Security in Online Gaming

Yale (Yunling) Ren

Submitted for:
ACC 626 Spring 2012 Final Report
Professor Malik Datardina
June 28, 2012
This report seeks to learn from the security measures adopted in **massively multiplayer online games (MMOs)** (italicized + bolded terms are defined in Appendix 1 – Glossary; only italicized + bolded the first time the term appears) and apply those lessons to securing other systems that represent **facilitated networks**. The report identifies common attack methods and weaknesses exploited in MMOs. In particular, some of the current hot topics related to security issues in MMOs are examined to provide further context. The report then draws on extensive academic research and industry practice to explore effective methods of defense that have been published or implemented. Applicability of report findings is not only relevant to security in the online gaming industry, but application extends to security issues in cloud computing, social networking, and systems design. Available frameworks are examined to determine whether they can be applied to securing MMOs and other similar networks/systems as there is currently no overarching framework for this area. Lastly, recommendations are made based on findings and these recommendations are relevant to Chief Information Officers (CIOs) of companies who use systems with high user interaction, MMO providers/developers, and security professionals.

The intended audience of this report falls into three categories. First, MMO providers and developers will naturally be interested in this report, since the report establishes a business case for security in MMOs and also provides insight on securing these games. This report will also be relevant for CIOs of companies not in the booming MMO industry\(^1\). The CIOs will be interested in the key findings in the report related to how to secure different types of facilitated networks, and **peer-to-peer (P2P) networks** because many defense mechanisms used to protect MMOs can be applied in other systems with similar architectures. Security professionals will also find this report useful as it discusses specific, technical aspects of a security topic that does not see as much exposure as other topics, and they would be interested in identifying opportunities to sell their services to this relatively untapped industry.

Since there is a diverse group of intended users of this report, some technical terms will be defined in Appendix 1 – Glossary, and these terms will be italicized + bolded the first time they appear in the report.

The report is intended to capture and synthesize existing research on the topic of security in online gaming as well as to provide further insights relevant to recent developments and applications outside of gaming. Sources of research include recent publications, conference proceedings, news, books, blogs, videos, frameworks, and personal experiences of the author related to the topic. Using the most recent and up-to-date information is highly important and increases the relevance and reliability of findings and recommendations. Moreover, this report does not study security solutions great technical detail, but instead takes a high-level approach and caters presentation of the analysis to C-suite executives.
MMOs are significantly different from traditional multiplayer video games. From a system architecture perspective, traditional multiplayer games are often hosted using client-server structures, whereas modern MMOs make use of P2P networks. From a player perspective, the player interacts with thousands of players when playing a MMO, compared to no more than 10 players in most traditional games. There is even a difference from a business model perspective. Traditional games generate revenue through a one-time sale and the main cost is the development cost. On the other hand, MMOs usually generate most of their revenue through a subscription model and incur not only development costs but also significant bandwidth, processing, and storage costs.

Research in online gaming security first emerged in the early 2000s\textsuperscript{2,3} which corresponded with the increase in popularity of MMOs. During mid-2000s, the popularity of World of Warcraft (WoW) and the significant revenues Blizzard, the developer of WoW, was able to generate drew more researchers to this topic. However, even the flagship game of Blizzard and the leader of the MMO industry did not have strong security measures. The most common security concerns could be categorized into: account thefts, game exploits, botting, and illegal client-side modifications\textsuperscript{4}.

Direct implications of security weaknesses in online games have historically been threefold. First, game integrity is compromised because some players get an unfair advantage. Second, massive inflation in the in-game economy to the degree of dysfunction results from widespread botting. Third, 3rd parties offering \textit{real-money transactions (RMT)} spawn and advertise in the game, and a portion of players make use of this service even though it is usually against the End User License Agreement (EULA) or Terms of Service (TOS) of the game. These direct implications combine to deteriorate the experience for the majority of the players, and as a result players start leaving a particular game for other games. Hence, security issues cause a decline in the player base and damage a company's reputation and ultimately result in significant losses in revenue\textsuperscript{5}.

\section*{Current Issues}

Security issues in online gaming continue to make headlines. In 2011, Sony’s Playstation Network suffered a breach from hackers who obtained vital customer information including credit card details\textsuperscript{6}. Diablo III, a highly anticipated Blizzard title, experienced account thefts within days of release\textsuperscript{7}. League of Legends, the MMO currently with the most registered users\textsuperscript{8}, was the victim of a successful hacker attack and was fortunate that no billing information was accessed\textsuperscript{9}. In addition to issues that receive media attention, game providers are also constantly patching their games to fix exploitable bugs. It is evident that hackers continue to attack MMOs to obtain personal information that can provide financial gains for the hacker. Most major MMOs also suffer from the continued existence of RMT facilitators. Studies have found an emergence of a group of players who are interested in obtaining virtual wealth and translating this virtual wealth to actual wealth, rather than being interested in the fun elements of the MMOs\textsuperscript{5}.
Many opportunities to cheat exist in MMOs. The methods to cheat are comparable to attack methods used by hackers to break into other systems. Some even compare the attack surface of MMOs to that of web browsers\(^\text{10}\). Some of these attack methods are highly technical and require extensive programming knowledge, while others rely on social engineering techniques and no programming knowledge. See Appendix 2 for a summary table for various common attack/cheating methods.

Rather than examining each attack method in detail, this report categorizes the attack methods into broader groups and examines the attacks from a vulnerabilities perspective, and discusses how these vulnerabilities are exploited.

### Social Engineering Techniques

<table>
<thead>
<tr>
<th>Vulnerabilities</th>
<th>Consequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>players and users lacking awareness and being careless</td>
<td>account theft, which leads to loss of private information of individual players and to loss of virtual and real possessions; results in a privacy breach</td>
</tr>
<tr>
<td><strong>Attack characteristics</strong>: non-technical methods that exploit social weaknesses in players</td>
<td></td>
</tr>
<tr>
<td><strong>Degree of technical sophistication</strong>: Low</td>
<td></td>
</tr>
<tr>
<td><strong>Examples</strong>: email scams, malicious websites, in-game fraud, and other <strong>phishing</strong> methods</td>
<td></td>
</tr>
</tbody>
</table>

### In-game Exploits

<table>
<thead>
<tr>
<th>Vulnerabilities: bugs in game mechanics, loopholes in rules, 0-day patching flaws, unintended use of game mechanics</th>
<th>Consequence: unfair advantage to parties participating in the exploit; the advantage could come in the form of in-game possessions or benefits, reduced difficulty due to unintentional effects, or additional information; in-game economies could also be negatively affected</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Attack characteristics</strong>: identify obscure bugs and repeatedly abuse it while trying to avoid drawing attention; some bugs require collusion to exploit</td>
<td></td>
</tr>
<tr>
<td><strong>Degree of technical sophistication</strong>: Low</td>
<td></td>
</tr>
<tr>
<td><strong>Examples</strong>: exploitation of the listed vulnerabilities; form varies depending on the opportunity available</td>
<td></td>
</tr>
</tbody>
</table>

### Client-side Modifications

<table>
<thead>
<tr>
<th>Vulnerabilities: unsecured game state, lack of lockdown of third-party programs' ability to affect gameplay, over-exposure of <strong>game state</strong> to client</th>
<th>Consequence: Unfair advantage and compromise of game integrity; modified clients can also obtain information from other clients or the server that would normally be confidential or private; some modifications may remove the need for human interaction and automate game tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Attack characteristics</strong>: accesses game state information that is not normally accessible; client granted abnormal rights to change the game state; difference in client usually detectable</td>
<td></td>
</tr>
<tr>
<td><strong>Degree of technical sophistication</strong>: High</td>
<td></td>
</tr>
<tr>
<td><strong>Examples</strong>: botting and other input automation programs, altering the code on the client program, use of third-party software that affect gameplay, hardware modification such as “mod chips” in consoles</td>
<td></td>
</tr>
</tbody>
</table>

### Targeted Attacks

<table>
<thead>
<tr>
<th>Vulnerabilities: system infrastructure, bandwidth and processing constraints, low security clients</th>
<th>Consequence: depending on the nature and intent of the attack, there could be disruption of service or theft of private information; game providers also have to announce breaches which may cause severe damage to reputation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Attack characteristics</strong>: direct, technical attacks seeking specific information or disrupting service</td>
<td></td>
</tr>
<tr>
<td><strong>Degree of technical sophistication</strong>: High</td>
<td></td>
</tr>
<tr>
<td><strong>Examples</strong>: denial of service attacks targeted at servers or other clients, key-loggers, viruses, hacks intended to breach databases of game providers</td>
<td></td>
</tr>
</tbody>
</table>
Most of the common attack methods fall into one of the four broad categories above, or is a combination of multiple categories. The degree of technical sophistication for client-side modifications and targeted attacks are listed as “high”. Although this description is true for the discovery and development of the hack, the technical requirements often drop significantly once the hack is ready for use because they are often pre-packaged into executable files to be distributed. As such, the average gamer will have enough technical expertise to make use of most of the hacks they get their hands on and the security issue could spread like wildfire if not properly handled. Ironically, a lot of average users fall victims to social engineering hacks when trying to obtain other working hacks.

For attack methods to have a real impact on a system, several conditions must exist. Obviously, the method needs to be effective in what it is intended to do, but surprisingly this is not the only or even the most important condition for a high-impact hack. A more important condition is that the attack method must have a low detection rate and relatively low media exposure. A less effective method that is also less detectible can survive much longer before being patched and therefore cause more damage overall. Another key factor is the ease-of-use of the method and the ability to spread among the player base. A single player using a highly effective hack will not cause a great amount of damage on a game since his/her playing time is limited and interaction with the rest of the player population is also limited. However, if a significant portion of the player base are cheating, hacking, or exploiting, then the game quality will be deteriorated for everyone. As a particular attack method becomes more popular, it is also more likely to be detected by the game provider who will eventually fix and patch the game so that the method no longer works. The most dangerous attack methods have low detectability, high effectiveness, and are easy to use. In-game exploits fall into this category the most often.

Architectural Defense Mechanisms

Now that attack methods have been identified and categorized, the report examines defense mechanisms used by game providers and also those suggested in academic literature. Defense mechanisms will be grouped by the type of attacks they intend to defend against. Traditionally, protecting games meant piracy protection, but this is no longer the area of focus with MMOs. The first type of defense mechanisms examined is mechanisms that pertain to system design. Many researchers of security in online gaming have proposed all kinds of underlying system improvements that make the network more secure. Kabus et al.12 suggests that MMOs should utilize the P2P architecture instead of the traditional client-server architecture to design its underlying system. A P2P architecture allows for what is known as distributed state dissemination, whereby the processing of the game state is distributed to multiple clients in the P2P network, rather than being centralized on a single server. This decentralization occurs both physically and logically. Distributing the game state across multiple clients enables mutual checking, a process that requires the game states to be agreed by multiple nodes clients in the P2P network before the game can proceed. With modern day computing power, these processes
can be completed speedily without hindering performance. However, since the game state now needs to be distributed to multiple clients, malicious clients could obtain access to information that they are not permitted to if the dissemination process is not properly locked down.

Li et al.\textsuperscript{13} expands on the suggestion to use P2P networks by exploring various ways the game state can be disseminated to the clients. The drawback of an all-at-once dissemination approach, where the game state information is completely distributed to clients at the start of a processing cycle, is that single client does not need to know all the information all the time but rather only needs relevant information to make appropriate actions. Dumping excess information to clients creates latency strains and also creates opportunity for attacks and cheating. An on-demand loading scheme is suggested as a method to disseminate the game state to the clients\textsuperscript{11,14}. In this method, excess information is not given to the clients and thus making it harder for them to use client-side modifications to obtain unauthorized information. The on-demand dissemination modification to the P2P architecture helps address various client-side modification attack methods. In addition to security benefits, the on-demand dissemination process also reduces bandwidth requirements\textsuperscript{11}.

In addition to the fundamental contributions of Kabus et al.\textsuperscript{10} and Li et al.\textsuperscript{11} to the study of securing online games, several other researches later made further advancements. Kabus and Buchmann\textsuperscript{15} suggests that the correct game state be decided through a majority voting process among all clients who possess part of the game state; this method help prevent collusive attack methods. Gauthier-Dickey et al.\textsuperscript{16} proposes a “lockstep protocol” that prevent latency-related attacks and also reduces bottleneck created by excessive latency and also suppresses some client-side attacks\textsuperscript{2}. Lastly, Assiotis and Tzanov\textsuperscript{17} suggest that large virtual persistent worlds should be split into regions using a non-static division that allows for a reorganization of the division, similar to cloud computing. Mirroring schemes can be used in this situation to cope with server failures, and regional controls for each division should be setup\textsuperscript{15}.

\textbf{Non-architectural Defense Mechanisms}

Not all research about security in online gaming is based on the P2P architectural improvements. Monch et al.\textsuperscript{18} focuses their recommendations on client-side preventative measures, but still recognized that P2P architectures have advantage over traditional architectures because storing all information server-side and doing all the processing there too puts too much load on the servers and has too high requirements for bandwidth availability. Monch et al. notes that clients should be built to be unmodifiable. If the client is unmodifiable, then all of the client-side attacks can be prevented and there are no problems with distributing game state information to clients. However, this level of assurance requires both a software lockdown and a hardware lockdown, the second of which can be almost impossible to achieve for game providers. In an enterprise setting though, the hardware lockdown is very much possible. Disallowing
client-side modifications essentially solves the Byzantine Generals Problem, which is a recognized issue in network security. Other recommendations such as timed lockdowns and randomly created check-sum algorithms are also suggested as ways to lockdown a client and prevent information leaks.

Researchers have also looked at various ways to reduce the attack surface of MMOs. Bono et al. identifies that user-created content and the ability to modify the client is essential to some games, so a full client-side lockdown as suggested by others is not feasible in some cases. In these cases, a standardized scripting language for the client-side is recommended and commands that could pose security threats should be removed from the players’ development toolkit. Furthermore, thorough testing of new patches, processing engines used by the game, and in-game functions can reduce the risk of 0-day vulnerabilities when a new patch is released, changes are made to engines, or new functions are introduced. Testing essentially reduces the number of in-game exploits available and fixes the security problem before the content is released to players.

Perhaps the most difficult attack methods to secure against are the social engineering techniques because these attacks target vulnerabilities stemming from the player. It is impossible for the game provider to control all actions of the player and security holes could be created by the player’s actions outside the game. For instance, a player may not have sufficient anti-virus protection on her computer and as a result she could get viruses and her game account could be compromised or stolen. Player education is the primary defense against social engineering attacks because an educated player has much less exposed vulnerabilities compared to an uneducated player. Large game providers are now devoting a significant portion of their websites to educating players on security issues. Blizzard, for example, provides a checklist for players relating to the things they can do to make their computer and game accounts more secure. Moreover, there is also detailed information about the various phishing techniques and how to recognize them. In addition to education, stand-alone authenticators can be used as another layer of account protection and requires the player to enter a constantly changing code generated with a secure algorithm on each login. When authenticators are attached to accounts, the hackers who obtain account information through phishing or other means still cannot log into the accounts unless they also have access to the authenticator.

In addition to prevention methods, other research has also focused on the detection of attacks. A comparison of server-side versus client-side bot detection reveals that the server-side detection methods do work more effectively. Blizzard has generally used a server-side detection program called Warden on all of its recent games, and other widely used detection programs include: PunkBuster, which is used by many first-person shooter games and Valve Anti-Cheat, which is used by Valve for games released through their Steam distribution channel. All of these detection programs will keep track of third-party programs running concurrently with the games, prevent unauthorized modifications to the game client, and also sometimes prevent players from playing if anomalies are detected. Some games, where player behaviour is always transparent to other players, use a reporting system to identify players who
repeatedly violate game rules. These manual reporting systems work well in games where attack methods involve abuse of in-game rules or mechanics, but the effectiveness is reduced by collusion. Finally, logging player actions and actively reviewing log activity is another strong detection mechanism.

Detection mechanisms are important because they allow for tracking of who is cheating. Players play MMOs through registered accounts and if the game provider is able to detect cheating, then the specific accounts can be easily banned to reduce future attacks. The presence of detection mechanisms also discourage players to engage in cheating activities if they know that there is a risk for severe punishment. However, detection mechanisms need to work in conjunction with prevention mechanisms to defend against attacks and minimize negative impacts; detection mechanisms alone are not sufficient as some attacks can cause significant, irreversible damage in a short amount of time.

A summary of defense mechanisms is compiled in Appendix 3. Please note that defense mechanisms for technical targeted attacks such as database breaches and denial of service attacks are outside the scope of this report. The defense mechanisms used to defend against targeted attacks are not unique to online gaming; security experts and IT consultants can be consulted on this issue.

**Business Case for Security**

The online gaming industry establishes a business case for security through a need to maintain the integrity of games to provide high enjoyment for players. Analysts have forecast that the global MMO market will reach $8 billion USD in revenues by 2014. In 2006, outages for WoW were estimated to cost $26,198 USD per hour. More recently, a lot of game providers are also incorporating real money micro-transactions as a source of revenue. For example, the free-to-play game League of Legends relies on micro-transactions completely as their sole source of revenue, whereas Blizzard's new micro-transaction system in the form of an auction/clearing-house in Diablo 3 is a nice supplement to their revenue from game sales. In fact, since Blizzard is charging $1 USD per transaction on the Diablo 3 real money auction house. It wouldn't be surprising if the real money auction house sees millions of transactions per month, and income from micro-transactions surpasses the amount of income from game sales for Blizzard.

So why then do larger companies in other industries often have a problem forming a business case for security? Gary McGraw discusses this issue with Bruce Schneier, both security professionals, in the podcast Silver Bullet. The discussion took a political perspective and mentions that security is still a "hard sell". Security needs to be bundled with other things to sell. Bruce points out that there are two ways to sell something: incite greed or incite fear. Inciting greed is easy, but generating enough fear to sell something is much harder. The experts are on a consensus that the business case for security has improved drastically in the last decade. With the emergence of cloud computing, there should be a corresponding need for network security and security on the cloud. Larger corporations are becoming more aware of security issues because they realize that many of their business processes may soon move to the cloud, and their CIOs now have more leverage in negotiating funding for security projects.
Lessons Learned from Securing Online Games

The online gaming industry is still in its infancy, but even so the security aspect of this industry has developed a strong business case and is one of the key success factors for companies in the industry. Companies in this industry have developed and successfully implemented security measures that approximate best practices in security. Security experts and CIOs definitely have something to learn from studying security in online gaming.

Below are case studies that draw parallels between MMOs and other facilitated networks, and identifies how similar security measures can be implemented on facilitated networks. In the first case study, the vulnerabilities of the two environments are compared, with Facebook as the context for facilitated network examples.

**Case study 1: Comparison of Vulnerabilities between MMOs and Facebook**

<table>
<thead>
<tr>
<th>Vulnerabilities</th>
<th>Form in MMOs</th>
<th>Form in Facilitated Networks</th>
<th>Example in Facilitated Networks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Users</td>
<td>Players</td>
<td>Users of the network</td>
<td>Facebook users are also susceptible to phishing</td>
</tr>
<tr>
<td>Exploitable bugs</td>
<td>Bugs in game mechanics, loopholes in rules, game mechanics with unintentional effects</td>
<td>Bugs in the functionality of the network, exploitable security flaws</td>
<td>Facebook private photo exploit26</td>
</tr>
<tr>
<td>Client-side exposure and modifiability</td>
<td>Game clients, hardware used to play the game, 3rd party programs</td>
<td>Client on which the network is accessed, web browsers, 3rd-party programs</td>
<td>Chrome allows for extensions to be installed, so of which may be able to pull protected data when browsing Facebook</td>
</tr>
<tr>
<td>Infrastructure</td>
<td>Game servers, other clients, bandwidth</td>
<td>Host servers, other users of the network</td>
<td>Facebook databases can be breached directly and information can be leaked</td>
</tr>
</tbody>
</table>

As evident in this comparison, the vulnerabilities that existed in MMOs are also existent in facilitated networks in different forms. Therefore, the same attack methods used in MMOs can be used to target all of the vulnerabilities in facilitated networks as well. Hackers may also have much more to gain from hacking facilitated networks other than MMOs because they can obtain personal information and real possessions rather than virtual possessions.

Since the attack methods are similar to those employed in hacking MMOs, the defense mechanisms used to secure online games can then also be used to secure facilitated networks. The following is another case study that identifies hypothetical security weaknesses related to EBay and examines whether there is an appropriate defense mechanism from online gaming security that would be applicable and effective.
Case Study 2: Applicability of Security Measures used in Online Gaming to Potential EBay Issues

<table>
<thead>
<tr>
<th>Hypothetical EBay Security Issue</th>
<th>Applicable Solution from Securing MMOs</th>
</tr>
</thead>
<tbody>
<tr>
<td>EBay user is unaware of ongoing email scam that trick users to give up account information</td>
<td>Educate the users through detailed informational blogs on the website and implement multiple layers of authentication, including the use of authenticators.</td>
</tr>
<tr>
<td>EBay user is defrauded by another user because the item purchased did not match with the description or pictures of the item</td>
<td>Review system that allows for user feedback and complaints; people with excessive negative ratings should be banned from selling items (this kind of mechanism is already in place on EBay).</td>
</tr>
<tr>
<td>EBay users are colluding to drive up the bidding price of an item they are selling</td>
<td>Log all user activity and actively review logs to identify suspicious activity; ban users who engage in activities that violate the ToS.</td>
</tr>
<tr>
<td>EBay user creates a custom extension to her web browser that captures payment information of other users bidding on her items; this information is normally hidden and inaccessible until the bidding time expires</td>
<td>Only allow approved extensions to run when users are on the EBay website, or disallow extensions/add-ons completely; could also not disseminate the payment information of bidders until the winning bid is determined when time expires</td>
</tr>
<tr>
<td>EBay user targeting specific item waits until there are 5 minutes left on the item, and then performs a DOS attack on EBay servers to crash the website; no other users are able to bid on this item</td>
<td>Hire IT consultants to build defenses against DOS attacks; consider limiting the amount of packets that can be sent across the network from a particular client to other clients/servers that it is interacting with during a short period of time</td>
</tr>
</tbody>
</table>

The second case study demonstrates a strong applicability of defense mechanisms used in securing online games to other facilitated networks. It is still very important to always tailor security solutions to a specific problem and security measures imported from MMOs should only serve as a guideline.

Usefulness of security measures from online games is illustrated by the above cases analyses in the context of facilitated networks. The applicability of security concepts from online gaming is strongest in other facilitated networks because MMOs are facilitated networks themselves. The usefulness of studying security of online games is not limited to facilitated networks, however. There is also relevance in securing cloud storage, applications, and platforms.

Other Tools and Frameworks

There are several tools and frameworks available consisting of security and governance frameworks and legal cases. Due to the infancy of the online gaming industry, it is no evident that there is widespread use of these tools and frameworks by game providers. On the other hand, these frameworks will be very familiar to CIOs and security experts in more developed industries.

The first available framework is ISO/IEC 27001 – Information technology – Security techniques – Information security management systems – Requirements. This framework outlines a management system that explicitly controls information security, and sets out specific requirements to be met by enterprises. ISO 27001 requires not only the existence of security controls, but also mandates the organization to examine information security risks from the perspectives of threats, vulnerabilities, and impacts. Furthermore, in order to keep up with changing security needs, a management monitoring
process should also be in place. A certificate can be obtained to ensure that a specific part of the business, such as the facilitated network itself, complies with ISO 27001 requirements. Other frameworks in the ISO 27000 family, such as ISO 27002 to ISO 27006, can be used to help management implement systems and processes that would meet ISO 27001 standards.

Another framework that could provide guidance is COBIT, which is a framework for IT governance and control developed by ISACA. Contrast to ISO 27001, COBIT focuses more on the governance of IT and is targeted towards executive management. COBIT helps organizations align their IT projects with their strategy to derive value from IT, and also provides tools for risk management, resource management, and performance measurement related to IT investments.

A third applicable framework is the Business Model for Information Security (BMIS) framework also developed by ISACA. It provides a model for a business-oriented approach to managing information security. The model, similar to COBIT, also does not focus on technological aspects but rather on security and management processes that connect the organization’s strategy, people, and IT systems.

Lastly, enterprises also have legal tools available to combat security threats. Again, the legal issues related to MMOs and virtual goods are not very developed, but there have been court decisions that will set precedence for the future. Legal issues surrounding security are more developed in the rest of the business world and enterprises can definitely resort to legal means to shut down hackers, although this may not be the most cost-efficient method to do so.

Conclusions

Security in online gaming is more developed than security in some larger industries. As the online gaming industry experiences rapid growth, game providers are finding that security is becoming a key contributing factor to their success.

Attack methods generally exploit four categories of vulnerabilities: users, in-game bugs, weakly secured clients, and system infrastructure weaknesses. Hackers exploit these weaknesses using social engineering techniques, individual and collusive abusive of bugs, modifications to the client, and targeted technical attacks.

Defense mechanisms generally suggest that a P2P network architecture is preferred to a traditional client-server architecture. However, a baseline P2P architecture has many security flaws and several improvements should be made, including the use of mutual checking procedures such as the lockstep protocol, on-demand dissemination of information rather than all-at-once, and flexible divisions of the whole network into smaller pieces. Besides architectural defense mechanisms, there are also other mechanisms that focus on detection and punishment of hackers and cheaters. Furthermore, implementing some degree of lockdown on the client is also recommended to prevent unauthorized access to information. The importance of educating the user and providing multiple layers of authentications is also noted.

For CIOs and security experts not particularly interested in the online gaming industry, there are plenty of takeaways from security in online games. Case studies comparing MMOs and facilitated networks find that the two environments have very similar vulnerabilities and as such hackers can use similar attacks. The report also finds that defense mechanisms that work in an online gaming setting can also work very effectively in facilitated network settings. Lastly, some further tools and frameworks, including ISO 27001, COBIT, BMIS, and legal cases, are provided to help CIOs and security professionals implement some of these measures.
Endnotes

7 Yin-Poole, Wesley. "Diablo 3 accounts hacked, gold and items stolen." Eurogamer. May 21, 2012.
20 Blizzard Entertainment. "Account Security".
Appendix 1 – Glossary

**Botting**: the use of peripheral 3rd party software to repeat a set of actions automatically to perform repetitive tasks. The requirement of human input is eliminated when a bot is used to play the game.

**Byzantine Generals Problem**: a conceptual problem presented to address issues surrounding distributed networks. The problem is laid out as follows:

- Several divisions of the Byzantine army are sieging a city
- Each division has a general who can only communicate with other generals through messengers
- Some generals may be traitors who are allied with the besieged city
- The generals must come to a consensus on the plan of attack
- Need algorithm to do the following:
  - All loyal generals must decide on the same plan of action
  - A small number of traitors cannot cause loyal generals to adopt a bad plan

**Client-server structure**: a system architecture where a computer (node) on the system is either a server or client. Workload is centralized on a server and the clients rely on the servers for resources and processing power.

**Dead reckoning**: a technique that compensates for variable communication latency and loss across a network by allowing a host to guess the state of another player when updates are missing based on the last known actions.

**Facilitated network**: a business model and systems architecture that connects similar-minded users and allows for high amounts of user interaction with other users across a platform. Facebook is a prominent example of this architecture; other examples include Google Search, Ebay, and Twitter.

**Game state**: the set of information that contains the current realization of the game, which could include player positions, player possessions, past actions, and actions yet to be executed.

**Massively Multiplayer Online Games (MMOs)**: a video game that is played online by hundreds of thousands of concurrent players in one or multiple persistent world(s).

**Peer-to-peer (P2P) network**: a network architecture where each computer (node) in the network can act as a server or client. Workload is split among members of the network instead of being centralized on a fixed server.

**Phishing**: social engineering techniques to acquire information by masquerading as a trustworthy party.

**Real-money Transactions (RMT)**: transactions where real life currency is exchanged for in-game currency or other in-game benefits.
## Appendix 2 – Classification of types of cheating and attack methods

<table>
<thead>
<tr>
<th>Classification of cheating types and attack methods</th>
<th>Vulnerabilities</th>
<th>Consequences</th>
<th>Exploiter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Underlying system</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>Game system</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>Fairness violation</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>Authentication violation</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>Integrity violation</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>Service denial</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>Theft</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>Independent</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>Cooperative</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
</tbody>
</table>

Collusive abuse of game rules
Abuse of game procedures
Manipulation of virtual assets
Exploitation of artificial intelligence
Modification of client structure
Denial of service
Manipulation of timing of packets and latency
Compromise of passwords
Exploitation of lack of secrecy
Exploitation of lack of authentication
Exploitation of a bug or loophole in game design
Compromise of game servers
Social engineering techniques

This table is a modification of a classification compiled by Jeff Yan., “A Systematic Classification of Cheating in Online Games.” Proceedings of 4th ACM SIGCOMM workshop (2005).
<table>
<thead>
<tr>
<th>Attack Methods</th>
<th>High-level Description of Defense Mechanism</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collusive abuse of game rules</td>
<td>Cut off communication channels or maintain anonymity when player identification is not required</td>
</tr>
<tr>
<td>Abuse of game procedures</td>
<td>Implement countermeasures to discourage abuse, such as banning cheaters</td>
</tr>
<tr>
<td>Manipulation of virtual assets</td>
<td>Make use of digital signatures, detailed logging, authorized trading system</td>
</tr>
<tr>
<td>Exploitation of artificial intelligence</td>
<td>Reduce reliance on AI, patch bugs in AI behaviour</td>
</tr>
<tr>
<td>Modification of client structure</td>
<td>Lockdown client, prescribe scripting language for modifications, authorize modifications</td>
</tr>
<tr>
<td>Denial of service</td>
<td>Outside the scope of this report. Use general prevention methods specifically related to DOS attacks such as hiding player IPs</td>
</tr>
<tr>
<td>Manipulation of timing of packets and latency</td>
<td>Implement lockstep protocol during synchronization, do not use client-side clock</td>
</tr>
<tr>
<td>Compromise of passwords</td>
<td>Enforce the use of authenticators, require stronger passwords, educate players</td>
</tr>
<tr>
<td>Exploitation of lack of secrecy</td>
<td>Encrypt hidden information or do not pass unnecessary information to the client</td>
</tr>
<tr>
<td>Exploitation of lack of authentication</td>
<td>Use authenticators, implement measures to automatically log players out if idle</td>
</tr>
<tr>
<td>Exploitation of a bug or loophole in game design</td>
<td>Test thoroughly before release, implement countermeasures to discourage exploitation</td>
</tr>
<tr>
<td>Compromise of game servers</td>
<td>Mirroring game state in P2P networks</td>
</tr>
<tr>
<td>Social engineering techniques</td>
<td>Educate players and use authenticators</td>
</tr>
</tbody>
</table>
### ANNOTATED BIBLIOGRAPHY

#### Cheating Methods

<table>
<thead>
<tr>
<th>Author</th>
<th>Title of Article</th>
<th>Periodical/website</th>
<th>Vol. / No. / Edition</th>
<th>Pub. year</th>
<th>Pages</th>
<th>Date accessed</th>
<th>Location, data base, website, link</th>
<th>Added in the final paper?</th>
</tr>
</thead>
</table>

**Annotation**

- **RMT =** real money transactions
- At the date of this blog, RMTs were not allowed in the EULAs of most major MMORPGs (i.e. WoW)
  - Blizzard has won injunctions against “gold farmers”
    - However, author notes that Blizzard have no power over those operations located outside the US
- RMTs end up being a huge business; one major provider of MMORPG currency states that they are worth $220M USD;
  - These companies usually use trial accounts to spam and advertise their services
    - MMOs have restricted communications of trial accounts as a result
  - Companies are hacking accounts to gain access to full accounts without paying
    - These accounts can be used to advertise or to farm gold
- Companies are also stealing credit card information to use in other MMOs or to create new accounts
- The demand for RMTs is created by players who do not have time to farm themselves (opportunity cost of farming is too high for some players)
- Author suggests that MMOs should offer their own authorized RMT system to out-price external RMTs

<table>
<thead>
<tr>
<th>Author</th>
<th>Title of Article</th>
<th>Periodical/website</th>
<th>Vol. / No. / Edition</th>
<th>Pub. year</th>
<th>Pages</th>
<th>Date accessed</th>
<th>Location, data base, website, link</th>
<th>Added in the final paper?</th>
</tr>
</thead>
</table>

**Annotation**

- **RTS =** real time strategy
- Suggest to replace client-side game simulations with server-side game simulations
  - Only visible parts of the game state is sent to the clients
    - Nullifies client-side hacking
- There may be performance issues with server-side simulations
- An important component of RTS is incomplete information of players (i.e. fog of war)
- Client-side game simulation model:
  - the server only acts as a facilitator to connect multiple client and send communication such as player actions across clients
  - games are not hosted/simulated on the server
- Client-side hacking leads to player turnover (unfair play leads to player dissatisfaction)
- If too much information is stored client-side, then users can modify clients to manipulate information and reveal hidden information; this results in complete information for one side and incomplete information for others and unfair gameplay

---

**Author** | **Title of Article** | **Periodical/website** | **Vol. / No. / Edition** | **Pub. year** | **Pages** | **Date accessed** | **Location, database, website, link** | **Added in the final paper?**
--- | --- | --- | --- | --- | --- | --- | --- | ---
Jeff Yan, Brian Randell | A Systematic Classification of Cheating in Online Games | ACM | Proceedings of 4th ACM SIGCOMM workshop | 2005 | | May 18, 2011 | ACM Digital Library | 

**Annotation**

- Paper summarizes the various methods of cheating in online games
- Cheating often arise due to poor or non-existent security designs in the systems
- Online cheating often extends beyond attacks and intrusions, which are major security threats in other systems
- Types of cheating:
  - Tampering or modification of game client code
  - Collusion (i.e. win-trading in games with ladder-ranking systems)
  - Exploiting game procedures (i.e. disconnecting to prevent loss)
  - Unfair advantage acquired through RMTs when RMTs are forbidden by EULA
  - Exploiting AI behaviour patterns
  - Client-side modifications that do not involve the game client (i.e. modifying graphics drivers to make walls transparent)
  - Denying service to peer players (i.e. DOS attacks on other players)
  - Timing cheating (i.e. purposely dropping packets to manipulate timing of actions)
  - Account theft and compromising passwords
  - Exploiting lack of secrecy (i.e. eavesdropping on and modifying packets)
  - Exploiting lack of authentication (another form of account theft due to lack of server-side authentication when multiple users connect from the same machine)
  - Exploiting a bug in game mechanic (no technical manipulation)
  - Compromise game servers
- Internal misuse (i.e. insider help)
  - Social engineering (i.e. email scans pretending to be Blizzard to get account info)
- Provides a table that categorizes the above list of cheats by vulnerability, possible failures, and exploiters
- Vulnerabilities
  - People (player, opponents)
  - System design inadequacies (in-game system, underlying system)
- Possible failures
  - Theft of information or possessions
  - Service denial
  - Integrity violation
  - Masquerade (change of identity)
  - Fairness violation
- Exploiters
  - Independent (single player, opponent)
  - Cooperative (multiple players, operator and players)

<table>
<thead>
<tr>
<th>Author</th>
<th>Title of Article</th>
<th>Periodical/website</th>
<th>Vol. / No. / Edition</th>
<th>Pub. year</th>
<th>Pages</th>
<th>Date accessed</th>
<th>Location, data base, link</th>
<th>Added in the final paper?</th>
</tr>
</thead>
</table>

Annotation
- Lack of security results in loss of significant revenues
- Some hackers have made a living from hacking MMOs for profit
- Emergence of a group of players who are more interested in obtaining virtual wealth (which could be translated to actual wealth) rather than playing the game for enjoyment
  - Entire industry of sweatshops with hundreds of thousands of Chinese workers
- Race condition: occurs when an assumption needs to hold true for a period of time, but might actually not
  - Every race condition has a window of vulnerability, that is a time during which violating the assumption will lead to incorrect behaviour
- Timing and synchronization problems are often exploited by hackers in distributed networks, in MMOs and also in real world software
- The biggest challenge for the distributed architecture of MMOs is that the game state must be shared with hundreds of thousands of concurrent users
- Most MMOs do not only have a single persistent world, but instead present copies of the world (i.e. realms in WoW) in order to reduce processing requirements and reduce delay in updates
  - There are player limits on each copy (i.e. about 50k concurrent users per realm in WoW)
- Most problems arise in the boundaries of each copy of the world, especially if players across different worlds can interact with each other; race conditions arise in these areas
Manipulating or terminating updates from the client near boundaries can result in updates being applied in more than 1 world (i.e. item duplication)

- Telehacking: manipulating coordinate information stored client-side to teleport rather than move in conventional ways
- These problems prevalent in MMOs can also be seen in other types of software with similar structures that result in trust model boundaries being confused (i.e. Google Desktop in 2007)

<table>
<thead>
<tr>
<th>Author</th>
<th>Title of Article</th>
<th>Periodical/website</th>
<th>Vol. / No. / Edition</th>
<th>Pub. year</th>
<th>Pages</th>
<th>Date accessed</th>
<th>Location, data base, website, link</th>
<th>Added in the final paper?</th>
</tr>
</thead>
</table>

Annotation

- Studies show that games are fun, but cheating and hacking activities in games are also fun for players who engage in such activities
- Cheating also driven by potential gain from virtual economies
- Cheating is often a breach of the EULA or TOS but has no other severe implications

Preventative Recommendations

<table>
<thead>
<tr>
<th>Author</th>
<th>Title of Article</th>
<th>Periodical/website</th>
<th>Vol. / No. / Edition</th>
<th>Pub. year</th>
<th>Pages</th>
<th>Date accessed</th>
<th>Location, data base, website, link</th>
<th>Added in the final paper?</th>
</tr>
</thead>
</table>

Annotation

- When distributed Peer-to-peer (P2P) techniques are used to reduce server load in MMOs, control over the game may fall into the hands of the clients
- Almost MMO has a client-server architecture
- Cheating can destroy in-game economies through item duplication or inflation
- Honest players are often aware of cheaters and either stop playing or cheat themselves
- In client-server (C-S) architecture, the server has control over almost all game information, which
makes it theoretically impossible to hack the game
- Design flaws can still leave room for successful cheating
- Griefer: players with the sole intention of hurting the playing experience of other players
  o May be performed within game rules, if the rules are loose in this area
  o However, griefers often also cheat
- Offers P2P approaches to replace traditional C-S architectures
  o Distributed state dissemination: saves bandwidth but messages across clients could be manipulated or disrupted
  o Mutual checking: game states must be agreed on by multiple clients
  o Log auditing: focuses on detection and punishment of cheaters rather than real-time prevention of cheating
  o Trusted computing: currently not used by the MMO industry
- MMOGs must devote resources to security in order to maintain high enjoyment of players

<table>
<thead>
<tr>
<th>Author</th>
<th>Title of Article</th>
<th>Periodical/website</th>
<th>Vol. / No. / Edition</th>
<th>Pub. year</th>
<th>Pages</th>
<th>Date accessed</th>
<th>Location, data base, website, link</th>
<th>Added in the final paper?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kang Li, Shanshan Ding, Doug McCreary, Steve Webb</td>
<td>Analysis of State Exposure Control to Prevent Cheating in Online Games</td>
<td>ACM</td>
<td>NOSSDAV ’04 Proceedings of the 14th international workshop</td>
<td>2004</td>
<td>140-145</td>
<td>May 18, 2012</td>
<td>ACM Digital Library</td>
<td></td>
</tr>
</tbody>
</table>

**Annotation**

- One common type of cheating is players accessing and modifying the game state that is exposed to their game client
- Knowing state information that is normally not accessible (full maps, positions of players) can provide advantages for a player
- On-demand (rather than all-at-once) dissemination of game state information is suggested
- A client only needs to know certain information at certain times to play; no need to give out all information all the time
- State dissemination strategies compared
  o Eager loading: game state is loaded on the client side at the start of the game; updates are done client side based on signals from the server
  o On-demand loading: only necessary information is loaded to the client
  o On-demand preloading: server predicts the information that will be needed by the client in a short period of time and sends that information before the client requests for it
- On-demand loading and preloading reduces bandwidth requirements but increases server CPU load
- On-demand loading introduces more delay to game players, but the delay can be reduced by preloading
<table>
<thead>
<tr>
<th>Author</th>
<th>Title of Article</th>
<th>Periodical/website</th>
<th>Vol. / No. / Edition</th>
<th>Pub. year</th>
<th>Pages</th>
<th>Date accessed</th>
<th>Location, data base, website, link</th>
<th>Added in the final paper?</th>
</tr>
</thead>
</table>

**Annotation**

- Moving from centralized server-client architecture to distributed structures complicates player interaction (from a technical perspective) and increases the potential for cheating
- Little or no real security to prevent cheating in most online games
- Dead reckoning: a technique that compensates for variable communication latency and loss across a network by allowing a host to guess the state of another player when updates are missing based on the last known actions
- Cheating is still possible in a dead reckoning system
- Even RTS games can be broken down into turns where one turn could be one unit of time in the game (one game second, which could be the same or different from an actual second)
-Suppress-correct cheat: host gains advantage in a distributed system by purposely dropping packets and synchronize the game state using manipulated but plausible packets;
- Look-ahead cheat: allows for reactions that are normally not possible using human reflex times; detects the actions of other players that were sent to the host and making a reaction based on those actions of other players
- Lockstep protocol suggested to combat both suppress-correct and look-ahead cheat
  - More secure way of announcing turn actions using cryptography and hashes
- Asynchronous synchronization: turn clock is decentralized to clients and synchronization only occurs when interaction is required; still follows the lockstep protocol
  - Can work even where there is packet loss

- Often MMOs use distributed P2P systems to distribute game updates
- Cheating puts persistent worlds at risk
- Cheating prevention measures should be built into the system, not added on later
- Basic principle to address arbitrary manipulation of game data is to replicate the game state among multiple nodes in the network instead of placing trust on a single one or a small group of nodes
- Suggests that replicas of game states to be managed by super-peers run by game providers
- Suggests that the correct game state be determined through majority voting
  - Dead reckoning and pre-reckoning techniques mentioned here also
- Partitioning the game into regions is easy as long as there is no interaction between regions (early WoW did this with different servers and battleground queue groups), but often MMOs require cross-region interaction (current WoW)
  - Also applies to instances of certain areas that are not exposed to the rest of the game world
- Regional controllers (RCs): nodes in the distributed network that are responsible for managing the game state for a region
- Describes how operations of the game should function without cheaters, such as logging into the game and actions when playing the game

<table>
<thead>
<tr>
<th>Author</th>
<th>Title of Article</th>
<th>Periodical/ website</th>
<th>Vol. / No. / Edition</th>
<th>Pub. year</th>
<th>Pages</th>
<th>Date accessed</th>
<th>Location, data base, link</th>
<th>Added in the final paper?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Daniel Zappala,</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Virginia Lo,</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>James Marr</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Annotation

- Other suggested event ordering protocol operated at the latency of the slowest player
- The low-latency event ordering protocol, referred to by authors as New Event Ordering (NEO), operates independent of network conditions and adapts to congestion
- NEO drastically improves performance without compromising trust
- Types of cheating related to abusing network protocol & how NEO addresses them
  - Fixed delay cheat: delay is added to every outgoing packet; gives player using the cheat to first see the other player’s updates before sending own packets
    - Addressed using bounded round lengths; late updates are ignored
  - Timestamp cheat: because events are ordered for processing and consistency checks, players could timestamp their own updates that is inconsistent with the global clock
    - Again addressed by bounded round lengths; client will not be able to decrypt and reapply the timestamp in time
  - Suppressed update cheat: a player suppresses all updates of own actions to other players but continues to receive theirs; periodic updates are sent before to prevent dropping out of the game
    - Missing packets from one player is interpreted as congestion; packets from other players will not be sent to that player until updates from that player resumes
- Inconsistency cheat: cheater sends different updates to different players of the game
  - NEO uses digital signatures and periodic audit of game states
- Collusion cheat: sharing packets and information not meant to be shared to gain advantage
  - Majority voting level can be adjusted to ensure that colluders cannot become majority
  - Prevent a single player from logging into multiple accounts to achieve artificial majority
- Lockstep protocol has a play-out latency of at least 3 times the slowest latency between any 2 nodes
  - NEO play-out is limited by the round length which is independent of network congestion levels or individual player bandwidth limitations

<table>
<thead>
<tr>
<th>Author</th>
<th>Title of Article</th>
<th>Periodical/website</th>
<th>Vol. / No. / Edition</th>
<th>Pub. year</th>
<th>Pages</th>
<th>Date accessed</th>
<th>Location, database, website, link</th>
<th>Added in the final paper?</th>
</tr>
</thead>
</table>

**Annotation**

- Maphacking: information exposure that reveals the opponents units and positions when they should be hidden (removal of fog of war)
  - Fog of war, summarized here: player A cannot see player B’s unit x unless a unit controlled by player A observes x. Each unit has a scouting radius and any enemy unit within this radius is revealed to the player. The player’s vision is comprised of the union of the vision of each of his units, and everything outside of that area is in the fog of war
- Author suggests that detection of cheating for online games is sufficient as users are tied to accounts which are in turn tied to purchased copies of the game or subscriptions
  - Suggests that information is not given to the client all at once, but only when needed
- In the context of RTS games, information only given for objects and units within a player’s viewable area

<table>
<thead>
<tr>
<th>Author</th>
<th>Title of Article</th>
<th>Periodical/website</th>
<th>Vol. / No. / Edition</th>
<th>Pub. year</th>
<th>Pages</th>
<th>Date accessed</th>
<th>Location, database, website, link</th>
<th>Added in the final paper?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Christian</td>
<td>Protecting Online</td>
<td>ACM</td>
<td>Proceedings of 5th ACM SIGCOMM</td>
<td>2006</td>
<td></td>
<td>May 18,</td>
<td>ACM Digital</td>
<td></td>
</tr>
</tbody>
</table>
### Annotation

- Focuses on client-side preventative measures rather than architectural recommendations
- Storing all information server-side and doing all the processing there puts too much load on the servers and has too high requirements for bandwidth availability
- Suggests that clients should be unmodifiable
  - Prevent cheating through modification of game client
  - Server can safely distribute processing to clients without fear of loss of information integrity
  - Removes the Byzantine generals problem
- Byzantine generals problem: an agreement problem when synchronizing distributed states back onto the master state
- Mobile guard concept: information on the client side are protected with a guard mechanism with a timer; the guard mechanism is removed when the timer expires, and replace with a new mobile guard if the information still needs to be protected
  - Use enforcement algorithms to ensure that the mobile guard is executed on the client
  - RCCAs used as enforcement algorithms
- Randomly created checksum algorithms (RCCAs) can be used to maintain information integrity
- Details application to different types of game configurations

---

### Author

<table>
<thead>
<tr>
<th>Author</th>
<th>Title of Article</th>
<th>Periodical/ website</th>
<th>Vol. / No. / Edition</th>
<th>Pub. year</th>
<th>Pages</th>
<th>Date accessed</th>
<th>Location, data base, link</th>
<th>Added in the final paper?</th>
</tr>
</thead>
</table>

### Annotation

- Hackers could use security loopholes in games or social engineering techniques related to gaming to access much more than just a person’s account in a game
- 0-day exploits are also common in games
- MMOs usually have a large attack surface, like web browsers
- Attack surfaces include:
- Client-server communication
- Modifications to game client and 3rd party plug-ins
- Numerous file formats
- Additional delivery mechanisms
- P2P communication

<table>
<thead>
<tr>
<th>Author</th>
<th>Title of Article</th>
<th>Periodical/ website</th>
<th>Vol. / No. / Edition</th>
<th>Pub. year</th>
<th>Pages</th>
<th>Date accessed</th>
<th>Location, data base, website, link</th>
<th>Added in the final paper?</th>
</tr>
</thead>
</table>

Annotation
- Security concern have often revolved around piracy protection, but not less applicable to subscription-based online games
- Cheating in single-player games do not matter, but cheating in online games affects other users and does matter
- Player retention is important for online games
- Focuses on examining collusion with the game of Bridge as context
- Suggests to keep communication channels open have collaborative monitoring (to detect cheating and discourage cheating)
- Also suggests to aggressively close covert communication channels in game that can facilitate cheating

<table>
<thead>
<tr>
<th>Author</th>
<th>Title of Article</th>
<th>Periodical/ website</th>
<th>Vol. / No. / Edition</th>
<th>Pub. year</th>
<th>Pages</th>
<th>Date accessed</th>
<th>Location, data base, website, link</th>
<th>Added in the final paper?</th>
</tr>
</thead>
</table>

Annotation
- Paper outlines cheating methods and recommendations to address them
- Cheating by collision can be addressed by cutting off communication channels or maintain high
- Abusing procedures or policy (i.e. disconnecting when about to lose)
  - Can implement countermeasures to discourage players from disconnecting (such as banning players who regularly disconnect from playing for periods of time)
- There are scams with RMTs where one party fails to delivery; very hard to track down and recover lost assets because the activity is not endorsed by the game provider and also due to online anonymity
  - Digital signatures, detailed logging, and authorized trade systems can help with this issue
- Compromised passwords can be prevented through stronger password requirements and educating players about potential password scams
- Denial of service attacks
  - Hide IP information from other players
- Cheating due to lack of communication secrecy
  - Encrypt
- Cheating due to lack of authentication
  - Re-authenticate before password changes
- Cheating by social engineering
  - Educate players about fraud and scam schemes
- General recommendations to prevent, detect, and manage cheating
  - Built-in cheating detection
  - Increase player security awareness
  - Good password practice and management
  - Implement fair trading systems in-game
  - Bug patching
  - Active complaint-response channel
  - Logging and audit trail
  - Post-detection mechanisms

### Annotation
- Botting: a form of cheating in which players use a program to play the game with minimal human interaction
- WoW uses an automatic bot-detection program called the Warden, which is client-side
- Bots often used to perform repetitive tasks with no input from the player
- Related work in bots also applies to website security (i.e. CAPTCHAs)
- Bots have negative impact on the game, including inflation
- Detection starts with extracting player movements and actions and detection repetitions that would
Bots usually take the exact same path every time, and perform the exact same actions on intervals. Server-side bot detection works better because the player does not know it is working and cannot make server-side changes to get around it. There are still ways to evade:

- Very long paths
- Bots that randomize unimportant actions or randomly deviate from path in order to avoid detection

- Split large virtual worlds into smaller regions, where each region is handled by a different server
- The suggested architecture addresses the following problems:
  - Bandwidth
  - Consistency, hotspots, congestion, and server failure
  - Cross-region (cross-server) player interaction
- Not a static division (server A always handles region A), but a design that allows for a reorganization of the division (similar to cloud)
  - Handles hotspots and load spikes better
- Outages for WoW in 2006 cost an estimated $26,198 per hour
- Mirroring schemes used to cope with server failure
Annotation
- Paper provides definitions for key terms relating to security in computing
- Will use this article to reference key definitions, some include:
  - System boundary
  - Total state, external state, internal state
- Also contains diagrams illustrating many different types of failures

<table>
<thead>
<tr>
<th>Author</th>
<th>Title of Article</th>
<th>Periodical/website</th>
<th>Vol. / No. / Edition</th>
<th>Pub. year</th>
<th>Pages</th>
<th>Date accessed</th>
<th>Location, data base, website, link</th>
<th>Added in the final paper?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brian Randell, Carl Landwehr</td>
<td>Computing Computing</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Annotation
- A conceptual problem presented to address issues surrounding distributed networks
  - The problem:
    - Several divisions of the Byzantine army are sieging a city
    - Each division has a general who can only communicate with other generals through messengers
    - Some generals may be traitors, allied with the besieged city
    - The generals must come to a consensus on the plan of attack
    - Need algorithm to do the following:
      - All loyal generals must decide on the same plan of action
      - A small number of traitors cannot cause loyal generals to adopt a bad plan

<table>
<thead>
<tr>
<th>Author</th>
<th>Title of Article</th>
<th>Periodical/website</th>
<th>Vol. / No. / Edition</th>
<th>Pub. year</th>
<th>Pages</th>
<th>Date accessed</th>
<th>Location, data base, website, link</th>
<th>Added in the final paper?</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Author</th>
<th>Title of Article</th>
<th>Periodical/website</th>
<th>Vol. / No. / Edition</th>
<th>Pub. year</th>
<th>Pages</th>
<th>Date accessed</th>
<th>Location, data base, website, link</th>
<th>Added in the final paper?</th>
</tr>
</thead>
</table>
Annotations

- Legal issues inherent in virtual works and MMORPGs are extremely complex and still in their infancy
- In China and Australia, virtual goods are subject to taxation
- *Marc Bragg v. Linden Research, Inc.* is a case related to disputes over virtual assets in Second Life; this case is the first US precedent case on virtual worlds; concluded that the EULA is the governing document for virtual worlds
- RMTs make a major sub-economy of MMOs
- EULA often prohibits RMTs
- It can be unclear who (player or game provider) can file lawsuits against RMT vendors
- Some game providers have gone beyond the EULA and accused RMT vendors of illegal activity, such as spamming (i.e. *Blizzard Entertainment, Inc. v. In Game Dollar, LLC*)
- In accepting EULA, players often waive significant individual rights

<table>
<thead>
<tr>
<th>Author</th>
<th>Title of Article</th>
<th>Periodical/website</th>
<th>Vol. / No. / Edition</th>
<th>Pub. year</th>
<th>Pages</th>
<th>Date accessed</th>
<th>Date accessed</th>
<th>Location, data base, website, link</th>
<th>Added in the final paper?</th>
</tr>
</thead>
</table>

Annotation

- Forecasted $6.1B revenue for MMOs just in China
- Chinese gamers are more likely to spend money on MMOs than any other type of game
- This article is an indication of strong growth and already large market for MMOs
- Growth around the world is not as fast as China, but still growing
- Growth in other parts of the world are more dependent on release of new games

<table>
<thead>
<tr>
<th>Author</th>
<th>Title of Article</th>
<th>Periodical/website</th>
<th>Vol. / Edition</th>
<th>Pub. year</th>
<th>Pages</th>
<th>Date accessed</th>
<th>Date accessed</th>
<th>Location, data base, website, link</th>
<th>Added in the final paper?</th>
</tr>
</thead>
</table>
### Info at Risk hacked/

<table>
<thead>
<tr>
<th>Annotation</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>- Information for 70M users on the Playstation Network (PSN) was compromised</td>
<td></td>
</tr>
<tr>
<td>- Hackers got access to customer's vital information, including credit card information</td>
<td></td>
</tr>
<tr>
<td>- PSN was temporarily shut down and security firm was hired to investigate</td>
<td></td>
</tr>
<tr>
<td>- Significant damage to PSN reputation, loss of customers, and down time all contributed to loss of revenue</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Author</th>
<th>Title of Article</th>
<th>Periodical/website</th>
<th>Vol. / No. / Edition</th>
<th>Pub. year</th>
<th>Pages</th>
<th>Date accessed</th>
<th>Location, data base, website, link</th>
<th>Added in the final paper?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yin-Poole, Wesley</td>
<td>Diablo 3 accounts hacked, gold and items stolen</td>
<td>Eurogamer</td>
<td></td>
<td>May 21, 2012</td>
<td></td>
<td>June 1, 2012</td>
<td><a href="http://www.eurogamer.net/articles/2012-05-21-diablo-3-accounts-hacked-gold-and-items-stolen">http://www.eurogamer.net/articles/2012-05-21-diablo-3-accounts-hacked-gold-and-items-stolen</a></td>
<td>YES</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Annotation</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>- Players have been forced off their accounts due to logins from another location</td>
<td></td>
</tr>
<tr>
<td>- Players have been left with nothing when logging back onto their account</td>
<td></td>
</tr>
<tr>
<td>- Blizzard has offered roll-backs to some but not all</td>
<td></td>
</tr>
<tr>
<td>- This is a widespread issue for those who do not have authenticators</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Author</th>
<th>Title of Article</th>
<th>Periodical/website</th>
<th>Vol. / No. / Edition</th>
<th>Pub. year</th>
<th>Pages</th>
<th>Date accessed</th>
<th>Location, data base, website, link</th>
<th>Added in the final paper?</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Annotation</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>- League of Legends now has 15M registered users, with 1.4M playing every day (this is the most in any single MMO)</td>
<td></td>
</tr>
</tbody>
</table>
- Streaming of live championships drew over 210K concurrent views with 1.69M total viewers over a 2-day period

<table>
<thead>
<tr>
<th>Author</th>
<th>Title of Article</th>
<th>Periodical/website</th>
<th>Vol. / No. / Edition</th>
<th>Pub. year</th>
<th>Pages</th>
<th>Date accessed</th>
<th>Location, data base, website, link</th>
<th>Added in the final paper?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Murphy, David</td>
<td>League of Legends Databases Hacked, But Payment Info Safe</td>
<td>PCMag</td>
<td></td>
<td>June 10, 2012</td>
<td></td>
<td>June 25, 2012</td>
<td><a href="http://www.pcmag.com/article2/0,2817,2405566,00.asp">http://www.pcmag.com/article2/0,2817,2405566,00.asp</a></td>
<td>YES</td>
</tr>
</tbody>
</table>

Annotation
- League of Legends database hacked but no personal information leaked

<table>
<thead>
<tr>
<th>Author</th>
<th>Title of Article</th>
<th>Periodical/website</th>
<th>Vol. / No. / Edition</th>
<th>Pub. year</th>
<th>Pages</th>
<th>Date accessed</th>
<th>Location, data base, website, link</th>
<th>Added in the final paper?</th>
</tr>
</thead>
</table>

Annotation
- Details the following to players:
  - Blizzard’s security responsibility
  - Provides a security to-do list for players
  - Information about various account theft methods and phishing techniques
  - What to do when a player gets hacked

<table>
<thead>
<tr>
<th>Author</th>
<th>Title of Article</th>
<th>Periodical/website</th>
<th>Vol. / No. / Edition</th>
<th>Pub. year</th>
<th>Pages</th>
<th>Date accessed</th>
<th>Location, data base, website, link</th>
<th>Added in the final paper?</th>
</tr>
</thead>
</table>

Annotation
- Provides the details on the functionality of a Battle.net authenticator
- Comes in mobile app form or a physical device that can be attached to a keychain
- Exactly the same as authenticators used by banks for online banking

<table>
<thead>
<tr>
<th>Author</th>
<th>Title of Article</th>
<th>Periodical/website</th>
<th>Vol. / No. / Edition</th>
<th>Pub. year</th>
<th>Pages</th>
<th>Date accessed</th>
<th>Location, data base, website, link</th>
<th>Added in the final paper?</th>
</tr>
</thead>
</table>

Annotation

- Provides the details on the functionality of a Battle.net authenticator
- Comes in mobile app form or a physical device that can be attached to a keychain
- Exactly the same as authenticators used by banks for online banking

<table>
<thead>
<tr>
<th>Author</th>
<th>Title of Article</th>
<th>Periodical/website</th>
<th>Vol. / No. / Edition</th>
<th>Pub. year</th>
<th>Pages</th>
<th>Date accessed</th>
<th>Location, data base, website, link</th>
<th>Added in the final paper?</th>
</tr>
</thead>
</table>

Annotation

- Provides useful graphics and revenue growth trends for the MMO industry
- Outlines key challenges for MMOs
- Compares free-to-play + micro-transaction models with subscription-based models

<table>
<thead>
<tr>
<th>Author</th>
<th>Title of Article</th>
<th>Periodical/website</th>
<th>Vol. / No. / Edition</th>
<th>Pub. year</th>
<th>Pages</th>
<th>Date accessed</th>
<th>Location, data base, website, link</th>
<th>Added in the final paper?</th>
</tr>
</thead>
</table>
- Interview with Bruce Schneier
- Misplaced priorities
  - Fear of cyber crime
  - But funding mostly goes to cyber terrorism
- Bad attempts by government to regulate cyberspace
- Technical problems with technical solutions, they’re not political problems and we don’t really need political solutions
- Latest book on sociology of security
  - Economics and security
    - More publications about this topic in the last decade
    - More interesting research to come
- Security is hard to sell and is usually bundled
- Game theory
  - Stealing too much
  - How do we induce cooperation
- Different way of thinking
  - Not “wow that’s how it works”, but “wow that’s how it could fail”
  - Ant farm mail
  - Can’t teach this mentality
  - Need defectors from ex-“thieves”
  - White hat v. black hat
  - Need to teach how to write malware
- How far behind is the government
  - Not much farther than everyone else
  - Bad guy moves faster

**Author** | **Title of Article** | **Periodical/website** | **Vol. / No. / Edition** | **Pub. year** | **Pages** | **Date accessed** | **Location, data base, link** | **Added in the final paper?**
--- | --- | --- | --- | --- | --- | --- | --- | ---

**Annotation**
- Describes the Facebook photo exploit
- Essentially allows strangers to browse private albums of others