Final Cyber Jawara 2020

Statistik & Solusi

Problem Setter

Reverse Engineering

Malware	Visat
Inothing	Visat
Suspicious	Fariskhi

Problem Setter

Pwn / Exploit

No Syscall	Fariskhi
Maze Solver	Fariskhi
Sorting Game	Fariskhi
Lab Virus	Fariskhi
Airport System	Fariskhi

Problem Setter

Web

Toko Masker 4	Fariskhi
WHO	Fariskhi
WHO 2	Fariskhi
COVID-19 Statistics	Visat
Tic Tac Toe	Visat
CJ Travel	Visat Fariskhi

Dynamic Scoring

Poin untuk suatu soal dihitung berdasarkan:

$$f(x) = \frac{b-a}{s^2}x^2 + a$$

x = jumlah tim yang menyelesaikan soal tersebut
a = 1000 (max point)
b = 300 (min point)
s = 20 (threshold)

Statistik

First Solve

ANYA GERALDINEEE

Soal WHO, 9:27:45 WIB

First RE Finisher

Dih males

Menyelesaikan semua soal RE pada 14:25:40 WIB

First Pwn Finisher

ROP Revenge

Menyelesaikan semua soal Pwn pada 19:28:23 WIB

First Solves

Malware	FUM, 12:42:46
Inothing	ZEN, 11:03:46
Suspicious	FUM, 11:08:28
No Syscall	魔王, 9:30:42
Maze Solver	Ahlul Jannah, 10:56:16
Sorting Game	Ahlul Jannah, 18:02:23
Lab Virus	魔王, 12:49:44

Airport System	ROP Revenge, 12:13:41
WHO	ANYA GERALDINEE, 11:03:46
Toko Masker 4	Apa Nih?, 11:00:23
COVID-19 Statistics	VtuberSIMP, 11:10:39
Tic Tac Toe	ZEN, 18:37:52
CJ Travel	VtuberSIMP, 17:56:22
WHO 2	VtuberSIMP, Freeze Time

Pembahasan

Intended solution dari problem setter

Malware (RE, 972, *Solved by 5 teams*)

- Windows Malware yang dibuat dengan .NET dan di-obfuscate dengan ConfuserEx.
- Binary yang diberikan adalah dropper dan memiliki anti-debugger. Dropper kemudian menaruh child. Child juga punya anti-debugger. Child adalah ransomware yang melakukan enkripsi.
- Dropper dan child berkomunikasi melalui AnonymousPipeServerStream. Dropper mengirimkan nama direktori yang akan dienkripsi dan key.
- Child menerima parameter dari dropper dan mengenkripsi menggunakan AES CBC dengan IV yang static.
- Dari static analysis atau dynamic analysis (dengan mematikan anti-debugger terlebih melalui patch) bisa didapatkan key dan IV. File yang terenkripsi kemudian bisa didekripsi.

Inothing (RE, 993, *Solved by 3 teams*)

- Arduino binary hex dump (AVR).
- Jika dijalankan di emulator atau dengan arduino hardware, program akan diam saja.
- Ada banyak tools yang dapat digunakan untuk static analysis (misal, avr-objdump).
- Kode asli program:

```
void loop() {
 unsigned char x[] = { 6, 14, 117, 118, 115, 112, 56, 49, 33, 127, 124, 62, 120, 38, 12, 21, 97, 24, 27, 9, 53,
100, 42, 47 };
 unsigned char i = 0;
 for (; i < 24; i++) {
   unsigned char y = i ^ 69;
   unsigned char z = x[i] ^ y;
   delay(z);
 }
}
```

- Dengan static analysis pada AVR assembly, flag bisa didapatkan dengan xor.

Suspicious (RE, 993, *Solved by 3 teams*)

- Backdoor Mach-o yang dibuat dengan Golang. Golang banyak dipakai oleh APT actor atau malware dev karena mudah untuk membuat malware yang reliable.
- Pada soal ini, ada debug symbol sehingga static analysis lebih mudah.
- Inti dari backdoor:
    - Menjalankan HTTP server dan me-register handler untuk “/robots.txt”
    - Handler akan membaca header X-Fowarded-For dan men-decode base64 sebanyak 3 kali. Katakanlah hasilnya A.
    - Selanjutnya header User-Agent dibaca sebagai B.
    - Backdoor akan mengeksekusi exec(A, B)
- Solusi:

```
curl http://1337.cyber.jawara.systems:5522/robots.txt -H "X-Forwarded-For: `printf cat|base64|base64|base64`" -H "User-Agent: flag.txt"
```

No Syscall (Pwn, 552, *Solved by 17 teams*)

- Service akan menjalankan shellcode dari input tapi apabila syscall dipanggil (execve, open, write, etc.) maka service akan berhenti.
- Alamat memori flag pada heap ditampilkan oleh service (berbeda-beda pada tiap eksekusi karena ASLR).
- Solusi:
    - Tulis shellcode untuk memindahkan byte pertama dari alamat memori flag ke suatu register. Brute force isi register tersebut. Apabila salah, jump ke infinite loop atau segfault sehingga diketahui perbedaannya. Lakukan hingga dapat byte yang benar dan lanjut ke byte berikutnya hingga dapat flag.
    - Untuk mempercepat brute force, dapat digunakan algoritma binary search dengan kompleksitas O(log N) (hanya 8 kali brute force tiap byte).

Maze Solver (Pwn, 957, *Solved by 6 teams*)

- Program menggunakan algoritma DFS (depth first search) rekursif untuk menyelesaikan labirin kita secara otomatis. Urutan rekursi: bawah, atas, kiri, kanan.
- Ada out of bound pada pengecekan apakah cell saat ini masih dalam range atau bukan. Pencarian jalan bisa keluar dari map.
- Map labirin dialokasikan di dalam stack. Beberapa bytes setelah map, ada slot yang dipakai untuk menyimpan return address.
- Cell awal dan tujuan DFS tidak bisa keluar dari maze, tapi penelusurannya bisa keluar (out of bound). Jalur solusi maze akan digambar menggunakan karakter yang bisa kita pilih. Hal ini bisa dimanfaatkan untuk mengoverwrite return address byte per byte. Selanjutnya bisa ROP untuk mengontrol eksekusi program.
- Karena di program sudah ada fungsi untuk membaca file dan menulis isinya, kita dapat memanfaatkan fungsi tersebut untuk membaca isi “/flag.txt”

Sorting Game (Pwn, 993, *Solved by 3 teams*)

- 32 bilangan 64 bit dialokasikan oleh program di dalam stack. Sebelum 32 bilangan tersebut, ada pointer yang menunjukkan alamat “cost” yang berada di stack juga.
- Pada saat program meminta user untuk memasukkan index 1-32 untuk memilih bilangan yang akan diubah, kita bisa memasukkan 0 agar pointer “cost” yang dipilih. Dengan strategi dan urutan yang tepat, kita bisa melakukan read/write anywhere.
- Karena GOT read-only (Full RELRO) dan ada PIE/ASLR, address yang dapat dipilih untuk di-overwrite adalah slot yang berisi return address pada stack frame.
- Alamat pointer “cost” dapat diperoleh dengan menggunakan side channel analysis menggunakan binary search karena pengecekan apakah 32 bilangan sudah terurut atau belum juga ada out of bound (pointer “cost” juga ikut tercek).
- Dengan strategi dan perhitungan yang tepat, libc address bisa di-leak untuk kemudian dilakukan ROP return to libc.

Lab Virus (Pwn, 957, *Solved by 6 teams*)

- Binary menggunakan custom memory allocator (heap-based) yang bisa mengalokasikan data sebesar 128 bytes, 192 bytes, dan 256 bytes. Struktur chunk adalah 8 bytes berisi size, 8 bytes berisi pointer alamat next chunk, dan 128/192/256 bytes berisi data.
- Ada format string vulnerability pada saat program menulis isi suatu slot kode genetik virus. Hal ini bisa dimanfaatkan untuk leak stack address dan libc address.
- Ada heap buffer overflow pada saat penulisan slot genetik virus. Hal ini dapat dimanfaatkan untuk meng-corrupt struktur chunk.
- Dengan strategi yang tepat (sama seperti eksploitasi heap pada glibc malloc), return address pada stack frame dapat di-overwrite untuk melakukan ret2libc.

Airport System (Pwn, 993, *Solved by 3 teams*)

- Use After Free pada:
  - Melihat airport connection. Untuk tiap koneksi dari suatu airport, program tidak memeriksa apakah airport yang terkoneksi telah lockdown (sudah di-free) atau belum. Dapat digunakan untuk leak.
  - Mengganti nama airport. Tidak dicek juga apakah airport sudah lockdown (sudah di-free).
- Heap overflow pada penggantian nama airport karena nama dicopy berdasarkan panjang nama airport baru (bisa lebih dari nama airport lama).
- Dengan kombinasi UAF dan heap overflow yang tepat, alur eksekusi program dapat dikontrol menuju ret2libc (misal: overwrite free_hook).

Toko Masker 4 (Web, 495, *Solved by 18 teams*)

- Chosen plaintext attack.
- Isi element pada array selectedItems dapat ditambahkan dengan field lain selain pk, price, dan quantity sehingga akan ikut terenkripsi.
- Jika menggunakan string yang panjang (misal, aaaaaaaaaaaaaaaaa…) akan terlihat repetitive blocks 16 bytes. Dapat diasumsikan enkripsi adalah ECB.
- Correct encrypted bytes dapat dikonstruksi hingga sedemikian sehingga price dari barang dengan indeks 3 (masker N99) adalah 0.
- Tips: Karena pemrosesan menggunakan int(num), maka baik indeks ataupun price dapat menggunakan padding 0 (misal, 000000000000000 atau 000000000003).

WHO (Web, 369, *Solved by 20 teams*)

- Ada banyak zero day IDOR pada plugin https://wordpress.org/plugins/rsvpmaker/
- Salah satu yang dapat digunakan:
    - Dapatkan email dengan wp-json/wp/v2/rsvpmaker/id
    - Leak guest name dengan https://who.web.cyber.jawara.systems/rsvpmaker/limited-covid-19-emergency-meeting/?e=who@cyberjawara.id&rsvp=1

Note:

Problem setter tidak mengantisipasi bahwa daftar guest bisa berantakan karena IDOR.

WHO 2 (Web, 1000, *Solved by 1 team*)

- Ada banyak zero day SQL injection pada plugin https://wordpress.org/plugins/rsvpmaker/
- Ada Blind SQL injection pada WP Rest API /rsvpmaker/v1/stripesuccess/ dan juga pada saat save RSVP.
- Ada multi-stage SQL injection pada save RSP yang dapat digunakan untuk leak database secara langsung tanpa blind.
- Salah satu payload, gunakan ini pada first name: testblablabla' union select concat("1 or 1=1 or 'a'='", concat((select name from wp_secretguest), """)) # . Flag akan muncul di Cookie. Nama kolom dari wp_secretguest dapat didapatkan terlebih dahulu dari information_schema.
- Lainnya: Ada zero day lain juga semacam unsafe deserialization!

COVID-19 Statistics

(Web, 789, *Solved by 12 teams*)

- Velocity server side template injection pada halaman not found
- Ada filter sehingga payload yang beredar di internet tidak ada yang bekerja
- Bisa baca dokumentasi Velocity
- Salah satu solusi:
    - /%5D%5D%23%24%7Bclass.inspect('java.lang.Runtime').type.getRuntime().exec('sleep%205').waitFor()%7D

Tic Tac Toe (Web, 1000, *Solved by 1 team*)

- DOM XSS dengan prototype pollution pada kode yang ada di https://tictactoe.web.cyber.jawara.systems/static/query.js . Kode ini adalah modifikasi dari jquery-deparam yang juga vulnerable (belum di-patch).
- Prototype pollution pada “cur = cur[key] = curVal;”
- Gunakan prototype pollution gadget pada jquery >= 3.0. Referensi: https://github.com/BlackFan/client-side-prototype-pollution
- Contoh XSS:
    - /?turn=O&constructor[prototype][url][]=data:,alert(1);//&constructor[prototype][dataType]=script

CJ Travel (Web, 999, *Solved by 2 teams*)

- Second order SQL injection pada email booking
- PDF generator untuk membuat e-ticket dapat di-abuse untuk SSRF. Referensi: *OWNING THE CLOUT THROUGH SSRF AND PDF GENERATORS* by Nahamsec. HTML to PDF bisa dieksploit untuk SSRF menggunakan <link> tag.
    - <link rel=attachment href="file:///etc/passwd">
- Memasukkan HTML bisa dengan union SQL injection.
- Hasil SSRF ada di attachment PDF dan bisa di-parse (misal, dengan PyPDF2).
- SSRF ke Google API Metadata untuk mendapatkan token. Ada 2 cara:
    - Menggunakan legacy Metadata API (tanpa perlu header)
    - Apabila akses ke legacy API dimatikan dan perlu header untuk API terbaru, dapat menggunakan CRLF injection pada service pdf generator (leak dulu source-nya).
- Gunakan token untuk mengakses Google Cloud Storage yang berisi flag