PROSEDUR PENGUJIAN KESELAMATAN TEKNOLOGI API: PENYEDIAAN ALATAN DAN PENGUJIAN

WAN SHARIL SHAM BIN SHARIF
ROSSILAWATI SULAIMAN

Fakulti Teknologi & Sains Maklumat, Universiti Kebangsaan Malaysia

1. Abstrak

Ujian keselamatan adalah aktiviti yang penting dan kritikal dalam memastikan sistem teknologi maklumat bagi sesuatu organisasi itu selamat. Di era aplikasi menjadi produk utama bagi kebanyakan organisasi dalam urusan sehariannya, sudah pasti aplikasi itu sendiri membuka vektor ancaman dan serangan yang baru terhadap organisasi tersebut. Pengujian keselamatan aplikasi sudah menjadi satu perkara yang wajib dan merupakan sebahagian daripada polisi bagi sesetengah organisasi serta mungkin menjadi undang-undang bagi sesetengah negara. Antara cabang dalam pengujian keselamatan aplikasi ini adalah pengujian keselamatan teknologi API, yang merupakan muka hadapan bagi perkhidmatan web. Memandangkan teknologi API adalah trend yang baru dalam dunia pembangunan aplikasi, maka prosedur pengujiannya juga masih belum cukup matang dan tidak seragam. Kajian mengenai jenis kelemahan bagi teknologi API juga tidak menyeluruh. Kajian ini akan meneroka jenis-jenis kelemahan yang berpotensi ditemui bagi teknologi API dan mencadangkan satu prosedur pengujian yang seragam. Dengan penghasilan prosedur tersebut, penilaian terhadap keberkesanan prosedur tersebut dijalankan sebagai kajian kes. Kajian kes ini menilai keberkesanan prosedur melalui kejayaan penemuan jenis-jenis kelemahan, termasuk juga tempoh masa bagi melengkapkan pengujian dan penggunaan bebanan CPU juga dijadikan kriteria dalam penilaian prosedur tersebut. Di akhir penilaian, hasil keputusan dibentangkan kepada pakar-pakar yang dipilih untuk mendapatkan pandangan dan ulasan mereka terhadap keberkesanan prosedur pengujian yang dicadangkan dan keselariannya dengan prosedur yang diamalkan di sektor industri. Pandangan dan ulasan ini bagi mendapatkan pendapat dari mereka yang berpengalaman dalam bidang pengujian keselamatan. Hasil ulasan pakar telah berjaya mengesahkan kesesuaian garis panduan ini digunakan di peringkat industri. Satu garis panduan yang bersesuaian untuk digunakan oleh pengguna telah dibentangkan di akhir bab empat. Secara kesimpulan, prosedur pengujian yang dicadangkan berjaya dan mampu memenuhi objektif kajian ini.

2. Abstract

Security testing is an important and critical activity to ensure the information technology system of an organization is secure. In the era of apps being a major product of most organizations in their day-to-day business, it certainly opens new vectors of threats and attacks against them. The application security testing is already a mandatory matter and is part of a policy for some organizations and may be legal in some countries. One of the branches in the security testing is API technology security testing, which is the front-end for web services. As API technology is a new trend in the world of application development, its testing procedures are still incomplete and inconsistent. Studies on the types of weaknesses in API technology are also not thoroughly done. This study will explore the potential weaknesses of API technology and propose a standard testing procedure. With the development of the procedure, an assessment of the effectiveness of the procedure was conducted as a case study. This case study evaluates the effectiveness of the procedure through the successful discovery of various weaknesses, as well as the length of time to complete the test and the use of the CPU load. At the end of the assessment, the results are presented to the selected experts, which are experienced in the field of security testing, to review on the effectiveness of the proposed testing

procedure and its alignment with the procedures adopted in the industry sector. The results of expert reviews have successfully confirmed the appropriateness of the procedure with guidelines used at the industry level. An appropriate user guide has been set at the end of chapter four. In conclusion, the proposed testing procedure was successful and was able to meet the objectives of this study.

3. Pengenalan

Ujian keselamatan secara umumnya adalah proses mengenal pasti kelemahan pada setiap ruang kawalan keselamatan yang melindungi teknologi maklumat. Felderer, et al. (2016) menekankan setiap kawalan keselamatan mampu membantu melindungi tiga elemen penting dalam keselamatan teknologi maklumat iaitu tahap kerahsiaan maklumat tersebut yang mana maklumat atau data hanya mampu diakses oleh pengguna yang sah (confidentiality), setiap perubahan kepada maklumat tersebut hanya dilakukan oleh pengguna atau sistem yang mempunyai hak yang sah (integrity) dan juga maklumat boleh diakses pada setiap masa ia diperlukan (availability) (Felderer, et al. 2016).

Keselamatan teknologi maklumat merangkumi pelbagai pecahan komponen keselamatan dan dalam memastikan tadbir urus yang baik, setiap komponen memerlukan strategi yang spesifik memandangkan vektor serangan atau ancamannya juga berbeza. Alsmadi, et al. (2018) membentangkan sekurang-kurangnya terdapat tujuh komponen utama di bawah payung keselamatan teknologi maklumat iaitu keselamatan terhadap komputer, internet, komunikasi, rangkaian, aplikasi, data dan maklumat itu sendiri (Alsmadi, et al. 2018). Namun, di antara semua komponen tersebut, 84 peratus serangan siber adalah tertumpu kepada keselamatan aplikasi. Ini kerana aplikasi mempunyai vektor serangan atau ancaman yang luas, memandangkan akses kepada aplikasi tersebut adalah terbuka kepada internet dan juga pelbagai latar belakang pengguna (Synopsys 2019).

Matlamat utama di dalam keselamatan aplikasi adalah terbahagi kepada dua, iaitu memastikan aplikasi berfungsi seperti yang dikehendaki dan mampu menangkis kepelbagaian serangan atau ancaman terhadap aplikasi tersebut (McGraw 2004).

4. Penyataan Masalah

Ujian keselamatan terhadap teknologi dapat memberikan jaminan keselamatan, mengekalkan privasi data dan memastikan kualiti perkhidmatan yang stabil (Esfandyari 2015). Namun dengan perkembangan teknologi yang pantas terutamanya di alam siber, secara tidak langsung mewujudkan cabaran baru dalam menjalankan proses ujian keselamatan.

Cabaran ini juga wujud bagi menguji keselamatan teknologi API. Dalam memenuhi keperluan untuk menjalankan proses pengujian, pelbagai pihak mengkaji dan menerbitkan metodologi ujian keselamatan. Namun, kepelbagaian metodologi boleh mengakibatkan beberapa kelemahan yang terlepas pandang. Dalam pada masa yang sama proses pengujian juga kerap dilakukan menggunakan metodologi dari teknologi lain, seperti untuk menguji API tetapi menggunakan metodologi ujian bagi teknologi aplikasi web (Esfandyari 2015). Ini akan mengakibatkan hasil dari proses pengujian itu akan menimbulkan keraguan kerana tidak merangkumi keseluruhan keperluan teknologi tersebut. Oleh itu, proses pengujian perlu disokong dengan prosedur pengujian yang spesifik untuk teknologi API. Hasil dari pelaksanaan prosedur pengujian yang dicadangkan tersebut perlu memenuhi keperluan dan mengenal pasti perbezaan yang terdapat di antara teknologi API dan teknologi aplikasi web (Vithanage 2014). Dari segi keselamatan, jenis kelemahan bagi teknologi API juga berbeza dengan kelemahan yang ditemui di teknologi aplikasi web (Gunatilaka 2011). Ketika menjalankan pengujian, beberapa kriteria perlu dititikberatkan seperti perbezaan hasil keputusan di antara alatan ujian, tempoh masa melengkapkan proses pengujian dan penggunaan bebanan CPU akan menjadi kayu pengukur terhadap keberkesanan prosedur tersebut (Holik & Neradova 2017).

5. Objektif Kajian

Kajian ini akan mencapai objektif seperti berikut:

a. Mengenal pasti jenis-jenis kelemahan yang berpotensi ditemui bagi teknologi API.
b. Mencadangkan prosedur pengujian bagi teknologi API.
c. Menilai prosedur pengujian dengan ulasan pakar.
d. Mengimplementasi prosedur penyediaan alatan dan aktiviti pengujian.

6. Cadangan Penyelesaian

Merujuk kepada pembentangan di bahagian sebelum ini, didapati senarai semak jenis-jenis kelemahan diperlukan sebelum menjalankan ujian keselamatan bagi sesuatu teknologi itu. Ini memastikan dengan penyediaan senarai semak tersebut, proses ujian yang dijalankan dapat memenuhi liputan fungsi teknologi API yang pelbagai dan berbeza dengan teknologi aplikasi web (Vithanage 2014).

Penyediaan senarai semak ini juga perlu disokong dengan penyediaan prosedur pengujian yang seragam. Prosedur ini bukan sahaja perlu memenuhi keperluan senarai semak tersebut tetapi juga mampu memenuhi keperluan teknologi API yang dibangunkan untuk kepelbagaian sistem dan platform (Vithanage 2014).

Kajian-kajian terdahulu didapati banyak membuat pengujian hanya spesifik kepada beberapa jenis kelemahan yang berkaitan sahaja dan tidak menyeluruh merangkumi keseluruhan liputan fungsi teknologi API. Kajian-kajian tersebut juga tidak menyediakan satu prosedur pengujian yang lengkap seperti yang dicadangkan oleh organisasi OWASP melalui panduan pengujian keselamatannya, OWASP Testing Guide v4.0 (Meucci & Muller 2015) dan organisasi CREST melalui panduan program pengujiannya, A guide for running an effective Penetration Testing program (Creasey & Glover 2017). Kedua-dua organisasi ini adalah badan bebas yang sentiasa menyediakan panduan dalam bidang pengujian keselamatan dan juga memberikan tauliah kepada organisasi yang ingin menjadi penguji keselamatan yang bertauliah. Panduan terbitan kedua-dua organisasi tersebut juga sering diadaptasi oleh pemain-pemain industri.

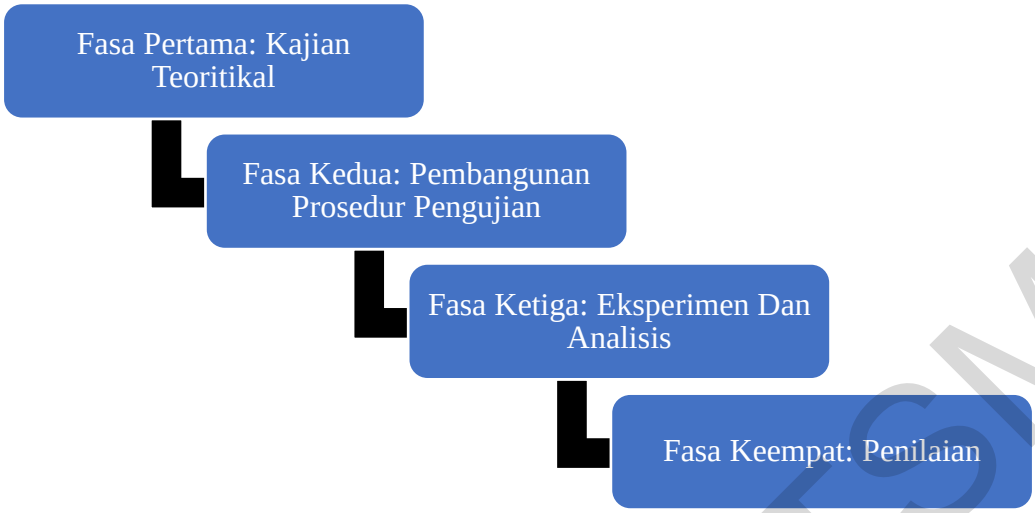
Kajian ini akan mencadangkan satu prosedur pengujian bagi menguji jenis-jenis kelemahan yang ditemui melalui kajian kesusasteraan. Sebagai kajian kes, prosedur ini akan mengimplementasi dan menguji langkah-langkah penyediaan alatan dan kaedah pengujian, seperti yang dicadangkan di dalam prosedur pengujian. Dengan merujuk hasil kajian (Holik & Neradova 2017), aktiviti ini akan dinilai berdasarkan kriteria seperti berikut:

a. Berapa banyak kelemahan yang mampu ditemui oleh alatan ujian jika dibandingkan dengan jenis kelemahan yang ditemui dalam kajian teoritikal.
b. Berapa lama masa yang diambil oleh alatan ujian untuk selesaikan proses ujian.
c. Berapa peratusan penggunaan bebanan CPU oleh alatan ujian ketika proses ujian dijalankan.

Dapatan daripada aktiviti pengujian akan dikelaskan mengikut pengkelasan risiko seperti yang dicadangkan oleh (NIST 2012). Kadar risiko yang akhir dan pelaksanaan prosedur tersebut akan dibentangkan kepada pakar yang dipilih bagi mendapatkan ulasan mengenai aktiviti secara keseluruhannya dari sudut pandang sektor industri.

7. Metodologi

Satu rangka kerja yang sistematik diperlukan bagi menyelesaikan aktiviti di dalam kajian ini. Kajian ini akan dijalankan dalam empat fasa, yang mana hasil bagi setiap fasa akan menjadi input kepada fasa yang berikutnya. Hubung kait antara fasa digambarkan seperti rajah di bawah:



Rajah 1 Rekabentuk Kajian

Jadual di bawah membincangkan butiran bagi setiap fasa, proses dan teknik yang terlibat serta hasil bagi setiap fasa, sebelum masuk ke fasa seterusnya.

Jadual 1 Perincian Setiap Fasa

Fasa	Tujuan	Proses	Teknik	Hasil
Fasa Pertama: Kajian Teoritikal	Menjalankan kajian teoritikal dengan mengenal pasti status terkini jenis-jenis kelemahan yang berkaitan dan prosedur pengujian sedia ada.	Kajian Teoritikal	• Pembacaan kajian terdahulu • Menganalisa dapatan • Perbandingan data • Mengkritik kajian terdahulu	• Latar Belakang Kajian • Penyataan Masalah • Persoalan Kajian • Objektif Kajian • Skop Kajian • Proses dan Langkah Pengujian
Fasa Kedua: Pembangunan Prosedur Pengujian	Bagi membangunkan cadangan prosedur pengujian dan mendapatkan hasil pengujian.	Penilaian dan Pengujian	• Melaksanakan aktiviti ujian melalui teknik eksperimentasi dan merujuk prosedur cadangan • Menggunakan alatan ujian yang dicadangkan	• Kaedah Pengujian • Laporan Kelemahan
Fasa Ketiga: Eksperimen Dan Analisis	Menganalisa hasil dapatan dan membandingkan di antara kumpulan data yang terhasil dari sesi ujian.	Analisa Hasil Ujian	• Menganalisa dapatan • Perbandingan data	• Laporan Analisa
Fasa Keempat: Penilaian	Menyatakan hujah pengukuhan bagi hasil kajian ini, melalui hasil dari analisis dan ulasan pakar.	Rujukan Data dan Pembuktian	• Menyatakan hujah pengukuhan	• Kesimpulan Kajian

7.1. Fasa Pertama: Kajian Teoritikal

Kajian teoritikal dijalankan bagi mengenal pasti latar belakang kajian dan maklumat yang sepadan dengannya. Fasa ini berkaitan dengan kajian teoritikal dan beberapa siri pembacaan terhadap kertas-kertas kajian, artikel, jurnal, prosiding dan sebarang maklumat di dalam talian yang berkaitan dengan ujian keselamatan teknologi aplikasi web. Fasa ini akan membantu untuk menjelaskan objektif, skop dan tujuan kajian ini.

Sebarang dapatan dari fasa ini akan dianalisa, dibanding, diulas dan dikritik supaya ia dapat mewujudkan sudut pandang yang pelbagai dan memenuhi keperluan kajian ini. Hasil dari kajian ini akan membentuk satu prosedur pengujian yang akan digunakan di fasa kedua.

7.2. Fasa Kedua: Pembangunan prosedur Pengujian

Bahagian ini akan membincangkan secara terperinci langkah-langkah di dalam prosedur yang dicadangkan. Secara dasarnya, langkah-langkah ini dipilih dengan merujuk amalan yang diamalkan di dalam industri dan disesuaikan dengan keperluan kajian ini. Prosedur ini terdiri daripada tujuh langkah yang perlu dilengkapkan oleh setiap penguji.

Sumber rujukan utama adalah kertas putih yang diterbitkan oleh (Creasey & Glover 2017) yang mana membentangkan panduan untuk membangunkan program pengujian keselamatan yang efektif oleh *The Council for Registered Ethical Security Tester* (CREST). CREST adalah badan yang memantau dan menganugerahkan akreditasi kepada penguji keselamatan yang bertauliah. Sumber berikutnya adalah panduan pengujian yang diterbitkan oleh OWASP (Meucci & Muller 2015). Di dalam panduan ini pengarang membincangkan secara terperinci langkah-langkah yang perlu diambil bagi setiap jenis kelemahan yang ingin diuji. Di akhir prosedur, setiap penemuan perlu dipadankan risiko yang sesuai seperti merujuk panduan pengkelasan risiko oleh (NIST 2012).

Aktiviti pengujian yang dijalankan akan merujuk kepada panduan pengujian oleh (Meucci & Muller 2015) yang mana pengarang menekankan supaya setiap satu jenis kelemahan perlu diuji mengikut turutannya supaya ia tidak tersasar. Jadual di bawah membentangkan padanan di antara jenis kelemahan 11 peringkat yang diterbitkan oleh (Meucci & Muller 2015).

Jadual 2 Padanan Panduan Pengujian dan Kategori Kelemahan

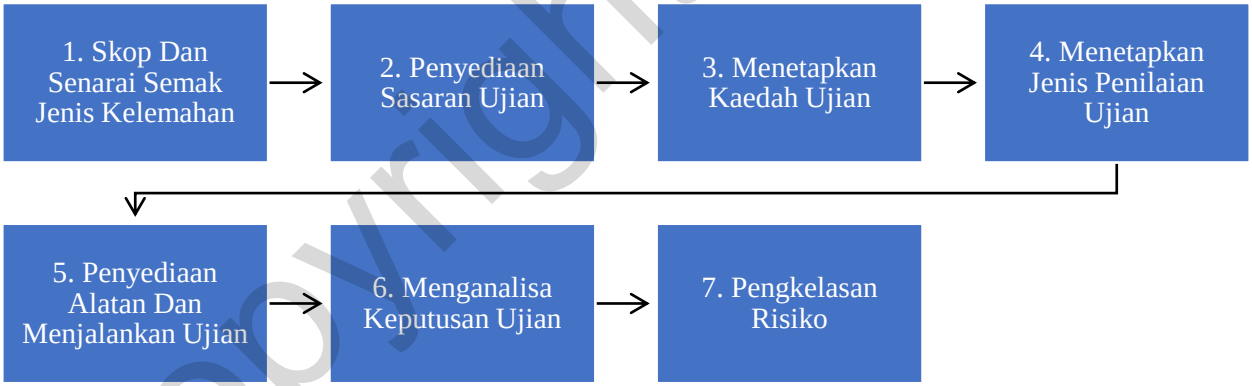
Peringkat	Kategori Kelemahan	
Peringkat Pertama: Pengumpulan Maklumat	*Verification on Digital Signature*	
Peringkat Kedua: Pengujian Pengurusan Konfigurasi	*WSDL Enumeration* *XML Structural Testing* *Not Supporting Multiple Transport Protocol*	*Insecure Communication Channel* *Data Transmit in Clear Text* *Vulnerable HTTP Content-Type*
Peringkat Ketiga: Pengujian Pengurusan Identiti	*Username / Password Input Validation*	
Peringkat Keempat: Pengujian Pengesahan	*Broken Authentication*	
Peringkat Kelima: Pengujian Kebenaran	*Insecure Direct Object References (IDOR)*	*Access to Critical Function*
Peringkat Keenam: Pengujian Pengurusan Sesi	*Cross-Site Request Forgery (CSRF)*	*JSON Web Token*

	Login-Logout Verification	
Peringkat Ketujuh: Pengujian Pengesahan Data	*API Keys Guessing* *Non-Validate Data Binding* *SQL Injection* *XPath Injection*	*XML Injection* *OS Command Injection* *Malicious SOAP Attachments*
Peringkat Kelapan: Pengujian Pengendalian Ralat	*Request / Error Message Handling*	
Peringkat Kesembilan: Pengujian Kriptografi	*Verification on TLS Certificates*	*Sensitive Information in HTTP Requests*
Peringkat Kesepuluh: Pengujian Fungsi Logik	*Rate-Limit* *Denial-of-Service Attack*	*Period of Information Delivery*
Peringkat Kesebelas: Pengujian Sisi Pemohon	*Cross Origin Resource Sharing (CORS)* *Click Jacking*	*Open Redirect* *Browser Protection on Phishing*

i. Cadangan Prosedur Pengujian

Berdasarkan penerangan seperti di bahagian sebelum ini, rajah di bawah adalah gambaran menyeluruh bagi prosedur pengujian yang dicadangkan.

Error! Reference source not found.



Rajah 2 Prosedur Pengujian Keselamatan API

Jadual di bawah merumuskan setiap langkah di rajah 2.

Jadual 3 Perincian Prosedur Pengujian Keselamatan API

Langkah	Rumusan
1. Skop Dan Senarai Semak Jenis Kelemahan	Penyediaan skop dan senarai semak jenis-jenis kelemahan yang akan diuji yang ditemui melalui sorotan susastera.
2. Penyediaan Sasaran Ujian	Penyediaan sasaran ujian API dan keperluan bagi API tersebut dipasangkan di *localhost* atau di perkomputeran awan.

3. Menetapkan Kaedah Ujian	Panduan pemilihan kaedah adalah berdasarkan keperluan seperti berikut: • Memilih *white box testing* sekiranya: o Mempunyai maklumat penuh mengenai sasaran termasuk dari segi teknologi dan organisasi syarikat. o Ingin menjalankan simulasi serangan seperti keadaan di mana penyerang adalah pekerja dalaman atau seseorang penyerang yang mempunyai hubungan terus dengan pekerja dalaman. o Ingin mengetahui keadaan prestasi dan kelemahan sasaran dari sudut penyerang dalaman serta penyerang yang mempunyai maklumat penuh mengenai sasaran. • Memilih *black box testing* sekiranya: o Tiada maklumat sasaran yang disediakan kepada penguji. o Ingin menjalankan simulasi serangan seperti keadaan di mana penyerang yang tiada maklumat mengenai sasaran. o Ingin mengetahui keadaan prestasi dan kelemahan sasaran dari sudut penyerang luaran yang tiada maklumat mengenai sasaran. • Memilih *grey box testing* sekiranya: o Mempunyai maklumat yang terhad terhadap sasaran seperti contoh hanya maklumat akses kepada teknologi. o Ingin menjalankan simulasi serangan bagi kategori penyerang yang mempunyai maklumat terhad terhadap sasaran seperti penyerang merupakan bekas pekerja bagi sesuatu organisasi tersebut. o Ingin mengetahui keadaan prestasi dan kelemahan sasaran dari sudut penyerang dalaman atau luaran yang mempunyai maklumat terhad terhadap sasaran. Bagi kajian ini, kaedah *black box testing* akan digunakan.
4. Menetapkan Jenis Penilaian Ujian	Panduan pemilihan jenis penilaian adalah berdasarkan keperluan seperti berikut: • Memilih Ujian Keselamatan Aplikasi Statik (*Static Application Security Testing – SAST*) sekiranya: o Penguji memiliki kod sumber aplikasi tersebut. o Hasil keputusan pengujian akan menemui kelemahan di setiap baris kod sumber tersebut yang ada ketika tidak mampu ditemui oleh jenis penilaian yang lain. • Memilih Ujian Keselamatan Aplikasi Dinamik (*Dynamic Application Security Testing – DAST*) sekiranya: o Aplikasi sudah dipasang di pelayar dan boleh diakses melalui URL. o Penguji tidak memiliki kod sumber aplikasi. o Hasil keputusan pengujian akan menemui kelemahan dari sudut aliran data dari satu muka ke muka laman aplikasi tersebut. o Pengujian juga merangkumi kelemahan pada setiap pautan yang mampu dicapai oleh alatan ujian tersebut. • Memilih Ujian Penembusan Manual (*Manual Penetration Testing – PenTest*) sekiranya: o Selepas selesai ujian DAST, hasilnya boleh digunakan untuk pengesahan menggunakan PenTest.

	○ Pengujian dijalankan bagi menguji fungsi logik aplikasi dengan pembuktian mampu dieksploitasi. ○ Hasil keputusan merupakan pengesahan bahawa kelemahan benar wujud dan boleh dieksploitasi. • Memilih Ujian Keselamatan Aplikasi Telefon Mudah Alih (*Mobile Application Security Testing – MAST*) sekiranya: ○ Aplikasi dibangunkan di atas sistem operasi Android dan iOS bagi kegunaan telefon mudah alih. ○ Pengujian bagi menguji aplikasi sebelum dipasang di dalam telefon mudah alih. ○ Pengujian juga akan menguji sistem sokongan dan komponen telefon mudah alih. • Memilih Ujian Keselamatan Rangkaian (*Network Security Testing – NST*) sekiranya: ○ Sasaran pengujian adalah perkakasan rangkaian. ○ Pengujian adalah bagi mengenal pasti kelemahan di peringkat rangkaian yang merupakan lapisan pertama akses dari luar atau internet. Bagi skop kajian ini, kaedah DAST akan digunakan.
5. Penyediaan Alatan Dan Menjalankan Ujian	Setiap alatan dibangunkan bersesuaian dengan jenis penilaian ujian dipilih. Pemilihan alatan ujian akan bergantung dengan keputusan pemilihan jenis penilaian seperti di atas. • Bagi jenis penilaian SAST, alatan pengujian yang sesuai digunakan seperti yang dicadangkan di (OWASP Source Code Analysis Tools 2020). • Bagi jenis penilaian DAST, alatan pengujian yang sesuai digunakan seperti yang dicadangkan di (OWASP Vulnerability Scanning Project 2019). • Bagi jenis penilaian PenTest, tiada alatan pengujian yang digunakan kerana aktiviti pengujian dijalankan secara manual. • Bagi jenis penilaian MAST, alatan pengujian yang sesuai digunakan seperti yang dicadangkan di (OWASP Mobile Security Testing Guide 2020). • Bagi jenis penilaian NST, alatan pengujian yang sesuai digunakan seperti yang dicadangkan di (Intense School Network Scanning Tools 2015). Setiap alatan yang digunakan akan disediakan menurut manual alatan tersebut. Setiap hasil ujian akan direkod untuk dianalisa di peringkat seterusnya.
6. Menganalisa Keputusan Ujian	Setiap alatan akan memberi tahap kritikal kepada setiap jenis kelemahan. Tahap kritikal tersebut akan dianalisa menurut kriteria-kriteria yang dibincangkan di bahagian sorotan susastera.
7. Pengkelasan Risiko	Berdasarkan dapatan dari langkah nombor enam, kesimpulan pengkelasan risiko akan dibuat berdasarkan metrik model risiko yang dicadangkan di jadual 5 dan pengkelasan impak dengan merujuk metrik model risiko di jadual 6.

Di akhir fasa ini, dapatan yang terhasil akan digunakan untuk aktiviti di fasa ketiga.

7.3. Fasa Ketiga: Eksperimen Dan Analisis

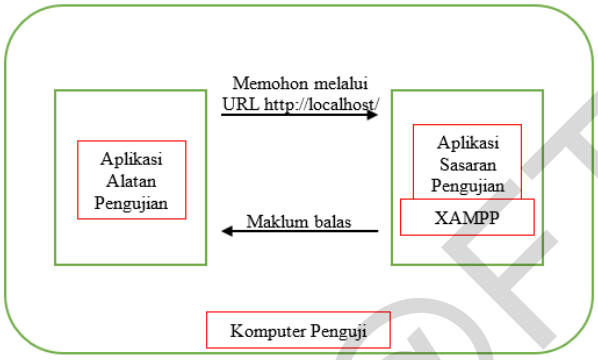
Setiap dapatan dari fasa kedua akan dianalisis dan dibandingkan di antara satu sama lain. Proses eksperimen akan dijalankan dan dapatan akan dianalisis untuk kegunaan di fasa keempat. Pada fasa ini, fokus diberikan untuk mengimplementasi Langkah 5, dalam prosedur cadangan (Rajah 2).

i. Penyediaan Ruangan Eksperimentasi

Bagi ruangan eksperimentasi, dua senario akan disediakan seperti di bawah:

Localhost

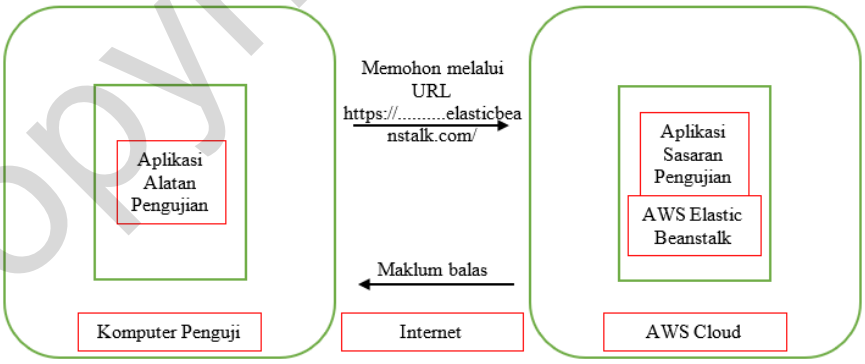
Penyediaan di *localhost* bermaksud menyediakan ruangan pelayan web di dalam komputer yang sama. Ruangan pelayan web ini menyediakan platform bagi aplikasi tersebut dilayarkan dan boleh di akses melalui pelayar di dalam komputer yang sama. Namun ia hanya terhad kepada pengguna yang mempunyai akses kepada komputer tersebut dan tidak boleh diakses melalui internet (Ray 2018). Aplikasi mampu diakses melalui alamat **http://localhost/**. Seni bina cadangan bagi penyediaan sasaran ujian di *localhost* adalah seperti berikut:



Rajah 3 Sasaran Ujian Localhost

Perkomputeran Awan

Penyediaan di awan adalah bermaksud menyediakan ruangan pelayan web di atas platform perkomputeran awan. Penyediaan ini akan memberikan aplikasi kebolehan di akses melalui internet. Setiap aplikasi akan diberikan alamat pelokasi sumber seragam (URL) yang di akses melalui pelayar dari mana-mana komputer yang mempunyai akses kepada internet. Seni bina cadangan bagi penyediaan sasaran ujian di perkomputeran awan adalah seperti berikut:



Rajah 4 Sasaran Ujian Di Perkomputeran Awan

ii. Kriteria Penilaian

Terdapat tiga kriteria yang akan dinilai di dalam kajian ini.

a. Prosedur Pengujian

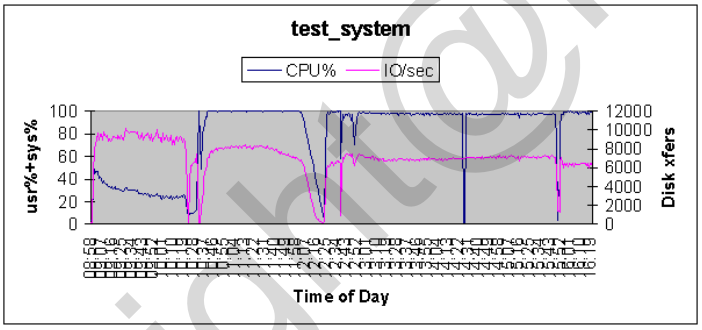
Kriteria pertama yang akan diuji, adakah prosedur pengujian ini mampu menemui setiap jenis kelemahan yang ingin diuji di dalam kajian ini. Setiap hasil dari alatan ujian akan dianalisa dan dibandingkan dengan senarai jenis kelemahan. Setiap jenis kelemahan yang ditemui, akan diambil tahap kritikalnya dan diletakkan di dalam keputusan jadual jenis kelemahan.

b. Tempoh Masa Yang Lengkapkan Oleh Alatan Pengujian

Setiap tempoh masa bagi alatan pengujian yang dijalankan akan diambil dan direkodkan. Bagi sesetengah alatan ujian telah tersedia dengan masa yang direkod. Bagi alatan ujian yang tidak mengambil masa, masa akan direkod secara manual.

c. Bebanan CPU Yang Digunakan Oleh Alatan Pengujian

Alatan ujian secara tidak langsung akan memberi kesan kepada bebanan CPU sasaran yang diimbas (Holik & Neradova 2017). Bebanan CPU juga berbeza kerana setiap alatan ujian mempunyai teknik dan strategi tersendiri dalam menjalankan imbasan. Oleh yang demikian, sepanjang tempoh pengujian, berapa penggunaan bebanan CPU akan direkodkan. Bagi tujuan ini, kajian akan menggunakan alatan *nmon analyser* yang dibangunkan oleh syarikat International Business Machines Corporation (IBM) bagi memantau untuk alatan ujian dari kategori sumber terbuka. Rajah di bawah adalah contoh bagaimana peratusan bebanan CPU dipantau dan direkodkan.



Rajah 5 Penggunaan Bebanan CPU Oleh Alatan NMON

Bagi alatan ujian dari kategori berbayar, sasaran yang digunakan akan dipasang di platform awan. Pembekal awan juga ada menyediakan modul bagi memantau penggunaan bebanan CPU dengan mengeluarkan amaran. Kajian akan menggunakan modul ini bagi memantau penggunaan bebanan CPU tersebut. Rajah di bawah adalah contoh bagaimana amaran dikeluarkan.

2019-11-03 00:44:39 UTC+0800	WARN	Environment health has transitioned from Degraded to Severe. 33.3 % of the requests are failing with HTTP 5xx.
2019-11-03 00:43:39 UTC+0800	WARN	Environment health has transitioned from Severe to Degraded. 10.9 % of the requests are failing with HTTP 5xx.

Rajah 6 Amaran Penggunaan Bebanan CPU Oleh AWS Cloud

iii. Pemilihan Sasaran Ujian

Bagi menilai keberkesanan prosedur ini, sasaran ujian yang akan digunakan adalah *Damn Vulnerability Web Services* (DVWS). DVWS adalah projek OWASP yang membangunkan API tanpa mengambil kira aspek keselamatan aplikasi. DVWS adalah API yang sedia ada lemah, dengan kata lain API tersebut

mempunyai kelemahan aplikasi yang boleh dimanipulasi dan diserang oleh penyerang atau juga penguji keselamatan (Jayaraj 2017).

Fokus utama DVWS diwujudkan adalah bagi menjadi medan sasaran untuk latihan kepada penguji keselamatan untuk mengasah bakatnya dan juga menilai keberkesanan alatan-alatan pengujian keselamatan yang baru diwujudkan. Projek DVWS ini adalah berasaskan sumber terbuka. Oleh yang demikian, terdapat beberapa versi DVWS yang diwujudkan oleh komuniti sumber terbuka. Bagi kajian ini, DVWS versi yang diwujudkan oleh snoopysecurity akan digunakan (Sanoop 2016).

iv. Kaedah Dan Teknik Pengujian

Bagi kaedah pengujian yang akan digunakan, kaedah ujian yang dipilih adalah kaedah ujian *black box*. Kaedah ini dipilih kerana ingin menjalankan ujian tanpa memahami seni bina aplikasi tersebut. Dengan berbekalkan alamat URL bagi API tersebut, ujian akan dilancarkan terus dari alatan ujian.

Bagi teknik pengujian, teknik ujian keselamatan aplikasi dinamik (DAST) akan digunakan. DAST adalah teknik pengujian yang sesuai untuk digunakan dengan kaedah *black box* (Scanlon 2018). Scanlon (2018) juga menyatakan asas ujian DAST adalah dengan menguji setiap vektor inputnya dengan mencuba kepelbagaian muatan data yang disuntik kepada aplikasi, ketika aplikasi tersebut dalam keadaan aktif serta berjalan. Alatan ujian DAST mempunyai set muatan data yang merupakan sampel-sampel kod untuk disuntik kepada aplikasi bagi menguji kelemahan-kelemahan tertentu, seperti yang digunakan oleh penyerang. Sampel kod ini terhasil dari eksperimentasi yang berterusan dan setiap penemuan sampel kod yang baru akan dikemaskini di set muatan data alatan ujian DAST tersebut. Sumber sampel kod juga adalah dari pendedahan di alam maya dari semasa ke semasa.

v. Pemilihan Alatan Ujian

Bagi memenuhi keperluan kaedah dan teknik ujian yang dicadangkan, alatan ujian DAST seperti yang dicadangkan oleh OWASP akan dikaji dan dinilai kesesuaiannya dalam kajian ini (OWASP Vulnerability Scanning Project 2019). Senarai tersebut menyenaraikan sebanyak 53 jenis alatan ujian DAST dari kategori sumber terbuka dan juga kategori yang berbayar. Dari 53 jenis alatan tersebut, sebanyak 10 alatan ujian dari kategori sumber terbuka dan dua alatan ujian dari kategori yang berbayar dipilih bagi kajian ini.

Antara ciri-ciri bagi alatan kategori sumber terbuka yang tidak terpilih adalah kerana tidak lagi diselenggara oleh pembangun alatan tersebut, tidak mampu menguji kelemahan secara menyeluruh dan hanya spesifik kepada kelemahan tertentu. Bagi alatan dari kategori berbayar yang tidak terpilih adalah kerana tiada penawaran percubaan lesen secara percuma bagi alatan ujian tersebut dan tidak mempunyai modul ujian DAST secara spesifik. Bagi pemasangan setiap alatan akan merujuk kepada dokumentasi yang disediakan di laman web rasmi alatan tersebut. Jadual di bawah menyenaraikan alatan-alatan ujian dan versi yang akan digunakan di dalam kajian ini.

Jadual 4 Alatan Ujian

Alatan Jenis Sumber Terbuka	
OWASP ZAP (v2.8.1)	Grabber (v0.1)
Nikto (v2.1.5)	Grendel-Scan (v1.0)
Arachni (v1.5.1)	GoLismero (v2.0.0b6)
Skipfish (v2.10b)	Wapiti (v3.0.0)

W3af (v2019.1.2)	Vega (v1.0)
Alatan Jenis Berbayar	
Qualys WAS (v2.42)	Rapid7 InsightAppSec (v2019.10.14)

vi. Model Risiko

Model risiko adalah merujuk kepada teknik dalam pengkelasan risiko kepada sesuatu subjek dan bagi kajian ini, model risiko akan digunakan bagi mengkelaskan risiko bagi setiap jenis kelemahan yang ditemui dari aktiviti pengimbasan yang akan dijalankan. Memandangkan hasil keputusan dari setiap alatan pengujian berkemungkinan berbeza antara satu sama lain, model risiko digunakan bagi mendapatkan kelas risiko yang disatukan dari keseluruhan hasil keputusan imbasan. Kelas risiko yang disatukan itu akan memberikan gambaran tahap impak setiap jenis kelemahan tersebut. Model risiko ini akan merujuk konsep model risiko yang dibangunkan *National Institute of Standards and Technology* (NIST 2012).

Model risiko yang akan digunakan di dalam kajian ini bagi mengkelaskan nilai kualitatif kepada nilai kuantitatif, adalah seperti di jadual di bawah:

Jadual 5 Model Risiko Menurut Tahap Kritikal

Tahap Kritikal	Nilai Kuantitatif	Penerangan
High	3	Tahap ini merujuk kepada tahap kritikal jenis kelemahan yang wujud dan sangat berpotensi untuk dieksploitasi oleh penyerang yang menemuinya.
Medium	2	Tahap ini merujuk kepada tahap kritikal jenis kelemahan yang wujud tetapi sederhana berpotensi untuk dieksploitasi oleh penyerang yang menemuinya.
Low	1	Tahap ini merujuk kepada tahap kritikal jenis kelemahan yang wujud tetapi mungkin mempunyai potensi yang rendah untuk dieksploitasi oleh penyerang yang menemuinya.

Bagi mentakrifkan skala model risiko menurut tahap impak, jumlah alatan pengujian yang digunakan perlu digandakan dengan jumlah maksimum model risiko menurut tahap kritikal. Formula bagi penghasilan skala adalah seperti berikut:

Jumlah Alatan Pengujian = a

Jumlah Nilai Tertinggi Tahap Kritikal = b

Jumlah Maksimum Skala Impak, c = a * b

Jumlah maksimum skala impak tersebut akan disesuaikan kepada tiga tahap seperti yang dibentangkan di jadual berikut. Jadual di bawah adalah dengan penggunaan lima alatan pengujian dan nilai tertinggi tahap kritikal adalah tiga.

Jadual 6 Model Risiko Menurut Tahap Impak

Tahap Impak	Skala Nilai Kuantitatif	Penerangan
High	11 - 15	Tahap ini akan memberikan impak yang tinggi kepada operasi perniagaan sehingga mampu mengakibatkan operasi terhenti secara keseluruhannya.
Medium	6 - 10	Tahap ini akan memberikan impak yang sederhana kepada operasi perniagaan dan mengganggu operasi perniagaan secara sederhana tetapi tidak menghentikan operasi.

Low	0 - 5	Tahap ini akan memberikan impak yang rendah kepada operasi perniagaan dan tidak mengganggu operasi perniagaan.

7.4. Fasa Keempat: Penilaian

Melalui keputusan di fasa ketiga akan membantu dari segi penilaian dan pembuktian bagi kajian ini. Di fasa ini, hasil dapatan akan dibentangkan kepada pakar bagi mendapatkan pandangan dan ulasan terhadap keberkesanan prosedur pengujian ini.

i. Ulasan Pakar

Ulasan pakar adalah bagi mendapatkan pandangan daripada pakar terhadap beberapa kriteria dalam menjalankan kajian ini. Satu pembentangan disediakan bagi membentangkan prosedur yang dicadangkan dan hasil keputusan pengujian kepada pakar bagi mendapatkan ulasannya. Ciri-ciri pengulas yang akan dipilih bagi ulasan pakar ini adalah seperti berikut:

i. Mempunyai pengalaman dalam bidang keselamatan komputer dari lima ke sepuluh tahun.

ii. Mempunyai pengalaman dalam bidang pengujian aplikasi dari dua ke lima tahun.

Butiran pakar yang dipilih bagi memberikan ulasan ada dilampirkan di Lampiran A. Pakar akan ditanya mengenai kriteria-kriteria seperti berikut:

i. Adakah prosedur pengujian yang dijalankan dapat menjamin ketepatan penemuan jenis-jenis kelemahan di dalam kajian ini?

ii. Adakah teknik yang digunakan di dalam kajian ini menepati piawaian seperti yang diamalkan oleh sektor industri?

iii. Adakah model risiko dan teknik pengkelasan risiko di dalam kajian ini menepati piawaian seperti yang diamalkan oleh sektor industri?

iv. Adakah secara keseluruhan prosedur ini menepati piawaian seperti yang diamalkan oleh sektor industri?

8. Keputusan

Merujuk kepada alatan ujian seperti yang disenaraikan di (OWASP Vulnerability Scanning Project 2019), terdapat dua kategori alatan ujian iaitu dari kategori sumber terbuka dan kategori berbayar. Bagi alatan ujian dari kategori sumber terbuka, terdapat 10 alatan ujian yang telah diuji bagi kajian ini. Alatan ujian dari kategori sumber terbuka akan diuji bagi sasaran *localhost*. Dari 10 alatan ujian tersebut, hanya tiga alatan ujian yang berjaya dilengkapkan dalam proses pengimbasan ini.

Manakala bagi alatan ujian dari kategori berbayar, terdapat 12 alatan ujian yang dianalisa terlebih dahulu bagi menilai kesesuaian alatan tersebut dalam kajian ini. Memandangkan alatan ini dari kategori berbayar, maka permohonan lesen percubaan perlu dimohon terlebih dahulu dari syarikat pengeluar. Alatan ujian dari kategori berbayar akan menggunakan sasaran ujian yang dipasang di platfrom awan. Hasil dari analisa mendapati tiga alatan ujian tidak menawarkan lesen percubaan, dua alatan ujian tidak menyokong spesifikasi yang diperlukan seperti tiada modul pengujian DAST dan tidak menyokong aplikasi yang dibangunkan berasaskan bahasa pengaturcaraan PHP. Baki tujuh alatan ujian mempunyai spesifikasi yang menepati keperluan kajian ini dan permohonan dilakukan secara terus kepada syarikat pengeluar alatan

tersebut. Hasilnya terdapat tiga alatan ujian diberikan lesen percubaan dan empat alatan ujian tidak diberikan lesen percubaan.

Jadual di bawah merumuskan status bagi alatan ujian sumber terbuka.

Jadual 7 Status Alatan Ujian Sumber Terbuka

Alatan Ujian	Status	Penerangan
Grabber	Tidak berjaya dipasang	Tidak mempunyai liputan kelemahan seperti yang diperlukan dan alatan tidak dikemaskini sejak dari 2006.
Grendel-Scan	Tidak berjaya dipasang	Tidak mempunyai liputan kelemahan seperti yang diperlukan dan alatan tidak dikemaskini sejak dari 2016.
Vega	Tidak berjaya dipasang	Tidak dapat dipasang kerana kehilangan komponen Python dan tidak dikemaskini sejak dari 2014.
Wapiti	Tidak berjaya dipasang	Tidak dapat dipasang kerana kehilangan komponen Python dan tidak dikemaskini sejak dari 2018.
GoLismero	Berjaya dipasang tetapi pengujian terhenti	Terhenti kerana terdapat ralat pada sistem operasi. Alatan tidak dikemaskini sejak dari tahun 2014.
Arachni	Berjaya dipasang tetapi pengujian terhenti	Terhenti kerana tidak dapat mengimbas Localhost.
W3af	Berjaya dipasang tetapi pengujian terhenti	Dihentikan selepas 12 jam pengujian berjalan. Didapati proses ujian tidak dapat disempurnakan kerana alatan menggunakan polisi dan data asas yang banyak perlu dirujuk.
OWASP ZAP	Berjaya menyempurnakan proses ujian	Berjaya menemui 23 daripada 31 jenis kelemahan. Hasil pengujian dibentangkan di jadual 9.
Nikto	Berjaya menyempurnakan proses ujian	Berjaya menemui 21 daripada 31 jenis kelemahan. Hasil pengujian dibentangkan di jadual 9.
Skipfish	Berjaya menyempurnakan proses ujian	Berjaya menemui enam daripada 31 jenis kelemahan. Hasil pengujian dibentangkan di jadual 9.

Jadual di bawah pula merumuskan status bagi alatan ujian berbayar.

Jadual 8 Status Alatan Ujian Berbayar

Alatan Ujian	Status	Penerangan
Synopsys Seeker	Tidak berjaya digunakan	Tidak menawarkan lesen percubaan.
Acunetix Website Security Scanner	Tidak berjaya digunakan	Tidak menawarkan lesen percubaan.
Netsparker Website Application Security	Tidak berjaya digunakan	Tidak menawarkan lesen percubaan.
CAST	Tidak berjaya digunakan	Tidak menyokong teknik pengujian DAST.
Contrast Security	Tidak berjaya digunakan	Tidak menyokong aplikasi yang dibangunkan di atas platform PHP.
Veracode DA	Bersesuaian tetapi tidak dapat digunakan	Tidak mendapat maklum balas mengenai permohonan lesen percubaan.

Checkmarx CxIAST	Bersesuaian tetapi tidak dapat digunakan	Tidak mendapat maklum balas mengenai permohonan lesen percubaan.
HCL AppScan	Bersesuaian tetapi tidak dapat digunakan	Tidak mendapat maklum balas mengenai permohonan lesen percubaan.
WhiteHat Security Sentinel	Bersesuaian tetapi tidak dapat digunakan	Tidak mendapat maklum balas mengenai permohonan lesen percubaan.
Fortify WebInspect	Tidak berjaya dipasang	Mendapat lesen percubaan tetapi hanya boleh mengimbas sasaran percubaan yang ditetapkan oleh Fortify sahaja.
Qualys Web Application Scanner	Berjaya menyempurnakan proses ujian	Berjaya menemui 15 daripada 31 jenis kelemahan. Hasil pengujian dibentangkan di jadual 9.
Rapid7 InsightAppSec	Berjaya menyempurnakan proses ujian	Berjaya menemui 12 daripada 31 jenis kelemahan. Hasil pengujian dibentangkan di jadual 9.

8.1. Hasil Kriteria Penilaian

Bagi kriteria penilaian, hasil pengujian bagi lima alatan ujian yang berjaya dibentangkan seperti di jadual di bawah. Bagi mendapatkan keseragaman, masa direkodkan di dalam format saat. Manakala penggunaan bebanan CPU dan peratusan kejayaan penemuan jenis kelemahan direkodkan di dalam format peratusan. Peratusan penemuan adalah hasil daripada jumlah kelemahan yang ditemui daripada jumlah keseluruhan kelemahan yang diuji di dalam kajian ini, iaitu sebanyak 31 jenis kelemahan.

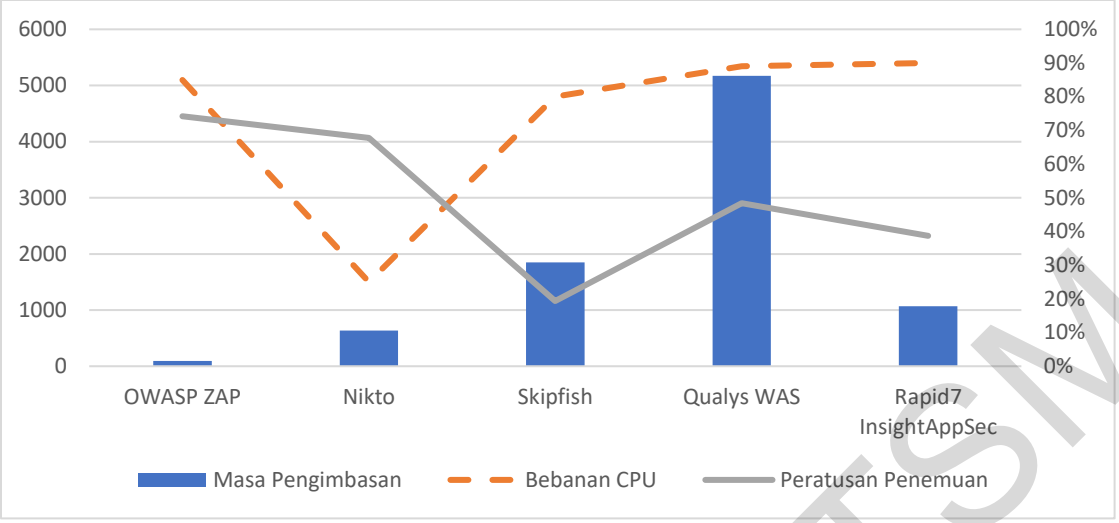
Jadual 9 Kriteria Penilaian

Kriteria Penilaian	OWASP ZAP	Nikto	Skipfish	Qualys WAS	Rapid7 InsightAppSec
Versi	2.8.1	2.1.5	2.10b	2.42	2019.10.14
Masa Pengimbasan	95 saat	633 saat	1847 saat	5172 saat	1069 saat
Bebanan CPU	85%	25%	80%	89%	90%
Peratusan Penemuan	74%	68%	19%	48%	39%

Bagi mendapatkan alatan ujian yang paling ideal, sesuatu alatan ujian tersebut perlu melengkapkan proses pengujian dalam masa yang singkat, peratusan penggunaan bebanan CPU yang rendah dan jumlah peratusan penemuan yang tinggi.

Rajah di bawah adalah carta bagi mengambarkan hasil dari jadual 9. Dari rajah tersebut menggambarkan bagi kategori alatan ujian sumber terbuka, OWASP ZAP mengambil masa yang paling pantas dalam melengkapkan proses ujian dengan peratusan penemuan yang tertinggi. Namun ia juga menggunakan bebanan CPU dengan agak tinggi. Manakala alatan ujian Skipfish mengambil masa yang paling lama untuk melengkapkan proses pengujian dan peratusan penemuan yang paling rendah. Penggunaan bebanan CPU juga tinggi.

Alatan ujian Nikto merupakan antara yang paling ideal kerana masanya tidak terlalu lama untuk melengkapkan proses ujian dengan peratusan penemuannya agak tinggi. Antara yang menjadikan Nikto sangat berkesan kerana penggunaan bebanan CPU sangat rendah berbanding alatan yang lain.



Rajah 7 Carta Kriteria Penilaian

Bagi kategori alatan ujian berbayar pula, alatan ujian Qualys WAS mengambil masa yang paling lama bagi melengkapkan proses pengujian dan penggunaan bebanan CPU juga agak tinggi. Hasil keputusan juga tidak membantu kerana peratusan penemuannya rendah berbanding dengan penemuan alatan ujian kategori sumber terbuka. Alatan ujian Rapid7 InsightAppSec dapat melengkapkan proses pengujian lebih cepat berbanding Qualys WAS tetapi menggunakan bebanan CPU lebih tinggi. Peratusan penemuan kelemahan juga rendah berbanding Qualys WAS.

8.2. Hasil Penemuan Kelemahan

Alatan ujian yang mampu menemui kelemahan sebanyak yang mungkin akan menjadikan alatan ujian tersebut antara yang ideal dan konsisten untuk digunakan dalam proses pengujian ini.

i. Liputan Penemuan Kelemahan

Merujuk kepada setiap hasil alatan ujian, tiada alatan yang mampu menemui kesemua jenis kelemahan yang diuji di dalam kajian ini. Namun sekiranya semua hasil penemuan alatan ujian tersebut digabungkan, kajian mampu menemui 30 daripada 31 jenis kelemahan. Dari peratusan penemuan ia hampir 97 peratus. Seperti di jadual di bawah, yang mana 'X' adalah merujuk kepada jenis kelemahan yang berjaya ditemui oleh alatan ujian tersebut.

Jadual 10 Hasil Pengujian Jenis Kelemahan

Jenis Kelemahan	OWASP ZAP	Nikto	Skipfish	Qualys WAS	Rapid7 Insight App Sec	Berjaya ditemui?
Insecure Direct Object References (IDOR)		X				Ya
Verification on Digital Signature						Tidak
Username / Password Input Validation	X	X	X	X	X	Ya
Broken Authentication	X	X	X		X	Ya

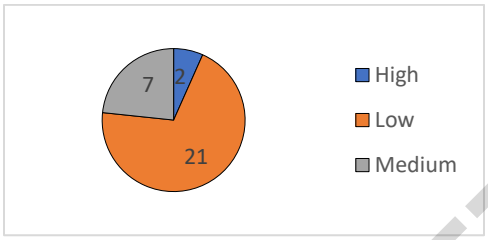
Cross-Site Request Forgery (CSRF)	X		X		X	Ya
Login-Logout Verification		X				Ya
JSON Web Token	X					Ya
API Keys Guessing	X	X	X			Ya
Verification on TLS Certificates	X					Ya
WSDL Enumeration				X		Ya
Request / Error Message Handling	X	X		X	X	Ya
Sensitive Information in HTTP Requests	X	X		X	X	Ya
XML Structural Testing			X			Ya
Rate-Limit	X		X	X		Ya
Denial-of-Service Attack	X	X		X		Ya
Period of Information Delivery				X		Ya
Access to Critical Function	X	X				Ya
Non-Validate Data Binding	X					Ya
Not Supporting Multiple Transport Protocol		X				Ya
Insecure Communication Channel	X	X		X	X	Ya
Data Transmit in Clear Text	X	X		X	X	Ya
Vulnerable HTTP Content-Type	X	X		X	X	Ya
Cross Origin Resource Sharing (CORS)	X	X		X	X	Ya
Click Jacking	X	X		X	X	Ya
Open Redirect	X	X		X	X	Ya
Browser Protection on Phishing	X					Ya
SQL Injection	X	X				Ya
XPath Injection	X	X		X		Ya
XML Injection	X	X		X		Ya
OS Command Injection	X	X			X	Ya
Malicious SOAP Attachments		X				Ya

ii. Tempoh Masa Keseluruhan

Seperti dibincangkan pada bab sebelum ini, bagi mendapatkan peratusan penemuan yang tertinggi adalah dengan menggabungkan kesemua keputusan alatan ujian tersebut. Oleh yang demikian, tempoh masa bagi melengkapkan proses pengujian kelima-lima alatan ujian tersebut adalah **8,816 saat** atau bersamaan dengan **dua jam, 26 minit dan 56 saat**.

iii. Pengkelasan Risiko Dan Impak

Dengan merujuk kepada model risiko seperti di bab dua, pengkelasan nilai kuantitatif dilakukan bagi setiap hasil penemuan dan setiap nilai tersebut dijumlahkan bagi mendapatkan tahap impak sesuatu kelemahan tersebut. Hasil dari aktiviti tersebut, kajian mendapati terdapat dua jenis kelemahan yang dikategorikan sebagai berisiko memberi impak yang tinggi, tujuh jenis kelemahan dikategorikan sebagai berisiko sederhana untuk memberikan impak dan 21 jenis kelemahan dikategorikan sebagai berisiko rendah untuk memberikan impak kepada aplikasi tersebut. Rajah di bawah memberi gambaran tahap impak dari hasil kajian ini.



Rajah 8 Carta Tahap Impak

Perincian bagi setiap jenis kelemahan, penemuan bagi setiap alatan ujian, nilai skala tahap kritikal, jumlah nilai skala keseluruhan dan tahap impak dibentangkan seperti di dalam jadual seperti di bawah.

Jadual 11 Perincian Tahap Impak

Jenis Kelemahan	OWASP ZAP	Nikto	Skipfish	Qualys WAS	Rapid7 Insight App Sec	Nilai Skala	Tahap Impak
Insecure Direct Object References (IDOR)	0	2	0	0	0	2	Low
Verification on Digital Signature	0	0	0	0	0	0	Low
Username / Password Input Validation	3	3	1	2	2	11	High
Broken Authentication	3	3	1	0	2	9	Medium
Cross-Site Request Forgery (CSRF)	1	0	2	0	1	4	Low
Login-Logout Verification	0	1	0	0	0	1	Low
JSON Web Token	3	0	0	0	0	3	Low
API Keys Guessing	3	3	1	0	0	7	Low
Verification on TLS Certificates	1	0	0	0	0	1	Low
WSDL Enumeration	0	0	0	1	0	1	Low
Request / Error Message Handling	1	1	0	2	1	5	Low
Sensitive Information in HTTP Requests	1	1	0	3	3	8	Medium

XML Structural Testing	0	0	1	0	0	1	Low
Rate-Limit	2	0	2	1	0	5	Low
Denial-of-Service Attack	2	2	0	1	0	5	Low
Period of Information Delivery	0	0	0	1	0	1	Low
Access to Critical Function	3	2	0	0	0	5	Low
Non-Validate Data Binding	3	0	0	0	0	3	Low
Not Supporting Multiple Transport Protocol	0	1	0	0	0	1	Low
Insecure Communication Channel	1	1	0	1	1	4	Low
Data Transmit in Clear Text	1	1	0	2	1	5	Low
Vulnerable HTTP Content-Type	1	1	0	1	1	4	Low
Cross Origin Resource Sharing (CORS)	2	2	0	1	1	6	Medium
Click Jacking	2	2	0	2	1	7	Medium
Open Redirect	3	3	0	3	3	12	High
Browser Protection on Phishing	1	0	0	0	0	1	Low
SQL Injection	2	3	0	0	0	5	Low
XPath Injection	2	3	0	3	0	8	Medium
XML Injection	2	3	0	3	0	8	Medium
OS Command Injection	3	3	0	0	3	9	Medium
Malicious SOAP Attachments	0	2	0	0	0	2	Low

8.3. Ulasan Pakar

Bagi mendapatkan pandangan dan pengesahan terhadap prosedur pengujian yang dijalankan, satu pembentangan dibentangkan kepada empat orang pakar yang berpengalaman dalam bidang keselamatan teknologi maklumat. Dapatan dari ulasan pakar dibentangkan di dalam jadual di bawah.

Jadual 12 Dapatan Ulasan Pakar

Soalan	**Pakar Satu**	**Pakar Dua**	**Pakar Tiga**	**Pakar Empat**	**Peratusan**
Merujuk kepada pembentangan di atas, adakah prosedur pengujian yang dijalankan dapat menjamin ketepatan penemuan jenis-jenis kelemahan di dalam kajian ini?	Ya	Ya	Ya	Ya	100%
Adakah teknik dan kaedah yang digunakan selari dengan piawaian yang diamalkan oleh sektor industri?	Ya	Ya	Ya	Ya	100%
Adakah model risiko dan teknik pengkelasan risiko di dalam kajian ini menepati piawaian seperti yang diamalkan oleh sektor industri?	Ya	Ya	Ya	Ya	100%

Adakah secara keseluruhan prosedur ini menepati piawaian seperti yang diamalkan oleh sektor industri?	Ya	Ya	Ya	Ya	100%

9. Kesimpulan

Kajian ini mencadangkan prosedur pengujian teknologi API dan menguji keberkesanannya dengan analisa hasil pengujian oleh alatan ujian, pengkelasan risiko dengan merujuk model risiko yang dicadangkan NIST dan mendapatkan pandangan pakar mengenai bagaimana keseluruhan prosedur ini dijalankan.

9.1. Perbincangan Mengenai Persoalan Kajian

Merujuk kepada persoalan kajian, berikut adalah perbincangan terhadap setiap persoalan:

i. Adakah jenis-jenis kelemahan teknologi API dari kajian teoritikal selari dengan yang digariskan oleh organisasi OWASP?

Jenis-jenis kelemahan telah berjaya diuji dan ditemui oleh alatan pengujian yang digunakan di dalam kajian ini. Melalui dapatan dari pengujian, hasil dari gabungan kelima-lima alatan ujian tersebut berjaya menemui hampir 97 peratus kejayaan iaitu 30 daripada 31 jenis kelemahan.

Perkara ini juga disokong dengan dapatan dari ulasan pakar seperti yang dibentangkan di jadual 12. Kesemua pakar bersetuju dan berpuas hati dengan hasil keputusan pengujian ini.

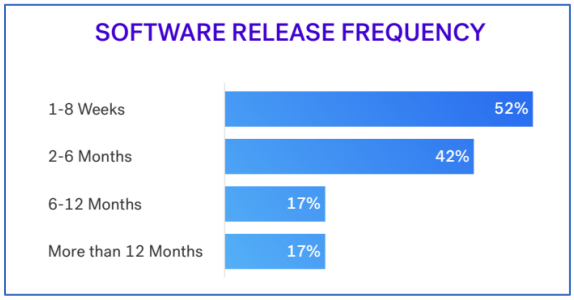
Secara kesimpulannya, persoalan kajian ini berjaya dirungkaikan di dalam kajian ini.

ii. Apakah jenis prosedur pengujian yang sesuai bagi teknologi API?

Kajian telah mencadangkan satu prosedur pengujian bagi teknologi API. Prosedur tersebut telah diuji dengan kriteria-kriteria seperti berikut:

a. Berapa lama masa yang diambil oleh alatan ujian untuk selesaikan proses ujian.

Bagi mendapatkan peratusan penemuan yang maksimum, maka hasil keputusan bagi kelima-lima alatan ujian perlu digabungkan. Oleh yang demikian, tempoh masa yang digabungkan bagi kelima-lima alatan ujian itu adalah **8,816 saat** atau bersamaan dengan **dua jam, 26 minit dan 56 saat**. Ini bermaksud, bagi setiap tempoh pembangunan aplikasi API, diperlukan sekurang-kurangnya **dua hingga tiga jam** bagi melengkapkan proses pengujian keselamatan aplikasi tersebut. Tempoh masa ini adalah relevan jika dibandingkan dengan kajian yang dijalankan oleh (Wong & Shema 2018), yang mana mendapati 52 peratus aplikasi dibangunkan antara **satu hingga lapan minggu**. Rajah di bawah melaporkan keputusan penuh dapatan kajian tersebut.



Rajah 9 Tempoh Pembangunan Aplikasi

b.Berapa peratusan penggunaan bebanan CPU oleh alatan ujian ketika proses ujian dijalankan.

Tiada alatan ujian yang mencecah 100 peratus dan boleh mengakibatkan aplikasi yang diuji gagal sepenuhnya dan tidak dapat diakses. Prosedur ini juga dijalankan secara berturutan, iaitu setiap satu alatan ujian beroperasi dan perlu lengkap sebelum alatan ujian seterusnya dilancarkan. Oleh yang demikian, penggunaan bebanan CPU dianggap efisyen dan ideal bagi alatan ujian yang digunakan.

iii. Adakah melalui perbandingan hasil pengujian, penggunaan prosedur pengujian dapat menemui keseluruhan jenis kelemahan tersebut?

Prosedur ini mampu menemui 30 daripada 31 jenis kelemahan yang diuji dengan menggabungkan kesemua hasil keputusan alatan ujian. Oleh yang demikian, prosedur ini mampu mencapai 97 peratus kejayaan penemuan jenis kelemahan. Bagi mengisi kekurangan satu lagi jenis kelemahan iaitu *Verification on Digital Signature*, (Meucci & Muller 2015) ada menggariskan teknik bagi memeriksa butiran itu secara manual. Bagi mengisi jurang ini, jenis kelemahan tersebut tidak mampu ditemui dengan menggunakan alatan ujian tetapi mampu dikenalpasti secara manual.

iv. Adakah pakar berpandangan prosedur pengujian selari dengan yang diamalkan di sektor industri?

Keseluruhan pakar berpandangan bahawa prosedur ini selari dengan yang diamalkan di sektor industri. Cadangan penambahbaikan mampu mengukuhkan lagi hasil pengujian prosedur ini dan boleh dijadikan sebagai jurang kajian bagi masa hadapan.

9.2. Perbincangan Mengenai Objektif Kajian

Merujuk kepada objektif kajian, berikut adalah perbincangan terhadap setiap objektif di dalam kajian ini:

i. Mengenal pasti jenis-jenis kelemahan yang berpontesi ditemui bagi teknologi API.

Kajian ini telah mampu menerbitkan jenis-jenis kelemahan yang berpotensi memberi impak kepada teknologi API. Perkara ini telah disokong dengan pembacaan dan huraian di sorotan susastera. Ia juga telah dibuktikan melalui aktiviti pengujian yang dijalankan dan hasilnya telah disokong oleh pakar-pakar yang mempunyai pengalaman di dalam bidang pengujian aplikasi. Sehubungan dengan itu kajian ini telah mencapai objektif bagi mengenal pasti jenis-jenis kelemahan teknologi API.

ii. Mencadangkan prosedur pengujian bagi teknologi API yang selari dengan yang diamalkan di sektor industri.

Melalui sorotan susastera, satu prosedur pengujian telah dicadangkan dan proses pengujian bagi kajian ini dilaksanakan dengan mengikuti prosedur tersebut. Hasil keputusan pengujian yang didapati telah dipadankan dengan hasil dari sorotan susastera dan mendapat keputusan yang positif. Setiap langkah di dalam prosedur dicadangkan telah dinilai oleh pakar-pakar dan mereka menyokong bahawa prosedur tersebut selari dengan yang diamalkan oleh pemain industri. Sehubungan dengan itu, kajian ini juga telah mencapai objektif bagi menghasilkan prosedur pengujian bagi teknologi API dan selari seperti yang diamalkan di sektor industri.

iii. Melaksanakan prosedur serta menilai prestasi melalui perbandingan hasil pengujian dan ulasan pakar.

Melalui pembentangan kepada barisan pakar, kesemua mereka bersetuju dengan prosedur pengujian yang dicadangkan dan juga hasil keputusan ujian. Kesemua pakar juga sepakat bahawa prosedur ini secara keseluruhannya mencapai piawaian seperti yang diamalkan di peringkat industri. Sebarang jurang dan cadangan penambahbaikan dari barisan pakar boleh dijadikan sebagai objektif kajian di masa hadapan.

9.3. Kesimpulan

Kajian ini dapat disimpulkan bahawa telah berjaya mencapai objektifnya. Setiap persoalan berjaya dirungkaikan dan dihuraikan di bahagian 9.1. Dengan menggunakan prosedur yang dicadangkan, satu keputusan yang lengkap dan efisyen berjaya dicapai dengan menggabungkan ke semua keputusan lima alatan ujian yang diuji di dalam kajian ini. Tempoh masa yang diambil bagi alatan ujian melengkapkan proses ujian juga bersesuaian dan boleh diadaptasikan dengan merujuk perbincangan di atas.

Bagi penambahbaikan di masa hadapan, prosedur pengujian ini boleh diuji bagi alatan ujian yang berbeza dari yang digunakan di dalam kajian ini. Terdapat pelbagai jenis alatan yang dibangunkan di pasaran. Sasaran pengujian juga boleh dilebarkan bagi aplikasi yang menggunakan teknologi yang berbeza. Jenis-jenis kelemahan juga boleh diterokai melalui terbitan kajian-kajian yang baru.

Barisan pakar juga ada mencadangkan dari sudut mendapatkan keputusan yang lebih komprehensif seperti menggabungkan kaedah pengujian seperti *white box* dan *grey box*. Bagi kajian ini, pengujian hanya menggunakan kaedah *black box*. Menurut pakar pertama, dari sudut pembuktian konsep ia telah tercapai di peringkat ini.

Pakar kedua juga mencadangkan supaya menilai prosedur ini dengan mensasarkan sistem operasi yang berbeza seperti Microsoft Windows dan Unix. Ini bagi membuktikan adakah prosedur pengujian ini boleh diguna pakai untuk sistem operasi yang berbeza.

10. Rujukan

Alsmadi, Izzat, Robert Burdwell, Ahmed Aleroud, Abdallah Wahbeh, Mahmood Al-Qudah, and Ahmad Al-Omari. 2018. "Introduction to Information Security." *Practical Information Security* 1-16.

Creasey, Jason, dan Ian Glover. 2017. *A guide for running an effective Penetration Testing programme.* White Paper, CREST.

Esfandyari, Azadeh. 2015. "A comparative study and classification on web service security testing approaches." *Advances in Computer Science: an International Journal* (ACSIJ) 4 (4): 46-50. Diakses August 9, 2019. https://pdfs.semanticscholar.org/f48d/ef21144dcc82cdc72339d626f076c59a9be9.pdf.

Felderer, Michael, Matthias Buchler, Martin Johns, Achim D. Brucker, Ruth Breu, and Alexander Pretschner. 2016. "Security Testing: A Survey." *Advances in Computers* 101: 1-43.

Gunatilaka, Dolvara. 2011. "A Survey of Privacy and Security Issues in Social Networks." Diakses August 30, 2019. https://www.cse.wustl.edu/~jain/cse571-11/ftp/social.pdf.

Holik, F., dan S. Neradova. 2017. "Vulnerabilities of modern web applications." *2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO).* Opatija, Croatia: IEEE.

Intense School Network Scanning Tools. 2015. *Top 10 Network Scanning Tools.* October 19. Accessed January 25, 2020. http://resources.intenseschool.com/top-10-network-scanning-tools/.

Jayaraj, Abhineet. 2017. *OWASP Damn Vulnerable Web Sockets (DVWS).* Diakses September 14, 2019. https://www.owasp.org/index.php/OWASP_Damn_Vulnerable_Web_Sockets_(DVWS).

McGraw, Gary. 2004. "Software Security Testing." *IEEE SECURITY & PRIVACY* 81-85.

Meucci, Matteo, dan Andrew Muller. 2015. *OWASP Testing Guide v4.* The OWASP Foundation.

NIST. 2012. *Guide for Conducting Risk Assessments.* NIST Special Publication 800-30 Rev 1, Gaithersburg, MD: NIST.

OWASP Mobile Security Testing Guide. 2020. *Mobile Security Testing Guide.* Accessed January 25, 2020. https://owasp.org/www-project-mobile-security-testing-guide/.

OWASP Source Code Analysis Tools. 2020. *Source Code Analysis Tools.* Accessed January 25, 2020. https://owasp.org/www-community/Source_Code_Analysis_Tools.

OWASP Vulnerability Scanning Project. 2019. "Vulnerability Scanning Tools." *OWASP.* 12 November. Diakses November 23, 2019. https://www.owasp.org/index.php/Category:Vulnerability_Scanning_Tools.

Ray, Partha Pratim. 2018. "An Introduction to Dew Computing: Definition, Concept and Implications." *IEEE Access* 6: 723-737.

Sanoop, Sam. 2016. "Damn Vulnerable Web Services (DVWS)." *Github.* Diakses November 23, 2019. https://github.com/snoopysecurity/dvws.

Scanlon, Thomas. 2018. “10 Types of Application Security Testing Tools: When and How to Use Them.” *Software Engineering Institute*. 9 July. Diakses November 23, 2019. https://insights.sei.cmu.edu/sei_blog/2018/07/10-types-of-application-security-testing-tools-when-and-how-to-use-them.html.

Synopsys. 2019. *11 Best Practices to Minimize Risk and Protect Your Data.* Datasheet, Synopsys.

Synopsys. 2019. *Managed Application Security Testing (AST).* Datasheet, San Francisco: Synopsys, Inc. Diakses August 16, 2019. https://www.synopsys.com/content/dam/synopsys/sig-assets/datasheets/managed-application-security-services-datasheet.pdf.

Synopsys. 2019. *OWASP Top 10 web application security risks.* Technical Report, Synopsys. Diakses August 28, 2019. https://www.synopsys.com/blogs/software-security/owasp-top-10-application-security-risks/.

Vithanage, Asanka. 2014. “How to Efficiently Test Service Oriented Architecture.” *WSO2*. 11 April. Diakses August 29, 2019. https://wso2.com/library/articles/2014/04/how-to-efficiently-test-service-oriented-architecture/.

Wong, Caroline, dan Mike Shema. 2018. *Pen Test Metrics 2018 - Data from a Pen Testing as a Service Platform.* Survey Data, Cobalt.io.