Part II. 플레이 검증

목표

- No Hack! No Abusing!
- 전제
  - 메모리 보호는 불가능하다
    - 증명?
  - 그래서, 항상 패킷 수정이 가능하다
- 게임의 룰 파괴
  - 이속, 사거리, 쿨타임 무시
  - 무적
  - 아이템 복사
  - 플레이 속도 증가 (스피드 핵)
  - 사례는 매우 많음

가이드

- 아크로드2의 보안 컨설팅 자료

게임 컨텐츠	**스킬 컨텐츠와 관련된 어뷰징 요소가 있는지 점검하는 항목**		
	GCS001	배우지 않은 스킬을 사용할 수 있는 어뷰징이 가능합니까?	미취약
	GCS002	스킬을 사용할 수 없는 상황 또는 대상에게 강제로 사용할 수 있는 어뷰징이 가능합니까?	취약
	GCS003	스킬의 기능을 조작하는 어뷰징이 가능합니까?	취약
	채팅 컨텐츠와 관련된 어뷰징 요소가 있는지 점검하는 항목		
	GCC001	권한이 없는 채팅 메시지를 사용하는 어뷰징이 가능합니까?	취약
	GCC002	원하는 형태로 채팅 메시지를 다른 유저에게 보여줄 수 있도록 어뷰징이 가능합니까?	취약
	대/소규모 전장 컨텐츠와 관련된 어뷰징 요소가 있는지 점검하는 항목		
	GCW001	전장 승리조건을 조작 가능한 어뷰징이 가능합니까?	취약
	부활 컨텐츠와 관련된 어뷰징 요소가 있는지 점검하는 항목		
	GCR001	부활 아이템 없이 부활 가능한 어뷰징이 가능합니까?	취약
	채집 및 제작 컨텐츠와 관련된 어뷰징 요소가 있는지 점검하는 항목		
	GCM001	채집 및 제작과 관련된 어뷰징이 가능합니까?	취약
	진영(에임하이, 데몰리션) 컨텐츠와 관련된 어뷰징 요소가 있는지 점검하는 항목		
	GCE001	채집 및 제작과 관련된 어뷰징이 가능합니까?	미취약
	보안(2차 비밀번호) 컨텐츠와 관련된 어뷰징 요소가 있는지 점검하는 항목		
	GCB001	2차 비밀번호 설정 관련된 어뷰징이 가능합니까?	미취약
	GCB002	2차 비밀번호의 우회가 가능합니까?	취약
	포인트(스탯 업) 컨텐츠와 관련된 어뷰징 요소가 있는지 점검하는 항목		
	GCP001	포인트 습득 및 사용과 관련된 어뷰징이 가능합니까?	미취약
	사냥 컨텐츠와 관련된 어뷰징 요소가 있는지 점검하는 항목		
	GCH001	사거리에 상관없이 공격 가능한 어뷰징이 가능합니까?	취약
	GCH002	탈것을 탄 상태에서 공격 가능한 어뷰징이 가능합니까?	취약
	GCH003	공격 불가능한 방향에서 공격 가능한 어뷰징이 가능합니까?	취약
	GCH004	여러 대상을 동시에 공격 가능한 어뷰징이 가능합니까?	취약
	GCH005	공격 불가능한 높이에서 공격 가능한 어뷰징이 가능합니까?	취약

스킬 조작

- 사거리 / 시전 위치 조작
  - 가능한가?
  - 왜 불가능한가?
- 뮤2의 경우 Projectile 처리 이슈
  - 어떻게 할 것인가?

스피드 핵

취약점 내용	매우 빠른 속도로 게임 플레이 가능(이동속도 및 공격속도 향상)	취약점 ID	VID004

상세내용

time 관련 Win32 API Hooking 등의 방법으로 게임 내 속도를 조작한 결과
비 정상적인 속도로 플레이 할 수 있는 문제점 발견됨.

Target API : timeGetTime, GetTickCount, QueryPerformanceCounter, QueryPerformanceFrequency 등...

추가 데이터

Address	Bytes	Opcode		Comment
WINMM.timeGetTime				
WINMM.timeGetTime	83 3D D48F8A...	**cmp**	dword ptr [WINMM.dll+28FD4],[00000000]	
WINMM.timeGetTime+7E9	01EC25E7	**jmp**	[illegible]	
WINMM.timeGetTime+CCC		**int 3**		
WINMM.timeGetTime+CE8	CEFFFFFF	**call**	WINMM.dll+26C0	
WINMM.timeGetTime+12B	05 D88F8A...	**sub**	eax,[WINMM.dll+28FD8]	[B7214B4B]
WINMM.timeGetTime+16A	00	**push**	00	0
WINMM.timeGetTime+11B	15 DC8F8A...	**sbb**	edx,[WINMM.dll+28FDC]	[0000001E]
WINMM.timeGetTime+268	10270000	**push**	00002710	10000
WINMM.timeGetTime+252		**push**	edx	
WINMM.timeGetTime+250		**push**	eax	
WINMM.timeGetTime+2E8	0C000000	**call**	WINMM.timeGetTime+38	->->ntdll.alldiv
WINMM.timeGetTime+203	05 E08F8A...	**add**	eax,[WINMM.dll+28FE0]	
WINMM.timeGetTime+3C3		**ret**		

게임 속도 조작을 위해 timeGetTime() 함수의 흐름을 공격자의 DLL 모듈로 변경한 화면 (mid function intercept 기법)

스피드 핵

- 클라이언트 동작 전체가 의존
- 서버에서 목표 지점으로 MoveTo 이동
  - 이것만으로도 많은 문제들이 완화됨
- 시간 동기화
  - NTP 알고리즘의 변형
    - https://en.wikipedia.org/wiki/Network_Time_Protocol
  - 서버 시간만 사용하여 지연을 측정
    - 이를 MoveTo에 반영 (속도를 올림)
  - 클라이언트 시간을 기록하여 전송
    - 클라 시간, 서버 시간의 차이를 보면 스핵을 어느 정도 판정 가능
    - 로그로 남김

아이템 트래킹

- 아이템의 생성, 획득, 사용, 소멸, 이동 (거래), 변경 (강화) 추적
- 고유한 생성 아이디 부여
  - Server / Date / Sequence (Rotating)
    - 64비트 단일 필드
  - DB 필드로 갖고 있음
  - 로그에 항상 포함
  - 서버에서만 갖고 있음

퀘스트

- 완료 조건의 강제 달성
  - 가능할까?
  - 단위 행위의 검증
- 클라이언트 메시지의 형태
  - 발생 가능한 클라이언트 상태
  - 발생 가능한 서버 상태
  - 양 측 사이의 검증

검증

- 검증이란 무엇인가?
- 검증이 어떻게 가능한가?
- 동시 시뮬레이션
  - 양쪽에서 일어나는 일의 검증
- 단일 시뮬레이션
  - 클라이언트는 더미
- 과거의 재현
  - Replay

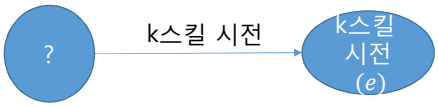
검증 – 함수 호출

- void increaseHealth(float value)
  - this.health += value;
- void castSkill(int skillId)
  - …
- 검증?
  - 어떤 모델이 필요

검증 – 상태 모델

게임의 상태 공간은 매우 큼

따라서, 이벤트 e와 연관된 상태만 고려
클라 요청 이벤트 e는 서버 처리 후 이벤트 F(e) 가 되어야 함



여기는 어떤 상태?

스킬 시전 중

스킬 시전 중 이전 상태의 조건
- k 스킬의 쿨 타임
- 시전자의 상태
  - 상태 이상, MP
- 위의 조건을 통과한 F(e)가 서버 이벤트

검증 – 상태 모델

- F(e)를 개념으로 만들고 공식화 하는 건 어떨까?
  - Precondition들 -> Pre(condition)
  - 상태 전환을 위한 동작 (결과) -> Eff(ect)
- 구현
  - <Pre, Eff>의 Sequence
    - 현재 함수 단위 체크와 실행 방식
  - <Pre, Pre, … >, <Eff, Eff, … >
    - 모든 검증을 마친 후 진행 방식
  - 유연하게 사용
- Precondition은 Constraint로 볼 수 있다.
  - 클라이언트가 생성 가능한 메세지의 제약으로 볼 수 있다.

검증 – 바람직한 이벤트 처리 특성

- 항상 Precondition 체크가 가능하도록 한다
  - 이게 최상의 목표지만 항상 가능하지는 않다.
  - 서버 시뮬레이션의 비용이 큰 경우
    - Projectile
      - 실시간 충돌 처리에 기반
- 이벤트가 직접 서버 상태를 변경하지 않는다
  - 값 지정
    - HP를 바로 복원 -> 물약 사용
      - 검증 : 물약 보유? 물약 사용 쿨타임?
  - 서버 시뮬레이션에 대한 입력 이벤트로 보는 게 최상
    - 위치 지정 -> 속도와 방향 전환

검증 – 다중 이벤트

- 하나의 작업을 완료하기 위해 여러 외부 이벤트가 연관된 경우
  - 예) 거래
  - 예) 아이템 분해
- 이벤트 누락 / 중복 / 순서 역전에 대해 고려
  - 예) 거래 수락 후 강제 종료 (빠르게 진행할 경우 어떻게 되는가?)
  - 예) 동일 아이템 분해 요청이 짧은 시간에 여러 번 올 경우
- 중간에 다른 이벤트 관여하는 것 검증
  - 예) 거래 대상 아이템을 판매하는 것

코드 리뷰

- Pre 체크가 불가능한 이벤트들
- Pre 체크가 없는 이벤트들
- Pre, Eff 순서가 잘못된 경우들
- 이벤트가 서버 상태 값을 직접 바꾸는 경우
- 서버 시뮬레이션이 없는 경우
- 코드 커버리지
  - 다양한 제어 경로의 검증
  - 타잎과 함수
    - 구조화된 클래스보다 작은 개념적인 타잎과 함수로 생각해 볼 수 있다

검증 - 프로젝타일

- 발사체
  - 클라에서 Arrive (Hit) 전송
  - 서버에서 발사체 아이디 검증 후 대상에 대한 피격 처리
- Arrive의 조작
  - 타겟의 변경
  - 안 맞거나 맞게 할 수 있음
- 어디까지 검증 가능한가?

참고 자료

- http://www.markrtuttle.com/data/papers/lt89-cwi.pdf
  - IO Automata (1988)