Base64 *encoding*.

Ada beberapa hal yang dapat dilakukan untuk melakukan pengamanan terhadap sisi cloud *interface*.

Untuk meningkatkan keamanan pada sisi *mobile interface*, setidaknya ada 5 poin yang dapat dilakukan.

1. *Default password* dan *default username* dapat diganti selama proses intial *setup*.
2. Pastikan bahwa *user account* tidak dapat dienumerasikan menggunakan fungsi sistem seperti mekanisme *password* reset.
3. Pastikan bahwa akun *user* akan terkunci apabila terjadi 3-5 kali kegagalan dalam proses *login*.
4. Pastikan bahwa *credential* tidak tersingkap saat terhubung dengan wireless *network*s.
5. Mengimplementasikan 2 faktor untuk proses *otentikasi* jika memungkinkan.

4.8 *Insufficient Security Configurability*

Celah keamanan yang kedelapan adalah *insufficient Security configurability*. Ada beberapa hal yang dapat diketahui dari celah keamanan ini. Pihak-pihak yang dapat menjadi ancaman adalah siapapun yang memiliki akses ke perangkat. Penyerang menggunakan kurangnya permintaan izin untuk mengakses data atau mengontrol perangkat. Penyerang serta dapat memanfaatkan kurangnya opsi enkripsi dan kurangnya opsi *password* untuk melakukan penyerangan yang dapat membahayakan perangkat atau data. Serangan dapat datang dari pengguna perangkat manapun baik sengaja atau tidak. Isu *insufficient Security configurability* timbul saat pengguna perangkat memiliki keterbatasan atau tidak sama sekali untuk merubah *Security* control dari perangkat tersebut. *Insufficient Security configurability* dapat terlihat saat *web interface* dari device tidak memiliki opsi untuk membuat izin-izin untuk *user* seperti, memaksakan untuk menggunakan strong *password*. Melakukan *review* manual pada *web interface* dan ketersediaan opsi-opsi yang dimiliki *web interface* akan menunjukkan celah keamanan dari *insufficient Security configurability*. Hal tersebut dapat membahayakan perangkat sengaja atau tidak serta dapat terjadi data *loss*.

Untuk mengetahui apakah terjadi isu *insufficient Security configurability*, dapat dilakukan beberapa pengecekan.

1. Melakukan *review* administratif pada *interface* dari device untuk menguatkan keamanan seperti menekankan pembuatan strong *password*.
2. Melakukan *review* administratif pada *interface* untuk dapat membedakan *user* admin dan *user* biasa.

3. Melakukan *review* administratif pada *interface* untuk opsi enkripsi.
4. Melakukan *review* administratif pada *interface* untuk opsi yang memungkinkan proses secure logging untuk berbagai macam *Security* event.
5. Melakukan *review* administratif pada *interface* untuk opsi yang memungkinkan peringatan dan pemberitahuan kepada end *user* untuk setiap *Security* event.

Ada beberapa skenario penyerangan yang dapat terjadi pada isu *insufficient Security configurability*.

1. Skenario pertama adalah ketidakmampuan untuk memberikan pemaksaan kebijakan penggunaan strong *password*.

   `Admins and users are allowed to create passwords for their accounts.`
2. Skenario kedua adalah ketidakmampuan untuk membolehkan enkripsi data pada saat proses transmisi.

   `Password or other sensitif data stored on the device may not be encrypted.`

Berdasarkan kasus tersebut, penyerang dapat mengambil alih kendali untuk mendapat akses ke akun user dengan *password* yang lemah atau akses ke data.

Untuk meningkatkan sisi konfiguritas dari sisi keamanan, setidaknya ada 5 poin yang dapat dilakukan.

1. Pastikan kemampuan untuk membedakan *user* biasa dari administratif *user*.
2. Pastikan kemampuan untuk melakukan enkripsi data baik saat berhenti atau saat transit pada proses transmisi data.
3. Pastikan kemampuan untuk memaksakan kebijakan penggunaan strong *password*.
4. Pastikan kemampuan untuk membolehkan proses logging pada *Security* event.
5. Pastikan kemampuan untuk memberikan pemberitahuan *Security* event pada end *user* .

4.9 *Insecure Software/Firmware*

Celah keamanan yang kesembilan adalah *insecure software/firmware*. Ada beberapa hal yang dapat diketahui dari celah keamanan ini. Pihak-pihak yang dapat menjadi ancaman adalah siapapun yang memiliki akses ke perangkat, akses ke jaringan yang terhubung dengan perangkat

serta pihak yang memperoleh akses ke *update server*. Penyerang menggunakan lebih dari satu *vector* seperti mengambil file *update* melalui koneksi yang tidak terenkripsi, file *update* yang tidak terenkripsi atau penyerang mampu untuk melakukan sendiri *malicious update* mereka melalui DNS *hijacking*. Tergantung pada metoda dari *update* dan konfigurasi perangkat, serangan dapat datang dari *local network* atau jaringan internet. Kurangnya kemampuan perangkat untuk melakukan *update* memunculkan kelemahan dari keamanan itu sendiri. Perangkat seharusnya memiliki kemampuan untuk melakukan *update* saat kemudahan untuk terkena serangan ditemukan dan *software*/*firmware update* dapat menjadi tidak aman saat file untuk melakukan proses *update* serta koneksi jaringan untuk mengirimkan file tidak terproteksi dengan baik. *Software*/*Firmware* serta dapat menjadi tidak aman jika *software*/*firmware* berisikan data yang sensitif dan bersifat hard*code*d seperti *credential*s. Isu keamanan pada sisi *software*/*firmware* cenderung lebih mudah ditemukan dengan melakukan inspeksi ke *network traffic* selama proses *update* untuk mencek enkripsi atau menggunakan sebuah hex editor untuk melakukan inspeksi pada file *update*. Permasalahan ini dapat membahayakan *user* data serta pengambil alihan kotrol terhadap perangkat dan melakukan penyerangan terhadap perangkat lainnya.

Untuk mengetahui *software*/*firmware* yang digunakan dalam sistem aman, dapat dilakukan beberapa pengecekan.

1. Yang sangat sangat penting dan paling utama, perangkat harus dapat memiliki kemampuan untuk melakukan *update* dan melakukan *update* secara *regular*.
2. Melakukan *review* pada file *update* apakah terdapat informasi yang sensitif dan dapat berupa format yang dapat dibaca manusia menggunakan hex edit *tools*.
3. Melakukan *review* terhadap produksi file *update* apakah menggunakan enkripsi dengan algoritma yang dapat diterima.
4. Melakukan *review* terhadap produksi file *update* apakah file *update* tersebut sudah disetujui.
5. Melakukan *review* terhadap metoda komunikasi yang digunakan untuk menghantarkan file *update*.
6. Melakukan *review* terhadap cloud *update server* untuk memastikan metoda enkripsi yang digunakan dalam pengiriman *up to date* dan konfigurasi yang digunakan sesuai serta *server* itu sendiri tidak rentan terhadap serangan.

7. Melakukan *review* terhadap perangkat untuk *valid*asi file *update* yang sudah ditandai.

Ada beberapa seknario penyerangan yang dapat terjadi pada isu *insufficient Security configurability*.

1. Skenario pertama adalah file *update* ditransimikan melalui HTTP.
   `http://www.xyz.com/update.bin`
2. Skenario kedua adalah file *update* tidak terenkripsi dan dapat dibaca oleh manusia.
   ◈v◈ň]◈◈Ü◈◈Qw◈ û]◈◈ƷDP◈ Ö◈◈ ∂]◈ƷDPadmin.htmadvanced.htmalarms.htm

Pada kedua kasus di atas, penyerang mampu mendapatkan file update atau file dan melihat isi konten dari file tersebut.

Untuk meningkatkan sisi keamanan pada *software*/*firmware*, setidaknya ada 6 poin yang dapat dilakukan.

1. Pastikan perangkat memiliki kemampuan untuk melakukan proses *update*.
2. Pastikan file *update* terenkripsi menggunakan metoda enkripsi yang diterima.
3. Pastikan file *update* ditransmisikan melalui koneksi yang terenkripsi.
4. Pastikan file *update* tidak berisikan data yang sensitif.
5. Pastikan file *update* terverifikasi sebelum diijinkan file *update* tersebut diupload dan diaplikasikan.
6. Pastikan bahwa *server* tempat file *update* aman.

4.10 *Poor Physical Security*

Celah keamanan yang kesepuluh adalah *poor Physical Security*. Ada beberapa hal yang dapat diketahui dari celah keamanan ini. Pihak-pihak yang dapat menjadi ancaman adalah siapapun yang memiliki *Physical* akses ke perangkat. *Vector* yang digunakan penyerang seperti *port* USB, SD card atau penyimpanan lainnya yang artinya mampu melakukan akses terhadap sistem operasi serta data yang ada di dalamnya. Kelemahan dari *Physical Security* hadir saat penyerang mampu membongkar perangkat sehingga mudah untuk mengakses media penyimpanan serta data yang ada di dalamnya. Kelemahan serta hadir saat *port* USB atau *port* eksternal lainnya digunakan untuk mengakses perangkat dengan menggunakan fitur yang dimaksudkan untuk konfigurasi atau perawatan. Hal ini dapat membahayakan perangkat itu sendiri dan data yang ada di dalamnya.

Untuk mengetahui apakah keamanan fisik dari perangkat, dapat dilakukan beberapa pengecekan.

1. Melakukan *review* seberapa mudahnya perangkat dibongkar dan media penyimpanan data diakses serta dihapus.
2. Melakukan *review port* yang digunakan seperti USB untuk menentukan apakah data bisa diakses dari perangkat tanpa harus membongkar perangkat.
3. Melakukan *review* jumlah dari *port* fisik eksternal untuk menentukan jika semua sudah memenuhi anjuran dan sesuai dengan fungsi device.
4. Melakukan *review* adminstratif terhadap *interface* untuk menentukan apakah *port* eksternal seperti USB dapat tidak digunakan.
5. Melakukan *review* adminsitratif terhadap *interface* untuk menentukan jika kemampuan administratif bisa di batasi hanya untuk akses lokal saja.

Ada beberapa seknario penyerangan yang dapat terjadi pada isu *poor Physical Security*.

1. Skenario pertama adalah perangkat dapat dengan mudah dibongkar dan media penyimpanan yang digunakan berupa SD card yang tidak terenkripsi.

   `SD card can be removed and inserted into a card reader to be modified or copied.`
2. Skenario kedua adalah USB ada pada perangkat.

   `Custom software could be written to take advantage of features such as updating via the USB port to modify the original device software.`

Pada kedua kasus, penyerang dapat mengakses *software* original pada perangkat dan membuat modifikasi atau hanya menyalin data.

Ada beberapa cara yang dapat dilakukan untuk mengamankan perangkat dari sisi fisiknya.

1. Memastikan media penyimpanan data tidak dengan mudah untuk dilepas.
2. Memastikan bahwa data yang disimpan terenkripsi.
3. Memastikan bahwa *port* USB atau *port* eksternal tidak dapat digunakan untuk mengakses perangkat secara tidak baik.
4. Memastikan bahwa perangkat tidak dengan mudah untuk dibongkar.
5. Memastikan bahwa *port* eksternal yang sesuai dengan standar seperti USB yang digunakan sesuai dengan fungsi produk.
6. Memastikan bahwa produk memiliki kemampuan untuk membatasi kemampuan administratif.

5. Kesimpulan

Internet of things merupakan sebuah kemajuan di bidang teknologi informasi akan tetapi hal tersebut berbanding lurus dengan besarnya ancaman yang mengancam para pengguna internet of things. OWASP berhasil memberikan rangkuman tentang 10 celah keamanan paling berbahaya yang dapat terjadi kepada para pengguna internet of things. Adapun beberapa pencegahan yang dapat dilakukan untuk meminimalisir ancaman dari para penyerang adalah:

1. *Password* dan *username default* dapat diubah selama *initial setup*.
2. Tidak menggunakan mekanisme yang dapat memberitahukan informasi tentang id *user* yang *valid*.
3. Pastikan bahwa jaringan yang digunakan aman dan tidak rentan terhadap serangan.
4. Pastikan terdapat fitur *lockout* yang dapat mengunsi akun setelah terjadi kegagalan untuk *login* sebanyak 3-5 kali.
5. Pastikan bahwa *credential* tidak mudah untuk tersingkap dengan berbagai pengamanan yang ada.
6. Menggunakan enkripsi yang dapat dipercaya untuk melakukan pengamanan pada data dan informasi yang akan ditransmisikan melalui jaringan.
7. Mengamankan berbagai informasi pribadi yang terkumpul agar tidak mudah untuk tersingkap.
8. Menggunakan *port-port* yang bersifat fisik dengan mengacu pada kebutuhan serta sudah diverifikasi.
9. Menggunakan informasi yang sulit ditebak untuk data yang bersifat *credential* dan *credential* tidak dengan mudah untuk tersingkap.
10. Jika memungkinkan, menggunakan 2 faktor untuk melakukan *otentikasi*.

6. Referensi

[1] O. Hersent, D. Boswarthick, and O. Elloumi, *The Internet of Things: Key applications and protocols*. Hoboken: John Wiley & Sons, 2011

[2] Alsaadi. Ebraheim, Tubalshat. Abdallah, "Internet of Things: Features, Challenges, and Vulnerabilities, International Journal of Advanced Computer Science and Information Technology (IJACSIT) Vol. 4, No. 1, 2015, Page: 1-13, ISSN: 2296-1739

[3] OWASP Internet of Things Project, 2015, [online] Available: online

[4] S. De, P. Barnaghi, M. Bauer, and S. Meissner. "*Service* modelling for the Internet of Things", Proc. of the Federeted Conference on Computer Science and Information System. September 2011.

[5] J. Gubbi, R. Buyya, S. Marusic and M. Palaniswami, "Internet of things (IoT): A vision, architectural elements, and future directions", *Future Gen. Comput. Syst.*, vol. 29, no. 7, pp. 1645-1660, 2013