



Test & Security

MM4220 게임서버 프로그래밍

정내훈

내용

- Test
- Security

Test

- Test의 필요성
 - Bug없는 프로그램을 만들기 위해서
- MMORPG 게임 고유의 Test
 - 부하테스트 (Stress Test)

Test

- Stress Test

- 많은 인원이 동시에 접속하여 Test
- 적은 인원 Test시에는 보이지 않았던 bug들을 발견할 수 있다.
- 서버 프로그램의 bottle-neck을 발견할 수 있다.
 - DB
 - Network Overhead
 - Memory Allocation Overhead
 - NPC AI Overhead

Test

- Stress Test의 문제
 - 많은 인원의 동원이 쉽지 않다.
 - 동접 5000 서버의 테스트???
 - 원하는 시간의 테스트가 힘들다
 - 통제가 힘들다
 - 원하는 시나리오
 - 해결책
 - 자동 테스트 프로그램

Test

- 자동 테스트 프로그램
 - Dummy Client라고도 불림
 - No Visual (부하를 줄이기 위해)
 - 하나의 프로그램에서 여러 개의 character로
동시 접속
 - 보통 수백 개의 접속
 - ID를 미리 만들어 놓아야 함
 - 내부 AI를 이용한 테스트
 - 이동
 - 전투

Test

- 자동 테스트 프로그램의 활용
 - PC 여러 대를 사용해 테스트
 - 500 접속 Dummy Client * 10대 PC => 5000동접
- 자동 테스트 프로그램의 한계
 - AI의 한계
 - 복잡한 작업을 수행하기 힘들다
 - 예) 퀘스트, 상거래
 - 실제 인간의 행동과는 차이가 날 수 밖에 없다
 - 실제 서비스에서 나타날 모든 버그를 잡지는 못한다

Test

- 자동 테스트 프로그램 작성시 고려 사항
 - 캐릭터 생성도 자동으로 하게 해 놓으면 편하다.
 - 지형을 인식해서 빈번한 잘못된 이동을 하지 않도록 하는 것이 좋다.
 - 테스트 캐릭터의 공간적 분포를 고려해야 한다
 - Teleport를 통한 분산을 위해 관리자 레벨의 ID발급 필요
 - 서버프로그램의 업그레이드 시 같이 업그레이드 해주어야 한다.
 - 아니면 필요할 때 쓰기가 힘들어진다.

Test

- 자동 테스트 프로그램의 구조
 - IOCP를 사용한 다중 접속
 - FSM을 사용한 접속 및 Test
 - Login id 별로 state가 있고, 서버에서 오는 packet의 종류에 따라 state가 변화 하면서 action을 취한다.
 - 대부분의 서버 packet 무시
 - Status 변경, 다른 캐릭터 이동
 - 간단한 Graphic으로 화면상에 전체 test character들의 분포와 state를 볼 수 있게 한다.

일정

- 12월 14일 금요일 텀프로젝트 발표
- 12월 17일 오전 11시 30분 기말고사

실습

- 더미 클라이언트 소스
 - 과 자료실에서 다운로드
 - 프로토콜을 자신의 서버에 맞출 것

실습

- TIP

- Protocol 정리 부터

- SC_POS과 SC_PUT_PLAYER만 사용
 - 그 이외의 패킷은 빠짐없이 **Passing**
 - 또는, 자신의 프로토콜에 맞추도록

- 서버의 최대 접속 가능 용량을 늘리도록

- 테스트 유저 수 보다 많아야 한다.

숙제 (#6)

- 게임 서버/클라이언트 프로그램 작성
 - 내용
 - 더미 클라이언트 확장
 - **HOT SPOT**이 아니라 동접 테스트가 가능하도록 더미 아바타를 맵에 골고루 퍼트린다.
 - 더미 클라이언트 실행 시 **HotSpot** 테스트를 할 것인지 동접 테스트를 할 것인지 선택 가능하도록 할 것
 - 더미 아바타를 컴퓨터가 버티는 최대치 까지 테스트해보고, **HotSpot**과 동접테스트 두가지 모드에서의 최대치를 비교해 볼것. <**REPORT제출!!**>
 - 목적
 - 더미 클라이언트 사용법 숙지
 - 대용량 동접 시 서버 에러 수정
 - 제약
 - Windows에서 Visual Studio로 작성 할 것
 - 제출
 - 제목에 “2019 게임서버 [화목] #6 학번 이름” 말머리를 달 것
 - 화목반 6월 13일 목요일 오후 1시
 - Zip으로 소스를 묶어서 e-mail로 제출
 - 소스만(sdf, obj, log, manifest 같은거 제외!)

내용

- Test
- Security

보안

- 보안
 - 위험이나 손실로부터 보호 받는 것
 - ‘안전’과의 차이 : 의도적인 공격으로 부터의 보호
- 보안의 종류
 - 컴퓨팅 보안
 - 네트워크 보안

보안

- 보안의 종류
 - 네트워크 보안
 - 패킷 암호화
 - 키 배포
 - 컴퓨팅 보안
 - 거짓 환경
 - 프로그램 수정

보안

- 온라인 게임의 보안
 - 네트워크 보안
 - 정보 유출 방지
 - ID, Password
 - 유출 원인 : 바이러스, 키로거, 패킷 감청
 - 방지 대책 : 이상한 프로그램 설치 금지, 운영체제 업데이트, 유무선 공유기 보안
 - 방지 대책 : 패킷 암호화, 바이러스 백신 권장, 해킹 감지 프로그램 설치

보안

- 온라인 게임의 보안

- 네트워크 보안

- 부정 접속 방지

- 서버 컴퓨터 해킹, DB 공격

- 방지 대책 : 서버컴퓨터 관리 철저, 디버깅용 백도어 삭제,
모든 텍스트 길이 관리 (`strcpy` -> `strncpy`), 특수 문자 대체 ('
' -> `%20`)

- 부정 접속 방지

- 개발자 컴퓨터 해킹

- 방지 대책 : 망 분리, USB 사용 금지

보안

- 온라인 게임의 보안
 - 컴퓨팅 보안
 - 클라이언트 수정
 - 내부 데이터 해킹
 - 예) 스타크래프트: 맵핵, 유 : 크리티칼 히트
 - Client/Server모델로 갈 수 밖에 없는 이유
 - 트로이 목마
 - 키 로거
 - 보안 프로그램 사용 필요 (n-protect)
 - Auto 프로그램
 - 가짜 클라이언트

보안

- 컨텐츠 보안
 - 클라이언트 수정
 - 스피드 핵
 - 메모리 변경
 - Miss/HIT/Critical 변경
 - 맵핵
 - 벽너머의 상대방 보이기
 - 투명/은신 무력화

보안

- Auto 프로그램
 - 자동 사냥/채집/레벨업/팩션노가다/스킬노가다 인공지능 프로그램
 - 클라이언트를 대체
 - ID/PASSWORD를 등록해 놓으면 24시간 앵벌이
 - 문제점
 - 경제 질서 왜곡/플레이 방해 -> 게임의 재미 저하 -> 사용자 떠남 -> 현실 감소 -> Auto 떠남 -> 게임 망함
 - 게임회사에서 Auto단속하는 것은 진심임.
 - 착해서 단속하는 것이 아니라 게임이 망하기 때문.
 - 해결방법
 - 클라이언트 검증
 - 특수 패킷, 클라이언트 내부 메모리 검사, Delayed 처벌
 - 사용자 검증
 - 이미지 글자 읽기, 간단한 산수 문제, GM 대화

보안

- Auto 프로그램

- 해결 OK?

- 클라이언트 검사

- 이틀만에 뚫림

- 10만 잉여 중국 해커를 당할 수 가 없음

- Auto 프로그램의 온라인화

- 자동 업데이트, 실제 클라이언트와의 온라인 연동

- 사용자 검사

- 인공지능의 완성 : 오토인가 사람인가 구분 불가능

- 조선족 알바 : 실시간 화면 전송, GM과 잡담해 줌

- 한국인 확인 질문????

보안

- Auto 프로그램

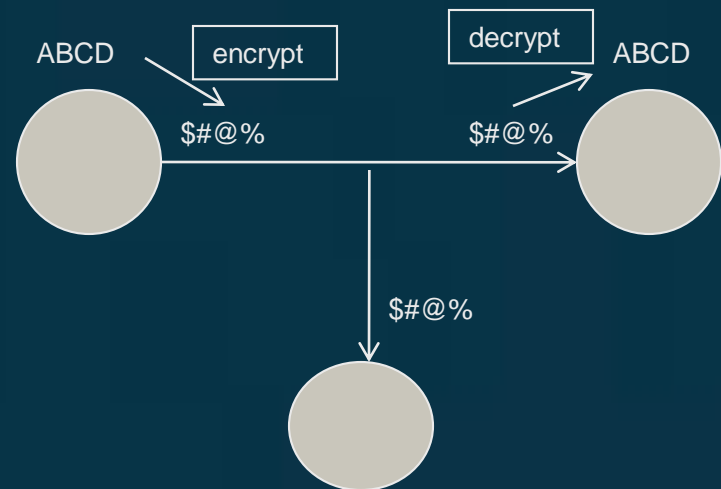
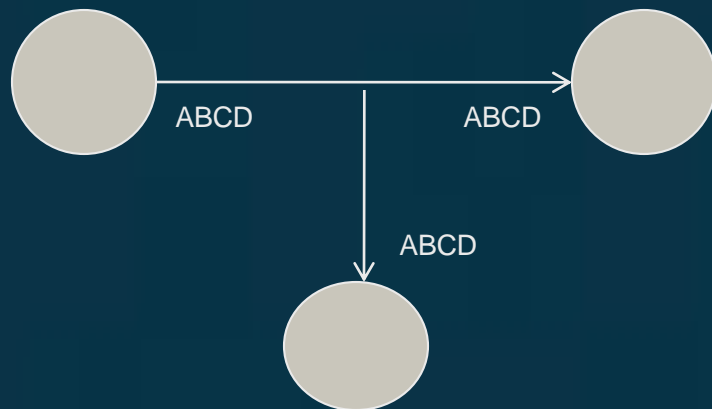
- 그래서?

- 계속 되는 업데이트로 auto 프로그램의 제작 원가를 높임
 - 미약한 몸부림
 - 중국 IP차단
 - 사용시간 제한
 - 사냥터 노후화
 - ...
 - 게임성 희생 : 모든 아이템 거래 불가 (ex:검은사막)
 - Auto를 게임에 포함 : 대부분의 모바일 MMO

보안

- 네트워크 보안

- 목적 : 도청으로부터 데이터 보호, 발신 수신자 확인
- 해결책 : 암호화



보안

- 암호화
 - 대칭 키 알고리즘
 - 블록 암호화
 - DES (Data Encryption Standard)
 - AES(Advanced Encryption Standard)
 - Stream 암호화
 - RC4(used in SSL, WEP)
 - 공용 키 알고리즘
 - Diffie-Helman key exchange
 - RSA(Rivest-Shamir-Adleman)

보안

- 대칭 키 알고리즘의 문제
 - 키의 전달 필요
 - 고정된 키
 - 프로그램안에 내장
 - 약한 보안성
 - 시간 기반의 키
 - 시간에 의해 변하는 키값
 - 시간 동기화 필요
 - 동적인 키 변환
 - 키의 암호화
 - 메타키?

보안

- 공용키 알고리즘
 - 각자 2개의 key를 쌍으로 갖는다
 - Private key : 자신만 안다.
 - Public key : 모든 사람에게 다 알려준다.
 - A가 B에게 메시지를 보낼때
 - A는 B의 public key로 암호화 해서 보낸다
 - B는 B의 private key로 암호를 해독한다.
 - A가 B에게 자신이 보냈음을 증명할때
 - A는 A의 private key로 암호화 해서 보낸다
 - B는 A의 public key로 암호를 해독한다.

보안

- 공용키 알고리즘
 - Key관리 문제 해결
 - DRM(Digital Right Management)의 기반

보안

- 게임 보안 정리
 - 힘들게 서버를 만들어야 하는 이유?
 - 보안 때문
 - Diablo1, 맵핵, Fantasy Star Online의 교훈
 - Client/서버에서 왜 보안이 필요한가?
 - Reverse Engineering
 - 프리 서버
 - Auto Program
 - 막을 수 없다.
 - 널리 퍼진 Auto 프로그램 차단 : update, n-protect
 - Auto의 제작을 어렵게 하는 것으로도 효과있음

기말일정

- 화목반
 - 18일 텀프로젝트 발표
 - 20일 기말고사
- 수목반
 - 19일 텀프로젝트 발표
 - 20일 기말고사

ToDo

- 팀 프로젝트
 - 6월 18일 발표
 - 6월 18일 오후 1시까지 제출
 - 팀 프로젝트 보고서, PPT
 - 프로그램 실행 방법
 - 프로그램 구조 설명
 - Stress Test를 통한 동접 허용량
 - 소스 코드, DB 백업
 - 주의 : 제시한 프로토콜 준수
 - Stress Test Server와의 호환성을 위해 준수해야 한다.
 - 프로토콜이 많이 변경되어서 Stress Test Server와의 호환성이 없을 시에는 수정된 Stress Test Server 같이 제출

질문

- ???