Pattra Audcharevorakul
April 8, 2014
COMP23: IEEE Security & Privacy – Securing Online Games

1. I believe the main vulnerabilities in online games are non-technical. While companies can try their best to release patch after patch of security updates, the most problematic attacks can't be patched against—e.g., social engineering schemes and phishing scams. Against those kinds of attacks, you just have to educate your playerbase and hope that they listen.

Crazy

2. I think the most interesting exploit was the Corrupted Blood incident in WoW. Firstly: how wildly ironic that the way that the disease spread was via hunter pets. In addition, upon some further research, I learned about an instance in which a boss (Lord Kazzak) was kited into Stormwind, wreaking havoc to the point where no player or NPC in the game could defeat it. In the end, Blizzard employees just had to delete him (which is admittedly somewhat anti-climactic).

Bots (and bot detection) is another really interesting topic. I hadn't thought about the fact that one would have to teach the bot certain behaviors and facts. In addition, figuring out server-side mechanisms to recognize bots is really clever. Exploiting the fact that, even if there's variation in the path, bots really only do the same thing over and over is a great idea for detection.

3. To start, I thought the article about server-side bot detection presented a really clever concept for catching bots in action. At best, truly dedicated hackers can think of workarounds for mechanisms such as the Warden, and at worst can send back malicious or misleading data to servers. Thinking of more server-side detection mechanisms would definitely be a step-up from current practice, especially because not only is local software not necessarily effective, but they also pose quite a few privacy issues (e.g.: did Blizzard essentially make spyware legal by simply putting a few sentences in an EULA?).

hard problem.

4. Of course. No one likes to envision this kind of stuff, but it's bound to happen. Our JSON save files and character stat are totally exposed, and if you didn't even really want to "play," people could easily write a bot that'd beat our game. All you'd need are some basic path-finding algorithms and a familiarity with the game mechanics. It's somewhat "overly optimistic" to think that someone could produce an unhackable app.

Nice

5. Mobile is definitely a problem. I don't necessarily know how people would go about doing it, but there's evidence of hacking issues in mobile everywhere. I remember when Flappy Bird was all the rage, there were Game Center top scores maxing out at 999,999,999. While we could potentially imagine the existence of 2-5 choice people who really sat there with the mental acuity and stamina to flap through almost a billion pipes, it's probably more likely that they just made a bot to beat the game for them. Now imagine how bad things could get if people could somehow exploit in-app purchases and the like.

→ I need to talk about this

Harry O'Sullivan
Comp23

One

After reading the presented articles, I think it's pretty clear that most of the cheating that occurs in the gaming industry can be attributed to poor technical security. Knowing how to program is not specific to the geek community anymore; the general population of today knows how basic programming works. Consequently, game developers ought to heighten the security placed on source codes and be weary that many gamers, as it has been proven, love to cheat. Strides have been made to thwart cheaters, for instance creating barriers to game completion or penalizing player performance. But cheating is obviously still rampant and will only decrease with stronger and thicker code.

Two

A number of vulnerabilities were described in the provided papers, and my favourite one has to be the *Exploiting a bug or loophole* example in the paper entitled "An Investigation of Cheating in Online Games." In this cheating method, cheaters exploit a bug in a game's program or the game's design without having to modify any of the source code. Catching these bugs will naturally give knowledgeable players a huge advantage. The article provides the example of Lucasfilm's *Habitat*, which had a bug that allowed people in the game to sell virtual items to a pawnshop at a higher price than they paid to get them from a vending machine. These players became instant virtual millionaires. While this cheating method is not the most ingenious or glorious, this one has to go down as my favourite because it is totally avoidable. The reason why such kind of a cheating method exists is purely down to the carelessness of the game developers. If the game developers want less cheating to occur, then they must try their utmost best in writing code that has minimal bugs.

Three

One of the ways cheaters cheat is by reverse engineering. Online games run entirely on the end user's machine, which can enable hackers to have complete control and scrutiny over game execution at run time and at rest to unravel source code. For instance, hackers can easily find and exploit cheat commands unless they are hidden properly. These hackers can also find out how the client receives game data from the server, allowing for the development of cheat codes. Thus, a solution to this would be to make it so that client code will always remain confidential. By doing so, they cannot make unauthorized changes to program functionality in order to exploit performance variables or insert malware. Furthermore, a cryptographic key can be used to prevent the discovery of public and private keys and can assist in securing communication between the client and the server.

good luck w/ that

True!

Four

I can indeed envision these issues to arise in our 2D game and final project. Cheating mostly occurs, however, with games that are popular and those that have very structured in-game economies. Our particular 2D game was very simple in that regard; no items can be purchased or sold and there are very few things a player can accomplish. That being said, our code is probably not secure, and anyone, after downloading the game, could tamper with it. On the other hand, our final project will have a multiplayer features and will thus involve communication a server. This is a great chance for us to see how well we can write secure code for such kind of communication and to make sure we find solutions to any bugs or insecurities within the game.

Five

These issues do indeed pertain to games geared for the mobile phone. It can easily be observed by doing a simple Google search that there are a myriad of websites that provide cheat codes for popular mobile games. Just because the gaming platform is not a computer does not make it impervious to hackers.

excellent obsenation

Connor McCann
Assignment 4
Comp 23

IEEE Security and Privacy Securing Online Games

1. I think the primary (and most significant) reason for vulnerabilities in online games is technical. As the scope, complexity, and depth of online games increases, so too does the sheer impossibility of securing all elements of the game. Good examples of this are the multimedia exploits in Second Life, where features that added a unique depth to the game opened unintended holes in its security, and Age of Conan and Anarchy Online, where adding a depth to chat interactions allowed scripting exploits. While insecurity in games that results in cheating the game economy or makes the game easier are still important considerations, their real-world consequences are less serious, in my opinion.

Crazy

2. My favorite exploit described in the papers is briefly mentioned in "Reducing the Attack Surface in Massively Multiplayer Online Role-Playing Games": the Corrupted Blood incident. The exploit allowed WoW players to be affected by a disease debuff (cast by a boss in a dungeon) that would spread to any adjacent characters, and then teleport to the most populous areas of the game world, thus creating a plague slaughtering thousands of characters for hours on end. This is my favorite because not only was I playing WoW when this exploit was discovered, but I directly saw the carnage it created. Although it was a headache for thousands of people, it was a fascinating experience that I found fairly comical, so I wasn't bothered in my gaming experience. Further, it spawned academic sociological papers that used the incident as a "Disease model", and as a "lesson for pandemics", which is pretty neat. (http://news.bbc.co.uk/2/hi/ health/6951918.stm ; http://uk.reuters.com/article/ 2009/04/27/us-flu-virtual-idUKTRE53Q4HI20090427).

Commonly done there is value to this but debatable

3. As games grow more and more complex, I do not think there is a possible singular solution to vulnerabilities in online games. One way to help mitigate the damage is to hire security experts during the development and maintenance of the game, but ultimately, in a game with thousands or tens of thousands of features it is near impossible to make sure every single one is 100% secure. The best you can do, I think, is to respond quickly and adaptively, and learn lessons from games that have come before.
4. No, as the scope of our project is small, and its audience will be quite limited. Although I'm sure our programming will leave vulnerabilities in the code, the potential monetary benefit of taking advantage of them would be next to nothing compared with popular games or apps (unless our game explodes in popularity a la Flappy Bird... fingers crossed).
5. Yes, absolutely. People store an incredible amount of data on their phones (credit card numbers, phone numbers, social security numbers, banking information, etc.), and the market for mobile apps is so unbelievably vast that it would be impossible to thoroughly examine all of them to minimize security vulnerabilities. Furthermore, it only takes one explosively popular game to be patched to a version with an exploit for potentially hundreds of thousands of phones to become compromised.

Thomas Schaffner

✓ 1: It seems to me that many problems relating to game security (or lack thereof) revolve around technical issues. Most of the issues discussed in these papers, at least, talk about software failures that render computers vulnerable to attack. These vulnerabilities seem to increase with increased complexity of a game's architecture, especially when it comes to networks or third party software.

Networked games (MMOs, etc.) provide an avenue for attack from one computer to another. When the game is poorly designed, too much information becomes available on client computers (as opposed to servers). This permits hackers to access or send information that they should not be able to access or send. The buffer overflow and change of directory attacks discussed in these articles would not be possible if client computers were adequately limited in their ability to communicate.

Likewise, when dealing with third party software like Quicktime, it becomes harder to find and safeguard against vulnerabilities. A security breach in one piece of software (again, like Quicktime) will lead to a security breach in the entire game. It is impossible for the game itself to control the functionality/safety of Quicktime, as it is third party software.

Yes, very simple

2: My favorite was the change of directory exploit. It is exceedingly simple. Simply because the game has too much access to a host computer's files, any player can access another player's files. There is no complex algorithm or software. The game merely hands access over to complete strangers.

3: Reducing use of third party software like Quicktime allows developers to fully control their game's security. Similarly, centralizing software onto servers, while more costly, also reduces the ability of individual users of the software to modify code.

true

✓ 4: I highly doubt many issues would arise with the 2D game. We have very limited input, and no networking or interaction with other systems. It is more possible on a mobile game. However, between sandboxing policies and the fact that my group does not plan on incorporating networking into our mobile game, I find it unlikely that security will be an issue for us either.

✓ 5: In general, these issues certainly apply to mobile games. While sandboxing of apps on mobile devices may cut down on the ability of one vulnerable app to compromise a full device, it's still possible that a single app might store valuable data (bank accounts, personal info, etc.). Therefore, I believe networked apps can still be quite vulnerable to attacks, at least if they do not keep security in mind. Additionally, many apps deal with servers or databases separate from individual devices. These too could become vulnerable to attack.

Param Chopra
Assignment 4

1. I think the main reason for vulnerabilities in a game is the human condition. If one were to make a game where the player has little to no freedom, then that game wouldn't be fun for very long. Profit is to be had in making games that allow players to create their experiences, but at the same time, people are greedy, and if given a chance to abuse the system, many people will do just that. Game developers can do battle with hackers/cheaters, but they can never fix the cheating problem fully while they greedy people continue to exist; there's always a way to bypass the system, and as we've seen, people have been getting better and better at cheating in increasingly advanced and secure games.

2. The exploit that interested me the most is the fact that a cheater could write code in Python and inject it into the interpreter at any time during a game. Not only could the cheater abuse the built in functions that control an avatar/game settings, but the cheater could also exploit dynamic recompilation to quickly create valuable items and assets that would normally take much longer to create when playing the game without cheats. Another aspect of this hack is a possibility to create a character or bot farm; this method defeats the purpose of even playing a MMORPG, and it fascinates me that players would do this just to excel at something that could have no bearing on their real lives.

3. My solution has both technical and legal implications; the companies that write the driver code for these games have to focus on making as much of their code as private as possible. Too much freedom gives leeway to abuses of the system, and thus the only way to limit hackers is to make sure that the client code does not reveal any information that should otherwise be hidden. The fact that players can create bot farms and fake valuable assets by manipulating in game scripts really speaks to the low level of security that some games currently have. Secondly, I personally believe that if a player wishes to use his/her real money in order to succeed virtually, then they have to be aware of the risks they are taking when they choose that path. Of course, fraud is not acceptable either, thus we should try to increase the amount of involvement of the U.S. (or any world government) legal system when combatting cheating in games. If we treat the theft of real goods/money as a crime, then I don't see why we can't do the same for the virtual world. Losing virtual money is one thing, but losing real money has much more serious implications, and I believe that cheaters who work to steal real money should be punished as criminals.

good luck w/ that

education ?

4. Some of these issues could affect my 2D game project, as it was written in Python, a language that dynamically recompiles with an interpreter. But, since my 2D game project has no way for a player to give his/her personal information to the game, no "real" crime will happen. Sure, a player could somehow hack our python code and change the physics of the game in order to cheat, but to me, cheating runs rampant most in a multiplayer setting; people love to stab each other in the back. Cheating in a single player game isn't nearly as fun.

5. These issues definitely apply for online games, because many of these games have multiplayer components, as well as ways for the player to spend real money in order to succees in the game. As long as a player can connect his/her personal information to a game, someone has the ability to hack the game using one of the methods described in the articles provided in order to scam a player for profit.

Max Smi..

Assignment 4: *IEEE Security & Privacy Securing Online Games* Readings

1. Technical – because even the non-technical issues are really technical ones. For example, cheating is sometimes a non-technical problem that is solvable by the technical problem of cheat detection and cheat punishment in a game system (or having an implementation that doesn't allow exploits of time and information in the first place). The same for bot detection, exploit detection etc. The majority of problems described in the readings seemed to be fully technical ones. Client-to-server to client is all about content filtering and validation, introducing platforms, plugins, or user-specifiable scripts, creates holes for technical problems, etc.

2. Bots! Bots are an extremely diverse and interesting exploit, both in detection and creation. To create a bot is very interesting. I made my simplest "bot" in elementary school when I taped the forward button down on my computer in Runescape so I could travel somewhere else and not be close to the computer. Bots take a lot of cleverness to create as a game's environment becomes more complex. They are an interesting interaction between inputs and feedback from the game, and this is sometimes impossible to avoid since a game must receive input from the user and must give something back to the user, otherwise we don't have a game. And even if the game does a good job of hiding the computational representation of what's going on (e.g. hash variables in memory, let the server decide if something is pathable), people can train their bots by feeding them the behaviors they should repeat. Bot detection is just as interesting, how can you determine that the actions of a user are not human? Funny enough one solution is to ask them! Obviously this isn't a consistent solution, but the problem with the more mechanistic ones, is their burden of computation. For example, if a bot uses a script that follows a set path, you can detect if a player has traversed the same exact path numerous times, a feat impossible for a normal human being, especially in a large 3d world. But a bot could simple enact variance in its script, changing the coordinates it moves by random amounts. And the bottom line is this detection is specific to games like WoW, in every game there can be a very different nature to bots, some not even using recognizable subroutines.

Hard problem

3. Well the "best" way to reduce vulnerabilities and exploits, would be to have all computations be handled outside of the clients domain – for example if WoW were to handle all animations, movements and control, processing, etc. on the server side and the client was glorified streaming video of the game. That way the user cannot manipulate any variables in the game beyond specified inputs and it makes it incredibly hard for them to exploit the game beyond faults in its design, rather than faults in its technical implementation. The less control users have over things, the less chance there is for there to be an exploit that is discoverable. Obviously this is an unreasonable solution for a lot of games, because it raises the computational burden for server-side several magnitudes, and not even the goliath that is WoW could break even with that kind of setup. It also makes games less fun. A lot of the exploits in these games are due to the providers giving players more avenues to enjoy their games – whether it be by letting users make their own macros, use their own media, or have access to information about the game on multiple platforms.

✓

4. Yes. Our 2D game is easily exploitable – as it depends on openly accessible and easily edited JSON files that determine character's stats and levels, and floor layouts. This not only allows the players to cheat the game by buffing their characters, but also allows them to spoil the game by seeing all of the enemy character sheets. It also allows to break the game for themselves by deleting some of these vital files or editing them into an unreadable state.

✓

5. Yes. The motivation behind hacking is just as prime. My uncle emailed me just last month asking me to hack candy crush for him on his iPhone. I didn't, but the point still stands that people are actively looking to exploit this format of games. And it is not as hard as it looks to exploit! Although smartphones often try to hide a lot of their implementation from the user, in the end they work just like a normal computer, with files, directories, and scripts. A good example of an exploit is SnapChat – although its up to discussion whether or not this is a game. SnapChat is about sending images to other users under the impression that these images are gone after a few seconds. The problem is – they are still files that exist on the device (albeit hidden)! If we were to port our 2D game to android as it stands, the bottom line is that those JSON files would still be created and all the same exploits would apply.

Jake Mingolla
April 8th, 2014
Comp 23 – Game Development

truth!

Assignment 4: IEEE Security & Privacy Securing Online Games

1. I think the main reason behind insecurities in games non-technical and has more to do with the attitude of gamers towards malicious exploits. One of the articles show "when it comes to handling email attachments, dealing with spam, disclosing personal information, and clicking on popup windows, but online gamers haven't yet had to learn analogous safety behavior for virtual worlds" (Bono, Caselden, Landau, Miller). Another example of this is the QuickTime exploit in SecondLife – the users were bombed through objects in the game they had assumed were safe. I believe that software as a platform for online games will always have inherent security vulnerabilities and it is up to the users to be come more prudent about their interaction with possible exploits.

very popular choice

2. My favorite vulnerability was the way in which the Pirates of the Caribbean game attempted to streamline the user's gameplay experience by making physics calculations on the user's computer rather than server side. However, this made the game's code incredibly easy to access using tools such as AntiFreeze and could be changed. The fact that the article even included screenshots of a player jumping higher than normal and walking across water showed the serious impact of having a game trust players to make their own calculations.

3. Give video game players more of an understanding of the risks behind a game. I feel as though when I play a game I think of it as being completely airtight and free of vulnerabilities. However, just like all pieces of software, bugs are inherent in the game's framework and it is only a matter of time before creative hackers find a way through the gaps. Just like how the end of Mitterhofer and Platzer's article on Bot Detection states that simple security measures can prevent botting in MMO games, players can be given rudimentary knowledge of what to avoid in order to reduce their risk of encountering malicious content. This assumes, just like the article states, there are no massive overhauls in the way exploits are carried out.

Your 2D game?

4. I do not see these issues of security and privacy being an important factor in our final iOS game. One of the biggest threats to mobile games is online connectivity and how the information is handled across a network. Like these articles showed, the biggest cause of exploitation was when user oversight came into contact with malicious code. Since there will be no advertisements or other extraneous objects the user can interact with, I believe we will cut out any ability for the user to come in contact with malicious third party software.

True!

5. While I do not feel our iOS game is an easy target for game exploits, I believe these issues apply for mobile games based on the growing trend of massively multiplayer mobile games. Games such as Candy Crush and Plants vs. Zombies provide the user with a mix of online play and DLC that can be purchased with real-life money. Because of the relatively high level languages used to program many (especially iOS) mobile games tied to real-life currency, "migration to dynamic languages carries with it unforeseen risks to intellectual property and, in many cases, makes it easier for malicious users to subvert the game"(Portnoy & Rizvi-Santiago).

Karan Singhal
Comp 23 – Assignment 4

1. After reading all the papers, what do you think is the main reason for vulnerabilities or insecurity in online games: technical or non-technical?
    - The biggest reason for vulnerabilities is, according to me, undoubtedly the thrill that comes from circumventing the rules of the game and the introduction of real currency into games. This coupled with the lack of legal recourse leads to an open battlefield for the "hackers".

Interesting

2. A number of vulnerabilities and exploits are described in the papers. Which is your favorite?
    - My favorite exploit from the papers is creating bots, because the repetitive tasks waste my time and don't give me the same pleasure as the other parts of the game. By creating a bot, I can leave the computer to level up my player and unlock the higher-end weapons and other items that make the game a blast.
3. What is your solution to reduce the vulnerabilities and exploits in online games (PC)?
    - While we can't be brazen and say that any solution will completely get rid of the problem, I believe a major part of the issue is that the code for the game is accessible from the host computer (see the Pirates of the Caribbean example). If the code was hosted on a secure server, and the developers used security practices that are in place for other software, we could limit the problem.
    - However, a better solution in my mind would be to redefine the laws surrounding hacking and exploiting games to give the hackers some consequences for their actions. Stealing is illegal, and by creating methods to track cheaters and punish them, we can limit the issue further.
4. Do you envision these issues to happen in your 2D game or final project?
    - My final project and our 2D group game project do not have virtual currency or any monetization process, so a lot of the incentive to hack is gone. However, I still do see people developing cheats for "infinite jumps" in World War Zoo or for "invincibility" in Wild Turkey.
5. Do you think these issues apply for mobile games?
    - I believe the same issues exist, but on a lesser scale (especially with iOS devices because Apple is so locked down). It is hard to exploit a game without any ability to code on the host device.
    - However, with Android games, since it is so open, there is a large opportunity for cheaters, and on both platforms the in-app purchase mechanisms introduce real money into the system, so there is more incentive for cheating as well.

Alex Goldschm

1. The greatest problem with online game security is technological. Although it takes human ingenuity and maliciousness to figure out how to hack into a game, it really just comes down to how the software was implemented. If there is a way to inject HTML or SQL via chat, or to alter the source code, that is a fault in implementing the program, and was simply discovered by the user.

2. My favorite exploit was the Pirates of the Caribbean hack. The other hacks involved exploiting the chat system, whether it was redirecting to other sites, or changing directories with addresses. This hack was done directly to the source code using machine level commands. It was also interesting to me because we've been coding in python, and this was a hack done to a python game.

3. My first solution would be to remove as much access to source on the client-side of things. This might be infeasable, because it would slow the game down immensely and might make some features impossible, because all of the stress would be on the game's server instead of passing it on to the user's machine. My second solution would be for companies to hire some sort of consultant hacker who would show companies where their vulnerabilitiers are and how they could better avoid them.

Plenty of this

4. Some of these hacks wouldn't be possible in our 2D games because they aren't developed enough. We have no online multiplayer mode, or a chat feature, so those pathways are obsolete. Because everyone playing had access to the source, and actually ran the game from the terminal, it would be very easy to go in and change the variables directly, and rerun it with those changed values to give them an unfair advantage. It would be hard, though, because the characters in our game draw from the same source for both teams, so changing one unit changes it for both players.

5. Absolutely. Not only are more and more mobile games incorporating global user interaction, they also rely more and more on in app purchases (gold, gems, powerups). If someone were able to cheat and could procure these items for free, it would give them an immensely unfair advantage, and if there were money at stake, or even something like a high score table, they would be taking full advantage of that system and ruin it for everyone.

Aromie Kim
COMP 23

April 4, 2014

Assignment #4

1. I believe that most of the insecurity in online games is technical in nature. Most major security leaks seem to have resulted from an inadequacy in the design of the games. Once found, it is exploited by players. Indeed, plenty of malicious events can follow due to non-technical methods (ex. a player's misplaced trust) but it seems that this scamming process can be deftly assisted by security bypasses, small bugs, and so forth that the malicious cheaters managed to unlock.

LoL

2. I think social engineering is one of my favorite forms of cheating (I never did it, I swear). I've seen some kinds of computer viruses that work this way. At times, it's clever, and it catches the player completely off guard. This type of cheating is an attack meant to tantalize and can often be difficult to diagnose as a cheating attempt. Its deadliness is well-noted; companies such as Blizzard had to step in and publish guidelines in an effort to avoid this problem. However, it's probable that such guidelines can't eliminate the issue entirely, especially in the way that the viruses perform their infiltration.

3. The obvious answer may be to improve the design of the game's underlying system and vamp up the technical barricade against malicious players, but this is definitely easier than done. With each new addition to the system, hackers/cheaters will attempt to find a way to trump it. And it's likely that they will be successful. There is some hope for cheats involving the way a game is designed (ex. colluding), and the game can be made so that unfair gameplay can be monitored and, if suspicious activity reaches a certain threshold, the player can be banned or warned; in other words, an improvement in game design that will avoid such problems. As for scamming and other people to people trickery, perhaps a scanning system much heftier than the Warden can be created, but this is problematic for a variety of reasons besides its likely limitations, such as the player's right to privacy.

That'll do it!

4. Besides the fact that our source code is up on Github, I do believe security issues can befall the games. Indeed, why not? The game is literally up online and is not as secure as other games are. There is no authentication system implemented, nothing to check that the code is being dangerously tampered with by some external source. Hopefully, no one would think of infiltrating it...but that's being idealistic.

5. I definitely think these issues can apply to mobile games. Cheating by methods such as colluding or exploiting inadequacies in game design can be very doable in a game of any platform. Security bypassing and authentication hacks can be very possible as well. The growing trend of interconnected networks and shared media is both impressive and treacherous. Now phones are equipped with ways to easily access anything from the Internet to apps. A mobile game can participate in this giant network, and thus, be susceptible to a lethal security attack.

Josh Mermelstein

1. To answer a tangential question first, the most fixable vulnerabilities are technical ones. Even in the most technically secure game, it is impossible to force players to have strong passwords and follow proper personal security practices. As to which is a bigger problem, I'd say that technical vulnerabilities in a game can probably be exploited at a much larger scale. A social engineering attack can only affect as many players as the attacker can interact with directly. However technical vulnerability could easily affect the entire player base.

Loved this one too

2. My favorite vulnerability was the Second Life QuickTime vulnerability. It is interesting to me because it is not intended as an attack on an avatar/character but on the person using that avatar. I similarly like the Age of Conan directory traversal attack because it used the game as a vector of attack rather than a thing to attack.

3. As I said in question 1, I don't think some vulnerabilities are unfixable because of the fact that it is humans playing these games. A largely infeasible solution to some technical problems would be to force players to play games which are run on hardware owned by a 3rd party and the video of which is streamed to them. This would make it nearly impossible to execute cheats such as modifying textures to make walls invisible or map hacking in RTSs. In many games though, the latency issues would make this type of solution impossible.

   To fix problems like the one experienced by Disney in their Pirates of the Caribbean game, they could perhaps check the hashes of local game files against the expected hashes of those game files to see if the files have been modified and refuse access to players whose hashes are wrong.

   To deal with directory traversal attacks a game could refuse to execute scripts unless the hash of the script is on a built in whitelist.

   I'm not sure what to do about runtime compiling to cheat. I think some OSs have the option to make memory executable but not writable. A superuser might be able to change those permissions though? I'm not sure I have the background to speak intelligently about this type of attack.

4. A user could trivially cheat at our 2D game. Nearly all relevant variable like player health, boss health, jump height, number of jumps, weapon damage, number of bullets, etc are trivially modified in the python files that compose the game. I expect it will be more difficult to mess with my final project but I don't know much about android app security.

   I'm of the philosophy that in a single player game with no leaderboards or other inter-player element, there is no reason not to let a player cheat. I don't see any reason to tell them how to enjoy my game.

5. These issues absolutely apply to mobile games. Hearthstone, Blizzard's take on the CCG genre which frequently has tournaments with prize pools in the hundreds or thousands of dollars. just came out for iPads. If there is a vulnerability which gives players on iPads an unfair advantage, someone might abuse it to unfairly win money in a tournament.

   Right now, the tools to exploit mobile games are currently less mature than the tools to exploit PC games but as the money flows into mobile game competition, I forsee those tools maturing.

Andr

Ming C

4/8/14

Assignment 4

IEEE Security & Privacy Securing Online Games Readings

1) **After reading all the papers, what do *you* think is the main reason for vulnerabilities or insecurity in online games: technical or non-technical?**
   a) I believe that the biggest reason for vulnerabilities or insecurity in online games isn't the technical problems, but non-technical. Since these things are "games" people do not see them as real crimes. For example, stealing a 100,000 gold in an online game is not punishable by law, but stealing 100,000 dollars in the real world is. Even though the gold and money are both acquired by devotion of time and hard work, the virtual currency is not considered real. This overall attitude extends to other aspects of the game. Since cheating is not punishable by law, hackers and cheaters are always prevalent in games and will never stop.

2) **A number of vulnerabilities and exploits are described in the papers. Which is your favorite?**
   a) My favorite exploit is the whole "sweatshop" concept of virtual economy. The idea that to cheat the virtual economy system, there are places set up to "farm" or "mine" virtual currency is fascinating. In order to do this, without bots, actual players must play the game solely for the purpose of making money. They do not even play the game correctly, just in an efficient way to make as much money as possible. To believe that some games have virtual currency that is this important is interesting to me.

3) **What is your solution to reduce the vulnerabilities and exploits in online games (PC)?**
   a) My solution to reduce the vulnerabilities and exploits in online games is to have a very strict administration process. In order to cheaters and hackers to go away, administrators must punish them for breaking the rules. I know that these systems are already in place, but they usually end in banning for a certain amount of time. In order to online games to crack down on vulnerabilities and exploits they must become stricter with punishment. Then, everyone will be too afraid to cut corners.

Don't we have this already?

4) **Do you envision these issues to happen in your 2D game or final project?**
   a) I'm sure there will be a way for hackers or cheaters to find a glitch in my game and use it to their advantage. In World War Zoo, I know there is an area in the map where if a person just waits there they can never be killed, which would take away from the game. I hope that cheaters never use this, but it is entirely possible.

5) **Do you think these issues apply for mobile games?**
   a) Of course. A majority of mobile games are based on who can get the highest score globally. They have leaderboards that everyone can see, and some hackers and cheaters really enjoy seeing their names on the top of that list. Therefore, they find ways to get the highest score possible in the game, not by playing it but by using a bot or finding a glitch. Do I really believe that someone can get a 999,999 in Flappy Bird?! I know that's impossible!!!

Jessie Chapman
4/8/2014
Comp23: Game Development

Assignment 4: *IEEE Security & Privacy Securing Online Games* Readings

The main reason for vulnerabilities/insecurities in online games is non-technical: it derives from the incentive for players to cheat in order to gain an in-game advantage over other players. While increased player-to-player communication, external applications, plug-ins, and other opportunities to bypass game servers leaves players' computers vulnerable to attack, cases of stolen financial and personal information are few and far between when compared with other cheat methods. The bots, scams, collusions, and virtual "entrepreneurs" negatively affect other players' performance, within the game world itself and psychologically, by getting them to waste money on virtual goods that is eventually stolen from them or making them feel so incompetent that they unsubscribe from the game.

The exploit that I found most interesting was the exploitation of a game's architecture and dynamic language. While it would take a proficient hacker to find break in and get their hands on source code, after that point it is relatively easy to change constant variables and certain rules of physics within the game to give yourself an advantage in the game. Avatars would be able to reach spaces in the game world that normally they wouldn't be able to in order to get more virtual currency, jump higher, and overall perform at an exceptional level that wouldn't be possible without this cheating mechanism. This is also one of the more straightforward and obvious methods of cheating, but also has immediate rewards to that player with little consequences to other players (compared to manipulating the virtual economy to purposefully give other players a disadvantage, this is more self-centered).

To help reduce some or all vulnerabilities and exploits in online games, there needs to be concrete laws surrounding virtual property. Because most cases of cheating deal with game economy in some way or form, and laws surrounding virtual goods don't exist because it's "not real" (but, indeed, very real), there is a state of anarchy in the gaming community. Theft of intellectual property and virtual wealth is occurring in online games, with real monetary consequences for players - hence, this type of theft is truly negatively affecting the well-being of members of our society. The addition of laws for virtual property that already exist for physical property would act as a deterrent for these types of exploits, just as they deter theft of any other kind.

But will lawmakers treat this issue seriously?

I don't envision these kinds of issues to be a problem for our final game (unnamed so far, but deals with shooting spells at wyrms to "restore color" to your village/kill the wyrm), as the only thing to be gained within our game would be to give yourself an advantage over the AI. This might come in the form of unlimited spells to use or insta-kill of the wyrm, or using bots to continuously kill wyrm after wyrm. Since it is a one-player game and there are no plug-ins or connection with a game server or other players, there would be no incentive to manipulate files to extract player information or any method to directly influence other players like there are with MMORPGs.

These kinds of issues could definitely apply for mobile games: hacking and malware would still pose a threat, as more and more people are storing tons of personal information on their phones (especially smartphones, with direct access to bank accounts, financial information, storing passwords on documents on their phone, etc). Extraction of source code and use of bots to gain personal advantage would still be possible, but might need extra platforms to work and would overall be more complicated than on traditional desktop computers. Any psychological vulnerabilities (scams, theft of property) to other players is platform-independent, and so would still just as easy to implement for mobile devices. However, malware would definitely be trickier to implement, as the surface for player-to-player communication in the form of mods, multimedia files, external applications, etc is significantly decreased on a mobile platform.

Jared Bronen
Comp 23
Assignment 4

After reading all the papers, what do *you* think is the main reason for vulnerabilities or insecurity in online games: technical or non-technical?

I believe the main reason is entirely technical in that the complexity of online games is so far out of reach that it is near impossible to adequately test for all possibilities of vulnerabilities in a normal development cycle. The non-technical aspect of this is that the business/competitive idea that more features must be added constantly to keep online games fun to play, but the issues and exploits are almost always technical in nature.

A number of vulnerabilities and exploits are described in the papers. Which is your favorite?

Loved this one too

My favorite is definitely the loaded multimedia; this was talked about in class, where it was discussed how one could load a picture of video onto a billboard and all users in Second Life who saw it in game would be be vulnerable to the attack. To me, this is super nefarious, using user-generated content to spread malware and spyware. It relies on the fact that old library functions, which might have since been updated to protect users from vulnerabilities, are used to gain access into user's machines.

What is your solution to reduce the vulnerabilities and exploits in online games (PC)?

My solution is more on the business side rather than technical. Spend more money and time testing and searching for security vulnerabilities in software. There are normally teams dedicated to art, programming, patching, writing, etc., and there should be a dedicated team of programmers who are constantly searching and tidying up any security vulnerabilities that might arise.

Insightful. Perhaps solution to movie piracy.

Do you envision these issues to happen in your 2D game or final project?

Our game is not online, and there is no in game currency that translates to real-world wealth. But since we are using Python, which is a dynamic language that relies on an interpreter, our code is most certainly vulnerable to modification. But we're also releasing the source code online, so the above point is pretty moot.

Do you think these issues apply for mobile games?

It most certainly can. There are quite a few games that involve online play that rely on in-app purchases of gold and and other material. Android is fairly susceptible to malware, and account information can always be compromised. An android phone is, after all, a type of personal computer; so it maintains most, if not all, of the vulnerabilities that online games for PC have today.

Ryan Schumacher
April 8th, 2014
COMP 23 Assignment 4 – Security and Cheating

Question 1: While it is impossible to blame any single reason for the vulnerabilities in online games, the massive scale of online games is their Achilles' heel. Simply put, the bigger the user-base, the more potential for exploitation (read as: "There are always idiots in big groups of people"). When millions of users play a game, some have no idea about proper computer and internet security.

Online chat is common to all MMO games. Even if the game does not support scripting, giving a hyperlink or sending just a URL is enough to make some users fall for a scam. This occurred frequently in *RuneScape*. There were always chat bots standing in crowded areas spamming a link to a website that gave you "free gold." Sure enough, I would always see a poor user approach the chat bot and say "Hey! I went to that website and now there's a virus on my computer!" Furthermore, there are too many people trying to exploit the system than can be caught. Again, citing *RuneScape*, when you reported a player's misconduct in that game, a clip of the past minute of your play experience would be sent for review to determine if the reported player did anything malicious. JaGeX, the company that produces *RuneScape*, would receive hundreds of thousands of these clips every day. This gives a very low probability of being caught! Banning a user was an insignificant consequence as well; the scammer could simply create another account. Across the board, the benefits of cheating tend to outweigh the insignificant consequences. According to Jeff Yan's article, there isn't much legal protection in these situations.

Question 2: Modifying the client infrastructure was a technique described in *An Investigation of Cheating in Online Games*. This caught my eye because it was a way of cheating at a game without modifying the game itself. For instance, if you modify your graphics card to not display certain textures or walls, you gain an edge over your opponents. I think this method could be extended in high-stakes FPS games. For instance, you could modify your sound card to play the sound of enemy footsteps at a greater volume to hear them approaching. Unless the game you are playing keeps tabs on your media drivers, this seems near-impossible to detect. Very crafty and very helpful in the gaming world!

Seconded W. Anzu

Question 3: One solution is a mandated tutorial for all users that makes the user aware of common scams, such as sending malicious links via chat, and makes them aware of illegal practices, such as buying gold online. By giving examples of shady behavior, the users will be less likely to fall for scams. Additionally, all games *must* have a basic tutorial on how a user should never give out his or her password, name, address, etc. As I mentioned earlier, there are *always* idiots out there who *will* fall for these.

Question 4: I don't envision my final project being able to connect to the outside world and share data such as high scores or achievements; nor do I anticipate a large user base for my final project. However, given the timeline, I do anticipate a large number of technical bugs which would result in the user being able to cheat. Last night, I found a bug in the 2-D game that essentially lets the user play without taking damage. Another bug lets the user progress through our maze without actually having to explore it; the ship can walk through walls if given the right angle. Problems of this nature are what I expect for the final project: potential to cheat, but no potential for exploitation of others.

✓ **Question 5:** These issues can absolutely apply to mobile games! In fact, I think mobile games are riskier than online games. Think about how a user gives an app certain permissions (access to contacts, pictures, etc.) People's phones are a hub for their personal information. Name, address, phone, and even credit card information is stored on people's phones. Mobile developers need to take caution, because if their games are exploited, a lot of personal data could be stolen. Passing executable code into a text field in a game could be one way to exploit mobile systems.

Will Hickey
4-7-2014
Comp23- Assignment 4

I think that the main reason for vulnerabilities is just a lack of creative thought and time spent by the game development team on the issue of security. Most of the time it seams to me that the vulnerabilities are either the developers did not even think about it or that they ran out of time while trying to finish and push their game.

thank you

I liked the ones that did not harm the players or their properties, but the ones that I think are the cleverest are the cheaters that flood their opponents line with request so that they can not connect with the game server.

I think that the best solution is not to try and create a hack proof online game but rather to not give the hackers an incentive to hack the game to begin with. In most cases this would be to remove the money transactions that happen in game. Path of Exile has created a world that has no in game currency but rather is based on trade.

In my groups 2D game I can see people changing the code. Especially since all of the numbers for the stats of the hero and enemies and weapon damage is in a single file. They could literally go down the list of global variables we have and change them at their whim.

SP

I think the issue of cheating defiantly applies to mobile game, but has not been brought to the public's attention because no major games/ large amounts of money have been hacked yet, but it is only a matter of time.

Louis Ades 4/8/2014
COMP23: IEEE Security & Privacy Securing Online Games Readings Ming Chow

1. I think that the main reason that many of these problems arise is non-technical, and sometimes the fix for them may be something technical. These fixes need to arise usually after the developers made a non-technical decision that opened up a security hole. Once in a while, there may be a lack of removing injections into textboxes or ways to overload the memory of a server, but these holes pale in comparison to the non-technical decisions to allow for HTML and JavaScript-executing in chatboxes, or for writing an entire game in Python and then including all of the game's globals in the client rather than on the server. Those aren't bugs that may cause a client to crash--those are design flaws that can go as far as to permanently damage players' personal property.

hehe

2. The funniest security flaw is easily the Walking on Water Pirates of the Caribbean story (aside from the fact that a 3D game was written in Python--I used to have it, and it did run terribly--the fact that Python's globals were handled locally on the client-side and that they were given such obvious names is a sign of incredibly laughable technique). However, the most "WTF were they thinking?!" case study was the Anarchy Online and Age of Conan HTML- and script-injecting issue. The ability to write custom scripts and execute them straight from a chatbox is a *huge* problem, but could've been fixed if they decided to take the necessary precautions and perhaps scan any script before it is run. However, the developers made a conscious decision to include this feature, and that is a *huge* security hole that can be exploited to the worst extent. Including HTML and Javascript compatibility inside a chatbox is an incredibly horrible design flaw, and these are powerful languages...and this was a *choice* to include it in the game! They could've *at least* made their own scripting language and barred it from any super-powerful functionality.

See Anzu's response You're one of few

3. As a gamer, while on the internet I'm always making a conscious decision to avoid suspicious-looking stuff or to only browse websites I trust, I often don't want to have to think of that constantly while playing an MMORPG. So while I'm all for openness in most cases, online gaming is an exception for me in terms of what to include outside of the client. I'm a big fan of the way EVE Online handles its security--it has its own clients for just about everything. It has its own scripting for writing cool stuff in chat boxes. Players can exchange real-world money purchases with in-game currency, thus making farming sweatshops obsolete for the game. Anything that's exploited in-game is not a bug--it's a feature, and sometimes that feature gets fixed, and sometimes it doesn't. In the long run, players are generally made aware from the beginning of the game that the game world is dangerous, and that people will try to take your trust for granted. In this way, players are prepared for scamming and trickery from other players, and every bad decision they make only educates them further, but no players are ever given the ability to go as far as to run malicious code on another player's client. In fact, this is the type of openness I enjoy--tight security for the game's code and mechanics themselves, but the mechanics themselves allow for player openness. This way, the worst security holes and flaws cannot be exploited, and the players are prepared for the stuff that's not quite as harmful.

✓

4. These issues are almost definitely going to happen in my game's final project. I have no idea how they would, but there's always a possibility even in the strangest cases. I do believe, however, that my chances are lower as I'm writing it in the Android SDK (which uses Java) rather than in Python + Pygame.

✓

5. Of course they apply to mobile games as well--if a game grants connectivity to others' phones as well, the wrong security hole can grant one player access to another's, phone number, contacts, personal information, passwords, credit card information, etc. And because mobile is such a huge market, there are probably more people of malicious intent than ever on the Android and iOS platforms.

I wanted one page man!.

Arthur Berman

1. After reading all the papers, what do you think is the main reason for vulnerabilities or insecurity in online games: technical or non-technical?

A major reason for vulnerabilities and insecurity is the social reason that people will go to extreme lengths to obtain a slight advantage in any competitive environment, whether virtual or analog. The sheer effort expended trying to find vulnerabilities creates their abundance. As games become platforms for financial investment (free-to-play games, Gold Farming, DLC, etc.) they become targets for exploitation; such exploitation can be lucrative for people willing to compromise the system. Because exploiting games is rarely prevented by law, it is relatively risk-free compared to other crimes.

2. A number of vulnerabilities and exploits are described in the papers. Which is your favorite?

Really hard stuff

As with any discussion of vulnerabilities, my favorite is the example involving overwriting the call stack with executable code that compromises the victim's computer. This vulnerability can be found in nearly any program on any architecture – a single insecure string input can compromise the entire system – and when it presents itself the options for exploiting it are enormous.

3. What is your solution to reduce the vulnerabilities and exploits in online games (PC)?

Fixing the technical components to exploiting online games has definite, clear-cut advantages, but without heightened penalties for cheating, it will continue. In order to reduce vulnerability, laws need to be introduced to punish any exploitation involving the loss of player-owned property, virtual or otherwise.

4. Do you envision these issues to happen in your 2D game or final project?

Soubhik Barari

1. From this brief foray into the study of game security, I would assess that primarily game security and cheating are primarily linked to *technical* elements of development. On the client-side of gaming there is certainly a decision-making/economic layer that is unlike most other fields in software. However in the end, we must remember that like any other software (online banking, private video streaming, etc.), there are abstraction boundaries and constraints that must be considered in maintain security. This must constantly be juggled with performance and computational reliability. Thus in this balancing act, game developers are faced this technical quandary – do we sacrifice game performance (an absolute feature of games) for game security (a grey area that is still being explored and defined)?

2. A core vulnerability was described in the architecture of the client-server software that online gameplay occurs on. Because this architecture is essentially relaying between sender/recipient clients and a server, there are targets on both ends for attackers. Particularly, it is likely that an attacker will "spoof" messages from the server back to a client – this may bypass less stringent input validation and affect the end client. Increased 'relays' only further complicates this online architecture allowing for more exposure to attack.

I loved this one when I reviewed the papers

3. One particular vulnerability that stuck out to me was that of exposed embedded virtual objects on player-controlled servers specifically in *Second Life*. It appears that a core problem is the lack of content control by the native game servers – user-uploaded content is certainly a viable feature, but by having a user-controlled buffer server, the game is exposing itself to attacks. Perhaps if the game server itself hosted this content, there could be heightened monitoring of security breaches as well regulation of media libraries and usage of only in-house, approved plug-ins. These would all be more preventative of security threats, however it comes at the price of increased server computation.

always helps

4. I imagine that these issues could potentially apply to our own 2D games – though there is no online/player interactive content, there is certainly open exposure to our source code and assets. Python being dynamically typed and typically not compiled does make it a little harder for an attack to happen in a 'break' in the game 'pipeline.' But nonetheless, we have done very little with user input validation.

5. These issues certainly apply for mobile games. Mobile games in this day and age rely heavily on online content (from PNS servers and such), where there is the same exposure to the server-side/client-side attacks mentioned above. Furthermore, native mobile platforms likely do not have as powerful virus/malware protection bundles making game security even harder.

IEEE Security & Privacy Securing Online Games Readings

Reading Response by Brian Pilchik

1. After reading all the papers, what do you think is the main reason for vulnerabilities or insecurity in online games: technical or non-technical?

Vulnerability primarily comes from non-technical sources. Sure, there may be a vulnerability in Quicktime, or a buffer overflow exploit that ought to be closed up, but at the end of the day, it really comes down to higher-level choices about the freedom developers are willing to give gamers. Is it safe to store so much data on the client-side (the "trust" boundaries mentioned in the readings)? Is it safe to assume that insiders won't abuse privileges? Is it safe to allow players to upload custom-designed items without a screening process? The technical details are important while patching, but the bigger-picture design choices will determine whether or not technical issues have the capability to really disrupt intended gameplay.

2. A number of vulnerabilities and exploits are described in the papers. Which is your favorite?

"An Investigation of Cheating in Online Games" mentioned an exploit in which users modify their own graphics drivers to render opaque walls invisible. I think that's brilliant. Even if information about the wall was stored server-side, *of course* the client graphics card needs to process it, so changing the way it interprets that command is an incredibly hard-to-fix hack. Short of not allowing players' machines to know anything about what goes on behind walls until they round the corner, I have no idea how developers could compensate for it.

Me too

3. What is your solution to reduce the vulnerabilities and exploits in online games (PC)?

I've been a proponent of server-side data for some time now, mostly to combat software piracy, but it translates to cheating, too. If the key information needed for gameplay is stored somewhere the developer owns, and that server can constantly be checking to see if you're a unique and authorized user, and whether or not your files match up with theirs, the developer can have more control over the player's experience. Of course, that requires faster internet connections. And it doesn't solve every problem - including the issue of custom-built virtual items. Perhaps, by extension, those items need to be verified and regulated by and on developers' servers.

4. Do you envision these issues to happen in your 2D game or final project?

Yes and no. Our games aren't advanced enough to allow for players to interact with other users online, so I'm not worried about identity theft or script execution on my server. But if someone wanted to modify the .txt files that builds my levels in my 2D game, that would make it a whole lot easier to win. Good thing we're not monetized! Lol

5. Do you think these issues apply for mobile games?

True

Absolutely. I don't know much about it, but I understand that Android devices aren't too hard to run your own code on. I would think that opens their apps up to all kinds of manipulation. Off the top of my head, I can already see how I would perform disconnect attacks to avoid losing online games (turn off my data/WiFi) or collusion attacks for team games.

Todd Pollak
4/6/14

Assignment 4: IEEE Security & Privacy Securing Online Games

1. I think there are several reasons for insecure online games. The first cause is the ever expanding complexity of games. The more massive the system, the more likely there are to be bugs. Two is that dynamically typed systems are easier to hack since type is only inferred at runtime (see Disney's Pirates of the Caribbean). Third, are the limitations of technology and costs. For massive online games with shared worlds, it would be two expensive to check and validate all data on the server. Games implement a client-server architecture that allows code to run and potentially be hacked on the clients computer. Games also have P2P models for chatting that allow users to pass each other text unvalidated. On the non-technical size, games are vulnerable because games are a huge and mostly unregulated market that seems like easy pickings.

2. My favorite exploit in the articles was one found in the paper "Reducing the Attack Surface in Massively Multiplayer Online Role-Playing Games." The article described how in games such as Anarchy Online, users would exploit the chat system by sending PM's containing links in HTML format. The links, if clicked on, would then downloaded a cookie on to the targets computer. After all that, the attacker would send another HTML link that then contained a game script file that would execute code in the cookie. It sort of blew my mind that users could exploit game's scripting system like that.

3. I think reducing vulnerabilities in online games is a very difficult problem to solve. There is seemingly always going to need to be a tradeoff between ease of use and speed versus maximum security. However, one way to solve problems would be to focus on the most game breaking cheats such as bots flooding and online economy with too much gold. To combat bots, I would suggest using the algorithm developed by Mitterhofer, Platzer, Kruegel, Kirda. Until the hardware and software is powerful enough to run all the code on the server, I think there will always be vulnerabilities. But, games have some choice in which vulnerabilities they can expose. Try to keep essential game code on the server and not with the client. Games could also implement some sort of secret handshake method with the client to make sure it hasn't been altered.

4. The Two-2d game project is very vulnerable to attacks because the users have the code right in front of them. It would be trivial to change anything about the game. I don't think my final project will be that vulnerable to a serious breach because there is nothing worth stealing and no online connectivity. Lol, true

5. Of course these issues apply to mobile games. If anything mobile seems more vulnerable since it is newer and phones seem to hold everything important in our lives. Android has already proven to be very susceptible to malware, so I would be surprised if hackers weren't trying to exploit mobile games.

yup!

Anzu Hakone

Security & Privacy Securing

1. I think there are different reasons why online games are a target for these vulnerabilities and exploitations. For the casual gamer, hacking is mostly for getting ahead in games whether it is because the game is too hard to beat or they want to get achievements quicker. This would be an example of a technical reason. What makes hacking in games a controversial topic is in the case of organizations. For these organizations with sweatshops, the purpose of hacking is for monetary gain. Because it is usually not illegal to exploit games in this way and the workers are technically "working," organizations can make a huge sum of money (talk about spending $100,00 in an online game!).

2. My "favorite" exploit would probably be the concept of botting. I think it walks on the edge of being terribly illegal to being sort of harmless. If used in a game sweatshop, where organizations are actually profiting in great sums, it forms an entire black-market. Not only would it be taking away from the game developers' profit, but it also endangers the players who purchase from these markets because the players will be giving the black-market sellers access to their monetary information. However, I think if used by a common individual who just wanted to get ahead of the game, it will only affect the player rather than exploiting the privacy of other players. Of course, he/she can still do so on a larger scale, but I double it will ever get as big as generating the same amount as the sweatshops.

3. I honestly cannot really provide a solution to avoiding these vulnerabilities. It is scary to think that although the player is educated enough to not be actively clicking and going to websites provided by other players, simply interacting with an object in a game can trigger downloading malware (as in the QuickTime example). Although filtering all of the exchanged content between players may reduce these exploitations, I think it is important to have a space where players from all over the world can interact with each other without much monitoring.

Perhaps that IS a solution

4. I can see these exploitations could happen in both the 2D game and the final project because although neither of them currently contains any downloadable content, code can be generated to predict and complete the tasks (an AI). Even though our 2D project (Microburst) uses the random function, randomness in computers are not exactly random. If a hacker can determine their time stamp and how the randomness is calculated, the positioning of the enemies can be predicted. The final project should be safe in terms of predicting which symbols appear in the memory game because it would not matter, but beating a memory game can be easily done on a computer. The computer just needs to remember the elements and the positioning of the elements.

very insightful

5. Yes. I think these issues apply to all games that are played on a device with Internet access. Any game that lets players interact with other players is susceptible to exploiting the progress of the other players. As great as the Internet is for it gaining information and interacting with others, it also means that we are putting ourselves out into the Internet.

10

Matthew Cardarelli
Comp 23: Game Development
Due 4/8/14
Assignment 4: IEEE Security & Privacy Securing Online Games Readings

In considering the root cause of vulnerabilities and insecurities of online games, I believe the main cause is technical. Observe the various categories of cheating listed in "Investigation of Cheating in Online Games." There are only a few non-technical vulnerabilities listed there, such as "exploiting a lack of authentication" or "social engineering." These items are vastly outnumbered by the various technical abuse categories, including everything from breaking into source code to setting up bogus servers to exploiting game design flaws. Furthermore, the non-technical issues can generally be countered by simple developer-side additions such as authentication, or are otherwise nearly impossible to prevent, particularly internal misuse. It is the vast array of technical vulnerabilities that form the most dangerous and effective means of cheating or otherwise misusing online games.

Of all the vulnerabilities listed, the most intriguing to me was the BlackSnow Interactive vs. Mythic Entertainment lawsuit discussed in "Virtual Judgment." BlackSnow's argument against Mythic was a very interesting one; they claimed that disallowing the sale of in-game items for real-world money was anticompetitive. It, in my mind, brings right to the forefront the question of who can claim ownership of virtual property. This only adds to the suit's exciting but ultimately anticlimactic potential to challenge the authority of a publisher's EULA. Perhaps this lawsuit is not considered an exploit in the sense of a player abusing the game, but it certainly was trying to poke holes in Mythic's power over Dark Age of Camelot and exploit the ambiguity of the EULA.

Insightful

One solution I have that pertains to design and non-technical cheating is to create tools that allow player communities to self-moderate. I'm sure this would make many people cringe when they think of giving selfish, obnoxious gamers power over each other, but I think if this system were aided and nurtured by the developer, it could be very effective. Many online games possess a matchmaking function. The system I have in mind would allow players to create and moderate their own matchmaking communities. In this way, the communities that grow are the ones that have the best balance between cracking down on cheaters while avoiding bans on innocent players who are wrongly accused. The system could integrate player reporting, player reviews and ratings that are tracked for moderators to review, and the ability for community owners to appoint assistant moderators with varying levels of power, similar to online discussion forums.

In our 2D game, it is quite possible that there are bugs which could be abused. For instance, when you switch your weapon and that weapon had previously fired bullets that are still in the air, those bullets freeze. Theoretically you could fill the air with bullets, and then start switching between all the guns and making them all travel slowly across the screen.

Mobile games must deal with all the same issues as computers, considering modern-day phones are basically mini-computers anyway. In addition, the simpler design and limited storage on phones likely makes it harder to fit adequate security into the phone, and allows hardware experts to easily disassemble and modify the phone to enable cheating.

lc

Jim Bonish
Comp 23 Homework 4: IEEE Security & Privacy Securing Online Games

1. I think that the obvious reasons for vulnerability in games are technical issues dealing with the performance, practicality, and costs associated with perfectly securing an online game. It would cost a game developer a lot of server usage to execute all game calculations via the network as opposed to the clients' machines, and when calculations do happen on the client side, there will usually be some way for the client to hack them for his own good. That said, I think the more impactful reasons that it actually *happens* are both its profitability in some contexts (there will always be a real-world market supply/demand when there is money to be made) and the lack of laws denouncing it.

2. My favorite vulnerability described in the papers is the ease of hacking games written in interpreted/dynamic languages like python. One reason I like this possible exploit is that it seems like something I could plausibly do if I had the AntiFreeze GUI tool that blows up the pyd file. It even has variable names so you can infer what they control. It allows for the fun, non-malicious, kind of cheating that I would be interested in, like the examples they use in the Pirates game – jumping super high and walking on water. It can also expose the infrastructure of games, like how you can jump over a wall and through a roof in the Pirates game because they didn't implement roof collision detection because they assumed it would never apply.

I wonder if lawmakers would be serious

3. I think establishing concrete laws is a more practical solution to game security problems than trying to prevent all possible technical hacks; I would guess that hacking would drastically decrease if the consequences for getting caught were real-life consequences. If it's profitable for you to cheat at WoW by collecting items or currency and then selling them for real-world money, you'll do it if the only risk is that Blizzard catches you and bans you from WoW. That's not a big deal. You'd definitely think twice or thrice if Blizzard could report you to authorities that could impose major fines or even jail-time.

4. I don't really envision these issues happening in my 23 projects unless I develop them far enough to get them out there. Though it is definitely possible to take our 2D game that was written in python and hack it like the Pirates game could be hacked, I can't imagine anyone putting in the effort.

5. I don't think that all the same issues apply for mobile games for a number of reasons. The following sentences should be taken with a grain of salt because I'm not overly knowledgeable in this field. I think that hacking mobile devices is a whole different animal because the phone providers like Apple can detect it and end your plan or something like that. Also, I don't believe it would be very easy because I don't believe apps are ever written in interpreted languages.

ok fair enough

Comp 23 Game Security Response
David Taylor
4/8/2014

Interesting

1. The insecurity in online games comes from the lack of legal precedent on the subject. The idea of having a company be the sole enforcer over issues that may arise in these massive games is an ineffective model. There needs to be more punishment than account banning for people to fear cheating in games. Your normal player usually does not do enough damage to draw recognition from the game company and the larger cheating groups are not as tied to single accounts. These groups that bot farm and commit other infractions can easily create a new account and continue farming their service.

2. The use of Bots to level accounts or repeat simple actions to collect money is my favorite exploitation of games. This is an interesting problem to tackle, because the user is technically using the game inside the rules set out for the game. There are not as many obvious signs that this is not another average as there are with those who break the games rules with their own items or other hacks that give them an innate advantage over other players. These bots just repeat simple actions and stay out of the spotlight. With games such as WOW, with a user based thousands of times larger than their workforce, it is hard to dedicate time to finding these bots. The user based doesn't really care because the bots are not hurting them.

Consider 4chan?

3. Crowdsourcing the discovery of cheating players is by far the most promising solution to the problem of cheating. Letting your user based activity report things they think are cheating gives you a massive force that is invested in making your game better and cheating free. This workforce is unpaid but have as much invested in ending cheating. These users will continue to be involved as long as you show that their work is providing results.

4. I do not but purely on the fact that there is nothing to gain from hacking my game. There is no reward for my game and no money invested in it. These users may chose to do it educationally, but this also will not help them much because the game is not coded by any sort of industry standard and will most likely be much easier to tamper with than a game produced by a company.

5. These issues absolutely apply for mobile games. There is lots of money pumped into the Facebook arcade style games along with other games like Clash of Clans. These games are played by hundreds of thousands of people and both have ways that you can pay money to increase your games appearance or strength against other users. If you were able to get accounts to these levels by cheating, people would pay to not have to invest the time and money to reach these goals the way the designers intended.

Insightful

Joe Canuel
4/10/14

Comp 23 Assignment 4 : IEEE Security & Privacy Securing Online Games Readings

1. After reading the papers, it is clear there are both technical reasons and non-technical reasons for vulnerabilities and insecurities in online games. However, I think the technical problems are the main reason (or bigger issue) for these insecurities. The papers seem to suggest that technical insecurities can lead to much larger problems, such as manipulating client-side data to create in game cheats or duplicate items/in-game currency. If a player can create in-game cheats (such as wall-hacks, improving their stats, or other things), the cheater will have a huge advantage over other players and can make other players quit the game. If a player can duplicate items, it can lead to a monetary profit for the cheater (as players can often sell virtual items for real money) and a ruined economy for the virtual world.

2. My favorite exploit is when players violate the trust that developers have given them. In particular when player's edited variables in the python code in Pirates of the Caribbean Online. While this cheat wasn't the most malicious one described in the papers (as it didn't seem like the hackers made any significant monetary gain), as a gamer I thought it was a very interesting hack. I have experienced many hackers across a wide range of games, and never knew it could be so easy as changing the game files on your computer. In addition, I thought it must have been pretty funny to see players jumping ridiculously high and walking on water.

LOL

3. I think many technical problems have solutions while some non-technical problems can't be solved. For example, if online Bridge partners communicate via phone to discuss their hands, they really can't be stopped by the game developers (at least with today's current technology). However I think there may be ways to stop or at least reduce the technological problems, such as writing code in a non-interpreted language like C (and not Python) so its hard for users to change the game files. Another solution would be to implement the bot following algorithm described in the paper "server-side bot detection in Massively Multiplayer Online Games".

4. I think many of these issues apply and could happen in our final project, although since our final project isn't web-based I don't think the severe issues apply. The only thing I can imagine happening is a cheater modifies the code so he/she can obtain a high score (which isn't that malicious compared to attacking other users or doing something for a monetary gain).

Nice thought

5. I absolutely think insecurity issues apply for mobile games. Mobile devices have a lot of vital information that attackers would love to have, such as contact information, saved usernames/passwords, and maybe even saved credit card numbers. Thus I think the less important issue is if a user is cheating in a game (especially a single player game), but rather if a user is trying to gain control of another user's mobile device (such as hackers were doing in Age of Conan).

Ben Helm
4-7-2014
Assignment 4

1. It is my belief that regardless of the technical advances in video game development, the main reason for vulnerabilities resides outside the code of the game. The reason for gamers finding ways to exploit games does not stem from the fact that the opportunity exists; it comes from the fact that they were looking for it. In general, PC gamers tend to be those with a higher than average understanding of the inner workings of their computers, and often tend to have a mischievous side when it comes to playing games. Because the laws regarding video game exploitation and cheating are so vague, there is little to no deterrent for doing so. Finding a new way to duplicate weapons in World of Warcraft may simply be a personal hackathon problem with a cash prize. However, the threat of a lawsuit or even jail time would surely make the average bored basement hacker think twice about their actions before attempting to do so.

LOL

2. My favorite vulnerability mentioned was the developer created disease that killed thousands of player's characters in World of Warcraft. After working on the 2D game project I have a perspective on how even on a very small scale the interactions between players and other elements of the game become immensely complex. Even with a rigorous testing plan it is difficult to determine every combination of events that might occur while the game is played. In a MMORPG such as World of Warcraft, where millions of players subscribe, it is truly commendable that these occurrences aren't more common. However, when they do, it is an interesting way for the average gamer to gain a glimpse of how complex and almost organism-like a large video game can be.

good luck w/ that. See SimCity

good thought tho

3. If technology were up to the challenge, making entire games server-side could help to mitigate the problem. The less data that is stored and processed on an individual's PC, the less they have to alter and exploit. However, this would put a massive amount of stress on the game servers, and would result in significantly less complex games. It would also not solve the issue of bot spamming. An over-arching solution to the issue would be more stringent laws to deter cheaters. For many it is more of a hobby than a livelihood, done purely out of a desire to crack the next puzzle and win the prize. Because the framework for tracing all the activity that occurs within a game is already in place, it would not be overly difficult to begin rigorously prosecuting people that break the rules of the game.

4. We ran into issues of exploitation more in the form of bugs than in the form of direct malicious intent. Because there was no economy or PVP within the game, there was really no possibility of a large-scale monetary exploit, or that of a bot that would automatically play the game to rack up points. However, had the game been distributed widely, there is no question that some way of cheating it would have been uncovered quickly, despite our efforts to preemptively fix them all.

5. While many of the issues specifically mentioned in the articles concerning MMORPGS, such as that with large amounts of in-game currency that can be converted to real US dollars may not apply to mobile games, the threat of exploitation still exists. It is hard to imagine a day where there is no gamer clever enough to find a strange way of tilting their phone such that it is easier to win a game.

10

Cody Chen
Tuesday, April 08, 2014
Security in Online Games

After reading all the papers, what do *you* think is the main reason for vulnerabilities or insecurity in online games: technical or non-technical?

I think it strongly depends on the nature of the game and perhaps more importantly what's at stake. Some games, such as the online Go game and League of Legends are extremely time sensitive; that is if a player drops out for a small amount of time they will be at a huge disadvantage. Other games, such as World of Warcraft and Team Fortress 2 are extremely wealth centric, and are more sensitive to account loss. It seems that for the most part, developers are very good at protecting your account information, but are not very good at protecting your internet connection, so the online GO and League are much more sensitive to technical attacks, whereas World of Warcraft and TF2 may be much more vulnerable to social engineering or other non-technical exploits.

A number of vulnerabilities and exploits are described in the papers. Which is your favorite?

Personally, my favorite kind of exploits are the fun ones and the victimless ones. The ones that let you change your jump height and runspeed, like in the pirate game letting you fly around and show off, but not have a hugely negative impact on other players. There's a gamemode in TF2 that puts players in a coop PvE wave-defense scenario, and one of the challenges in this mode is to collect money from fallen enemies. I admit to using a texture pack that makes the money more visible, because I think it makes the gamemode more fun when I'm less stressed about finding that one money clump that someone dropped and forgot about, and since it's PvE, no one is victimized.

Don't we already have this?

What is your solution to reduce the vulnerabilities and exploits in online games (PC)?

I think we need better reporting materials for players to accuse others of unfair play. This works best for things like aimbots and ddos attacks, but we need to be able to identify when a player was cheating or attacked, and compile a list of all the players involved. Cross referencing long term activity, we can find lists of players involved in a high number of these attacks, and systematically identify which ones are cheating. I also think positive reinforcement for reporters could help, if you help identify a cheater, you should be notified so that the act of reporting feels more dependable.

Do you envision these issues to happen in your 2D game or final project?

Considering none of these games have internet connectivity, the worst someone could do would be to make local changes to the code, which really isn't that much of a worry as they are the only one affected by these changes.

Interesting

Do you think these issues apply for mobile games?

Definitely, especially with the prevalence of account information on smartphones these days, having someone with access to your phone could reveal all your account information for email, banks, credit cards, etc. These games have fewer P2P connections than, say, MMOs, but it's possible.

✓

10

Andrea Compton
4/8/14
Game Security

1. I think the main reason for vulnerabilities in online game is non-technical. While technical bugs were the first vulnerability that was exploited I think virtual worlds have progressed to bigger issues. There are some technical vulnerabilities such as bugs that players can exploit and bypassing the server to give another player a bad script that reads information off their computer; however, because these are technical they have the potential to be fixed and have been fixed in the past. The lasting vulnerabilities, or ones that require more thought and legal action than a technical vulnerability are non-technical. These include the problem of buying and selling items for actual currency without a strict law enforcing rules that a good citizen would follow. Because online games have evolved into players virtual reality and "some statistics state that 20 to 30 million participants are so involved in gameplay that they spend more time engaged in digital societies than real-world activities" (Kane) rules need to be set in place in order to govern the society as if it were reality. Games are turning into worlds without a government and, as everyone knows, this results in chaos.

2. My favorite exploit in a game would be taking advantage of the source code as in the Pirates of the Caribbean example where players can walk on water. The fact that python is a dynamic language means that players can modify the source code and change variables that allow them to perform actions others can't. While players may gain more points or not die as quickly, this exploit is my favorite because it mostly only affects the player who modified the code. To me this is a legal cheat because it does not take advantage of other naïve players or try to obtain critical information from other's computers. It seems relatively harmless and used to make gameplay more fun.

Interesting

3. My solution to reduce vulnerabilities and exploits in online games is to create a virtual government that has the same rules as a real world government. This government should be put in place and enforced by the makers of the game. The government will enforce rules such as patent protection and safeguarding against theft. This will make honest players more likely to continue playing the game because they don't have to worry about theft, and will also remind them to be cautious because the virtual world has many of the same vulnerabilities as the real world.

4. I do envision issues to happen to my 2D game or final project but only in the case of users exploiting the source code. My game will not have any opportunity to buy or sell items but if players want to make the game more interesting or are stuck on a level they may change the source code in order for them to continue playing.

5. I think some of these issues can apply to mobile games. If people are willing to go through the time and effort to exploit and attack people through PC games then there is no reason they will not do the same for mobile games. This is especially interesting because mobile users tend to be a different segment of the population and thus would not have already been exploited. I also think that mobile games are easier to use and understand so players will be more naïve and vulnerable to attack.

✓

John Rodli

Assignment 4: Security and Privacy in Online Games

1. The main reason for vulnerability in games is the drive to create ever expanding and "open" universes where players are granted greater and greater freedoms to choose how they interact with the game and other players. This genre of game, the MMORPG, relies on countless interactions between many different clients and the game servers themselves. Minimizing exposure, while still providing a sandbox like environment, is the crucial difficulty for these games. Hence, while the technical implementation is at the end of the day the actual mechanism that prevents and creates opportunities for security breaches, it is the nontechnical fundamental movement towards greater freedom and nonlinear gameplay in the virtual realm that continually creates new or changing security concerns.
2. My favorite technical exploit is the Quicktime hack that was proven possible by researchers in the game Second Life. Second Life, like plenty of other games, relies on third party software (in this case Apple's Quicktime to display multimedia objects in game) and vulnerabilities in these third party apps can act as a gateway for hacks and exploits. The specific exploit involved embedding a malicious script inside a Quicktime multimedia object, that when happened upon by a player in the Second Life world, would be downloaded by that user's computer. This script can then gain complete access of the user's computer (due to the underlying flaw in Quicktime).
3. The basic means to reduce vulnerabilities and hacks in online games is to simply reduce exposure. What is meant by exposure is the degree to which game information is gone unmonitored (i.e – allowing P2P transfers vs. only game server interaction), and stored or processed on the client's machine. The more powerful the filters are on the game server, and on the receiving client's game will make for a more secure architecture. Effectively, don't give uncontrolled access to users to modify the game to a point where damage could be done beyond intention.

sp

4. Out 2D game is only played on a single machine, and hence many of the concerns that arise when networking are not in play. However, if we were to provide a means to play Bloodlines across multiple devices (logical as it is a turn based game that must be played by two players), then we would need to consider all of these problems (although on a lesser degree as our gameplay is very linear and allows only for a narrow level of interaction).
5. Certainly. Most of the most popular mobile games involve multiplayer interaction across servers. The massive scale of these games and their user bases poses a ripe opportunity for a clever hacker, especially as many of the more popular games involve valuable virtual wealth, items and more (if not actual credit card information itself).

Alex Schaefer
Tuesday, April 08, 2014
Comp 23
IEEE Security & Privacy Securing Online Games

The main reason for vulnerabilities and insecurities in games are most likely due to technical reasons. Most of the exploits that hackers use to play the game in ways that the developer did not intend them to can be patched and the exploits will no longer work. For example, game developers have issues with bots trying to accumulate in game virtual currency and other virtual goods that have real world monetary value. This is a technical problem. The *Server-Side Bot Detection in Massively Multiplayer Online Games* article proposed a server side solution to detecting bots in game. While bot developer may always be one step ahead of game developers trying to thwart them, the solution is a technical one.

My favorite exploit presented in the article was the Buffer Overflow that caused the game to crash. The game *Age of Conan* had an exploit that crashed the game when a script contains a line that is longer than 1024 bytes. The buffer overflows and the game crashed, giving a chance to overwrite the games stack with executable code that can be run to do anything on the client computer.

Completely eliminating all vulnerabilities from online games would be impossible. But reducing the number could be done by playing the majority of people who mod games into two categories. The first is the group of people who mod games for the enjoyment, while the second group is those who mod the gain for financial profit. The first group is easier to address, and a possible solution could be to provide a separate part of the virtual world that allows and even encourages modding of the game. The second group is harder to address. Running bot detection on the game server could be a possible solution.

Someone did this recently

Our 2D game would not likely be a target for modification as the game is a local multiplayer game. If the game were to be hacked, it would ruin the enjoyment for the other players.

Certain mobile game platforms are harder to hack than their desktop counterparts. For example, iOS does not allow the running of unsigned code, and all officially signed code is distributed through the App Store which Apple curates. It would be difficult to get code to run on an iOS mobile device that allowed a user to exploit a mobile game.

What about Android?