Ready Raider One: Exploring the Misuse of Cloud Gaming Services

Guannan Liu
Virginia Tech
Arlington, Virginia, USA
guannanliu@vt.edu

Daiping Liu
Palo Alto Networks, Inc.
Santa Clara, California, USA
dpliu@paloaltonetworks.com

Shuai Hao
Old Dominion University
Norfolk, Virginia, USA
shao@odu.edu

Xing Gao
University of Delaware
Newark, Delaware, USA
xgao@udel.edu

Kun Sun
George Mason University
Fairfax, Virginia, USA
ksun3@gmu.edu

Haining Wang
Virginia Tech
Arlington, Virginia, USA
hnw@vt.edu

ABSTRACT

Cloud gaming has become an emerging computing paradigm in recent years, allowing computer games to offload complex graphics and logic computation to the cloud. To deliver a smooth and high-quality gaming experience, cloud gaming services have invested abundant computing resources in the cloud, including adequate CPUs, top-tier GPUs, and high-bandwidth Internet connections. Unfortunately, the abundant computing resources offered by cloud gaming are vulnerable to misuse and exploitation for malicious purposes. In this paper, we present an in-depth study on security vulnerabilities in cloud gaming services. Specifically, we reveal that adversaries can purposely inject malicious programs/URLs into the cloud gaming services via game mods. Using the provided features such as in-game subroutines, game launch options, and built-in browsers, adversaries are able to execute the injected malicious programs/URLs in cloud gaming services. To demonstrate that such vulnerabilities pose a serious threat, we conduct four proof-of-concept attacks on cloud gaming services. Two of them are to abuse the CPUs and GPUs in cloud gaming services to mine cryptocurrencies with attractive profits and train machine learning models at a trivial cost. The other two are to exploit the high-bandwidth connections provided by cloud gaming for malicious Command & Control and censorship circumvention. Finally, we present several countermeasures for cloud gaming services to protect their valuable assets from malicious exploitation.

CCS CONCEPTS

• **Security and privacy** → **Network security**; *Web application security*; Systems security; • **Networks** → *Cloud computing*.

KEYWORDS

Cloud Gaming; Crypto-mining; Command & Control; Censorship

CCS '22, November 7–11, 2022, Los Angeles, CA, USA.

ACM ISBN 978-1-4503-9450-5/22/11.
https://doi.org/10.1145/3548606.3560647

ACM Reference Format:
Guannan Liu, Daiping Liu, Shuai Hao, Xing Gao, Kun Sun, and Haining Wang. 2022. Ready Raider One: Exploring the Misuse of Cloud Gaming Services. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security (CCS '22), November 7–11, 2022, Los Angeles, CA, USA.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3548606.3560647

1 INTRODUCTION

Cloud Gaming, also known as Gaming-as-a-Service (GaaS), has become an emerging computing paradigm that attracts significant attention from the gaming industry. Many enterprises from tech giants such as Google and Nvidia to small businesses like Shadow.tech and LoudPlay have all invested abundant resources to develop and commercialize cloud gaming services. Moreover, the Covid-19 pandemic and global GPU shortage may also be contributing factors for the rapid growth of cloud gaming [2, 6]. Cloud gaming services such as Nvidia's Geforce Now [18] have attracted more than 12 million registered players at the end of September 2021 [7]. Statistics suggest that cloud gaming has a global market value of $612.31 million in 2020 and could reach $5.4 billion by 2026 [4].

Cloud gaming enables players to enjoy a superior gaming experience on lower-end computing devices. The player's control signals, including keystrokes, mouse movements, and mouse clicks, are captured and transmitted to the cloud gaming services. The cloud gaming services process the received control signals and render the game frames. The game frames are sent back to the player as a video stream, which would be displayed on the player's monitor after video decoding. Using cloud gaming infrastructures, players only need computers with fast network connections and video decoding capabilities to enjoy a gameplay. Meanwhile, the cloud gaming services would be equipped with powerful hardware to accommodate all the computing tasks required by games. While many previous studies [30, 47, 48, 60] have focused on advancing cloud gaming technology, few have investigated cloud gaming services from a security perspective.

In this work, we present an in-depth study to reveal security vulnerabilities that could be exploited to abuse cloud gaming services. We investigate popular cloud gaming services including Nvidia Geforce Now [18], LoudPlay [13], and Shadow.tech [19], as their abundant computing resources in cloud gaming services are potentially an attractive target for adversaries. Under the camouflage of

playing computer games, adversaries can intentionally inject and execute malicious code, scripts, and programs in cloud gaming services, accomplishing complex computing tasks without investing a fortune in computing hardware and service maintenance. In cloud gaming services, players are typically provided with VM instances containing 8 cores of vCPU, 12-16GB memory, and a top-tier Tesla series GPU. The subscription fee varies across different services, ranging from $9.99 to $29.99 per month. By contrast, AWS EC2 instances with similar CPU and memory configurations cost users around $100 per month. With an additional GPU, the monthly bills could further rise to above $200. Therefore, the lower cost of cloud gaming services is an appealing factor for potential misuse and exploitation.

We uncover that adversaries are able to exploit cloud gaming services using game mods.[1] Specifically, there are two attack vectors in which adversaries can intentionally inject and execute malicious programs/URLs in cloud gaming services. Malicious programs, including internal functions and standalone programs, can be injected directly inside a game mod and executed via in-game subroutines and game options. These programs can be used to abuse cloud gaming services for malicious purposes. We also show that adversaries can inject malicious URLs into the cloud gaming services as mod descriptions. Adversaries can launch built-in web browsers in the cloud gaming services to visit the malicious URLs to execute malicious scripts.

To demonstrate that such resource misuse and vulnerability exploitation may pose a serious security threat, we conduct four proof-of-concept attacks. In the first attack of crypto-mining, our injected programs can abuse the CPUs and GPUs in cloud gaming services for mining operations. Using the crypto-mining scripts and programs, we benchmark the hashrates of each cloud gaming service to estimate the monthly profit, and our result shows that adversaries can easily earn more than $88 profit per account per month for mining cryptocurrency. In the second attack, we demonstrate that adversaries can take advantage of the high-performance GPUs provided by cloud gaming services to train machine learning models. As for the other two cases of abuse, the high-bandwidth connections of cloud gaming are exploited for malicious Command & Control (C&C) and censorship circumvention, respectively. More specifically, in the third exploit, we inject a UDP hole punching script to cloud gaming services to establish communication channels between cloud gaming servers and external clients, which can be used as a C&C infrastructure. In the fourth abuse, we demonstrate that cloud gaming services can also be utilized for censorship circumvention. This is because cloud gaming services only send encoded and encrypted video streams back to players, making it difficult for firewalls and traffic analyzers to detect and block network accesses. Finally, we present several defense mechanisms as a preliminary and generic guideline against resource misuse in cloud gaming.

To summarize, our study makes the following contributions:

- We uncover a new security vulnerability of exploiting cloud gaming services. Specifically, malicious programs could be injected into cloud gaming services via game mods, executing for adversarial purposes.
- We conduct measurement studies, showing the feasibility of exploiting such a vulnerability. We reveal that a large number of computer games support game mods, which adversaries can leverage to misuse cloud gaming services.
- We demonstrate four proof-of-concept attacks to exploit real-world cloud gaming services. We verify that adversaries can gain substantial benefits by exploiting the abundant GPU and network bandwidth resources.
- We disclose our findings to cloud gaming service vendors and propose mitigation practices.

The remainder of this paper is structured as follows. Section 2 introduces the background of cloud gaming and computer games. Section 3 presents the motivation, threat model, and ethical discussions. Sections 4 presents detailed exploitation procedures and Section 5 shows the feasibility of abusing cloud gaming services through game mods. Section 6 demonstrates four practical proof-of-concept attacks. Section 7 describes mitigation practices, as well as limitation and future work. Section 8 surveys related work, and finally Section 9 concludes our paper.

2 BACKGROUND

2.1 Games and Game Mods

Computer gaming has become a multi-billion industry for many decades [3]. In general, computer games can be categorized as single-player games and multi-player games. Single-player games allow players to enjoy a game session by themselves on a standalone machine. Typically no Internet connection is required (after legitimate online validation). Multi-player games require more than one players to collaborate or compete in one game session. Such games require network connections in order for all players to communicate with one another.

While game developers are constantly developing game content and mechanisms to enrich playability, each player may have individual gaming preferences that may not be supported by the official game release. Because of this, many game developers have introduced modding capability in their games. *Game mods*, also known as customized content, allow players to include unofficial contents that are developed by other players or third-party modding communities. Game mods can only be executed along with the original game since they are essentially add-on components. Many popular categories of game mods may include model assets, scenery, and storyline scenarios. Game mods may also contain additional programs and functions which fundamentally modify the gaming mechanisms.

Game platforms provide centralized services in which players can purchase, manage, and launch computer games. The Steam platform [20] developed by Valve Cooperation has been one of the largest game platforms in today's gaming industry. A recent report shows that Steam has listed more than 50,000 computer games in its U.S. game store in 2021 with a growth rate of 8,000 to 10,000 new games every year [10]. In early 2022, a statistic shows a record-high 27.9 million concurrent users on Steam [21].

With the increasing popularity of game mods, the Steam platform incorporates a mod management system named *Steam Workshop*.

[1] A mod is a player- or community-created modification of a game that extends the original game, enhancing the game's experience with new features or functions (*e.g.*, new characters, maps, or missions) [15].

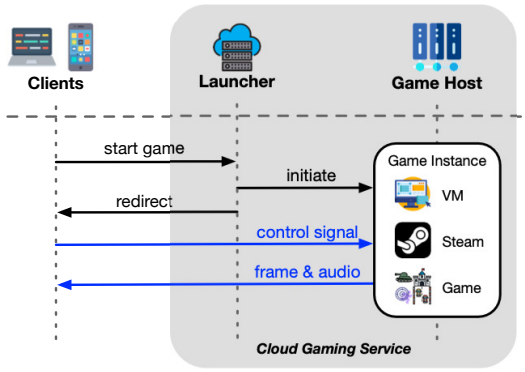

Figure 1: Architecture overview of cloud gaming.

It enables mod developers to host their game mods on Steam Workshop. Usually, game developers provide detailed procedures for the modding community on how to construct game mods and upload them to Steam Workshop. The uploaded game mods are automatically set to private by Steam Workshop and can only be seen by the owner of the game mods. Then, the game mods have to be manually configured to be public so that other players can subscribe to them. Players can subscribe to any public game mods in Steam Workshop when they own a copy of the corresponding original game. Once a game mod is subscribed, Steam Workshop automatically downloads the mod and the game can load the mod into a game session.

2.2 Cloud Gaming Services

Cloud gaming is essentially a computer gaming solution built atop of the cloud infrastructure to offload game execution and frame rendering from the player's local machine. The cloud gaming services provide abundant computing resources to players, *e.g.*, top-tier GPUs and high-bandwidth network connections. The top-tier GPU is to render all model assets, scenery, and objects in a game frame. The high-bandwidth network connections not only support the need for communication between multiple players in a game session, but are also required to exchange video frames and control signals between the player's machine and cloud gaming services. Both are key components for a smooth, responsive, and enjoyable gaming experience as they ensure fast game processing and low latency data transmission.

Cloud gaming inherits many characteristics from traditional cloud computing platforms. The cloud gaming services that we investigate in this work, including Nvidia Geforce Now, LoudPlay, and Shadow.tech, all share similar system design. Figure 1 illustrates the fundamental system architecture of a cloud gaming service. To launch a computer game in cloud gaming services, a player starts the game from a thin client (PCs, laptops, or mobile devices). The game launcher then initiates a game instance in the client device with a virtual machine (VM) running the game management platform (*e.g.*, Steam [20]) and the corresponding computer game. Once the game instance is ready, the game launcher redirects the player to establish communications with the game instance. The player then directly sends control signals such as mouse movements, mouse clicks, and keyboard strokes to the game instance in order to play the computer game. Based on the control commands, the game instance processes the game frames using the powerful computing hardware equipped with the cloud gaming services. Meanwhile, the encoded video streams are encrypted and transmitted back to the player's client. Finally, the player's client decrypts the transmitted data, decodes the video streams, and displays the game frames on the player's monitor.

While maintaining untarnished playability, many security regulations have already been implemented to avoid resource misuse and establish limitations on the player's authorizations and capabilities. For example, Nvidia Geforce Now allows players to only run games from their supported game lists. This prevents computer games listed on the Steam platform with malicious code such as Abstractism [1] from executing in Nvidia's system. In addition, a web browser in the system is configured so that players cannot download any files or install any browser extensions.

3 MISUSE OF CLOUD GAMING SERVICES

3.1 Motivation

In 2020, PCGamer and Tom's Guide published their top choices of gaming computers, and the price of a fully equipped gaming computer ranges from $1,049.99 to $3,799.99 [22, 23]. As a global chip shortage occurs in 2021, the price of GPUs has increased to a record high with the largest inflation rate of over 140% [5]. This significantly impacts the gaming industry as many gamers are unable to obtain a GPU due to the lack of GPU production and ever-increasing price inflation. The global GPU shortage has also impacted other GPU-accelerated applications such as crypto-mining and machine learning model training.

One of the major intriguing features of cloud gaming services is to save players' cost on expensive gaming hardware. Cloud gaming services provide top-tier computing hardware with an affordable monthly subscription fee while delivering enjoyable gaming experiences. However, it also becomes an attractive target if adversaries could exploit the rich resources of cloud gaming services for malicious purposes. In this study, we show that cloud gaming services are vulnerable to resource misuse. We demonstrate four case studies in which adversaries could gain financial benefits by exploiting cloud gaming services.

3.2 Playability vs. Security Trade-offs

Cloud gaming services are devoted to delivering an enjoyable gaming experience for all computer games. Such a commitment requires cloud gaming services to provide abundant computing resources to players. In terms of computing power, it is essential for cloud gaming services to provide powerful computing hardware to their players. This is because cloud gaming services need to accommodate some power-hungry computer games that require high-performance computing facilities to execute, even though many other games may not need powerful hardware support. Moreover, as illustrated previously in Figure 1, cloud gaming services need to transmit game video and audio streams to a player's machine while players need to send control signals back to the cloud gaming services. Thus, abundant network resources need to be allocated to ensure smooth, high-quality gaming frame delivery with low latency for providing responsive gameplay. However, such rich computing and network resources could attract adversaries to exploit them for malicious purposes.

Cloud gaming services are also committed to enable all functionalities of the games running in their systems. However, in order to meet various demands, some security restrictions have to be relaxed. For example, computer games, especially multi-player games, require Internet connections to third-party servers for hosting game sessions. Meanwhile, the IP destinations and domains, protocols, and port numbers vary across different games. Thus, it is very hard for network firewalls to perform conservative traffic blocking in/out of the cloud gaming services. Adversaries can take advantage of the high-bandwidth connection provided by cloud gaming services and host malicious services on the Internet.

Furthermore, cloud gaming services may have to allow game mods as many computer games enable the modding capability nowadays. This could be considered safe by cloud gaming services as Steam officially supports game mods via Steam Workshop. However, our work reveals that cloud gaming services are vulnerable to potential exploitation based on game mods. Using such a method, adversaries could execute malicious programs or visit dangerous URLs, which may cause serious damage to the cloud gaming services. Note that existing security regulations become futile if cloud gaming services are exploited via game mods, because adversaries do not need to download anything from the Internet other than Steam Workshop, and game mods are supported by many games.

3.3 Threat Model

In this study, we consider the specific exploitation by which adversaries are able to misuse cloud gaming services for non-gaming purposes. Figure 2 shows an overview of the misuse of cloud gaming services. Specifically, we unveil two attack vectors to misuse the resources of cloud gaming services: (1) Malicious Programs and (2) Malicious URLs. Adversaries can construct malicious programs and URLs by themselves. This may require fundamental knowledge of programming and game mod development. Adversaries can also package existing programs. Both of them are packaged into game mods by adversaries and injected into cloud gaming services.

We illustrate that adversaries are able to inject malicious game mods to cloud gaming services using Steam Workshop. and execute malicious programs using the in-game subroutine and game launch option. Such programs can abuse powerful computing hardware and high-speed networks for malicious activities. Furthermore, we reveal that the web browsers, including VM browser, Steam browser, and in-game browser, can also be exploited to execute malicious scripts and visit censored content.

To exploit the cloud gaming services, adversaries are required to register an account for the Steam platform and for each targeting cloud gaming service. Adversaries can register these accounts by obtaining email addresses from popular email providers for free, or they can obtain a large number of email addresses from disposable email services. In addition, while some cloud gaming services offer free trial options, the majority of them demand a subscription fee. Adversaries would also need to pay the subscription fee in order to launch such an attack on cloud gaming services. Furthermore, adversaries are required to obtain a copy of the game that they wish to carry out the attack. This can be accomplished by obtaining a free-to-play copy of the game or purchasing the game directly on the Steam platform.

Adversaries also need to develop game mods that contain malicious programs/URLs. Adversaries are required to acquire fundamental knowledge of programming and game mod development. Adversaries can also use programs developed by others, package them into a game mod, and upload it to Steam Workshop. The malicious game mod can be kept private since adversaries do not intend to distribute it to other players. This increases the stealthiness as no one other than the adversaries can subscribe to the game mod.

3.4 Ethics

To better understand the security vulnerabilities of cloud gaming services, we comprehensively explore the potential exploitation approaches. We intentionally launch various proof-of-concept attacks in each cloud gaming service in order to demonstrate its exploitability and profitability. We carefully design our experiment procedures to minimize any negative impacts on cloud gaming services, game hosts, and regular players. Also, our experiment is conducted under a controlled environment and does not involve any human interactions.

Our experiment is conducted in a legitimate manner. We registered all of our accounts legally through cloud gaming services, and we pay all required subscription fees. We also purchased the Transport Fever 2 [24] game on Steam which is used in our experiment to study malware injection through game mods. Moreover, we incorporate several precautions in order to minimize the potential influence on other players.

In terms of exploitation study, we use existing programs for the crypto-mining attack, and we develop our own programs for machine learning model training, C&C, and censorship studies. Especially, the crypto-mining programs used in the experiment are the latest release of the WebMinePool script and the NiceHash program. We download these programs directly from their official websites without any modifications or code injections. To host the malicious URL with WebMinePool script, we establish a domain name that has never been registered before. The game mod containing the NiceHash mining program is uploaded to Steam Workshop by following the official mod uploading procedures.

Moreover, we execute the crypto-mining programs for only one minute to obtain the hashrate of each cloud gaming service. The financial gain from the misuse of cloud gaming services is estimated based on the hashrate. We believe that one minute is sufficient to trigger a crypto-mining detection mechanism, as many countermeasures against intensive resource misuses can achieve real-time detection [33, 63]. We are aware that some players were banned from Shadow.tech due to their crypto-mining activities [8, 9]. This may indicate that cloud gaming services have already deployed defense mechanisms against crypto-mining. To confirm this, we explicitly contacted all three platforms to see whether they have deployed any form of detection mechanisms in their systems. Unfortunately, we have not yet received any responses from them about the deployment of defense. On the other hand, the fact that two cloud game vendors acknowledge our findings clearly evidences that our proof-of-concept attacks are indeed effective. This may also suggest that, given detection mechanisms, a very short duration attack (e.g., less than one minute) could successfully evade detection, benefiting adversaries in terms of attack stealthiness.

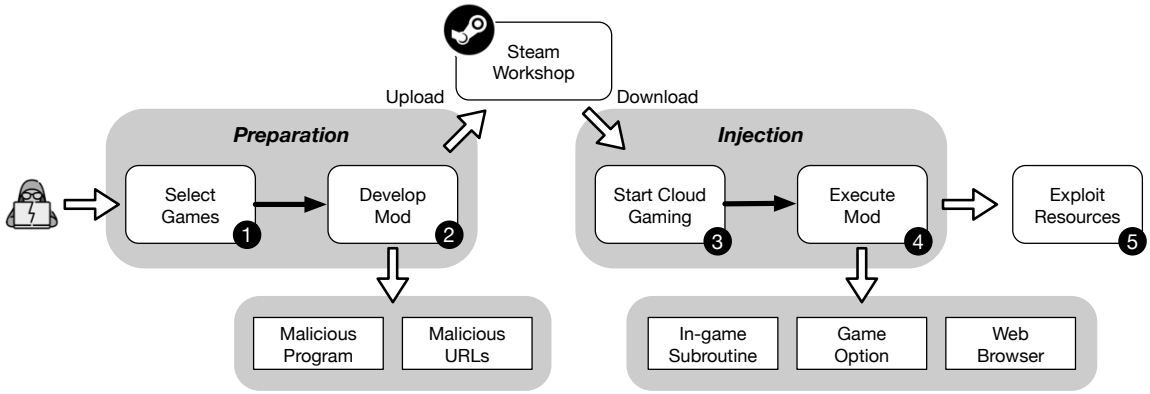

Figure 2: Overview of the misuse of cloud gaming services.

We estimate the monthly financial gains based on the hashrate. This can prevent the game hosts from consuming a large amount of power and potentially overheating [44, 75]. To further avoid potential damage to the cloud gaming services and our neighboring players, we configure the crypto-mining program to execute with 90% CPU usage. In our study, we only demonstrate the feasibility of abusing cloud gaming services for both activities. Therefore, our developed program should have little impact on the overall normal operations of cloud gaming services.

Finally, at the end of our study, we manually deleted all accounts in cloud gaming services. We also remove the URL with the crypto-mining script and all the game mods containing malicious programs. We disclose our findings to the affected cloud gaming services via emails and official bug reporting channels. Our disclosures also include some mitigation approaches described in Section 7. We have received responses from both Nvidia and Shadow.tech, in which they acknowledged our findings and stated that they will conduct further investigation. We indeed observed some changes in cloud gaming services after our disclosure. For example, Nvidia has completely blocked players from visiting the NiceHash website. Unfortunately, it is not confirmed that Nvidia made such a change due to our disclosure.

4 METHODOLOGY

4.1 Malicious Content Tag-along

Our study reveals that adversaries can exploit cloud gaming services for malicious purposes via game mods. Essentially, we demonstrate that game mods can carry malicious programs/URLs into the cloud gaming services. This is achieved using Steam Workshop, Steam's official mod management system. In this section, we conduct a systematic study on Steam Workshop. We show that Steam Workshop cannot detect malicious content to be tagged along with game mods in order to exploit cloud gaming services.

As mentioned above, the purpose of incorporating mod functions into a game is to allow players to customize game sessions so that they can enjoy extra content that is not officially released by the game developers. Steam Workshop provides a centralized management service for both mod developers to host their developed mods and for players to subscribe to the mods that they are interested in. Once a mod is subscribed by a player, Steam Workshop downloads the mod to the player's computer. When the players launch the game, all subscribed mods are automatically loaded and incorporated into the computer game.

To inject malicious content, adversaries need to interact with Steam Workshop to upload and download malicious mods. Specifically, adversaries upload their own game mod to Steam Workshop along with all malicious content. They also subscribe to the same malicious mod in Steam Workshop. Then, the adversaries launch the computer game in cloud gaming services and execute the malicious contents in the mods

Steam Workshop is the key component in the process of injecting malicious content to cloud gaming services. Understanding the mod management policy in this system provides us with key insights into the feasibility and effectiveness of exploiting cloud gaming services using game mods. While the exact procedures for uploading mods to Steam Workshop differ across different games, the game developers, through the official website or Wiki page, publish the methods of uploading mods to Steam Workshop. To comprehensively explore the mod management policy, we conduct a proof-of-concept experiment in which we attempt to upload game mods to Steam Workshop. We use Transport Fever 2 as our test game and we construct a game mod. To simulate the real attack, we intentionally inject a "malicious" payload in the mod containing a crypto-mining script written in Python. Since this is only an exploration, we do not execute the mining script to mount any actual attacks at this stage. Following the procedures stated on the Wiki page [25], we successfully uploaded our mod to Steam Workshop. We launch the Transport Fever 2 game in all three cloud gaming services and observe that the mods are indeed automatically downloaded into the cloud gaming services along with the "malicious" crypto-mining script.

Furthermore, in Steam Workshop, the mod developers can configure the mod to be either public or private. Whereas public mods can be subscribed to by all players who own the game, private mods can be only seen by their developers. This, from the adversaries' point of view, helps the stealthiness of the attack. The goal of such an attack is to exploit the resources in cloud gaming services but not to spread out malicious mods. Therefore, configuring malicious mods as private is sufficient to launch attacks by executing malicious code with cloud gaming services. Others, including players and system administrators, may be less aware of the malicious contents in the game mods uploaded by adversaries.

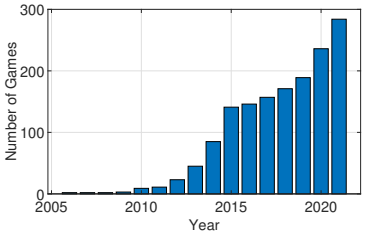

Figure 3: Number of games released per year with mod support.

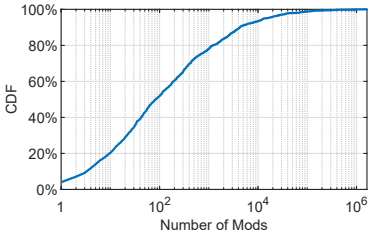

Figure 4: CDF of number of mod for each game.

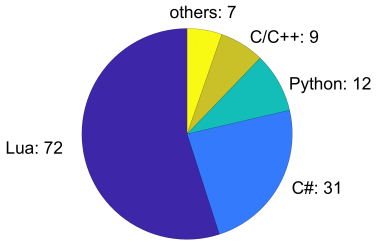

Figure 5: Distribution of programming languages used in mod.

4.2 Exploitation Process

To exploit cloud gaming services, adversaries can follow the steps shown in Figure 2. In general, the exploitation consists of five steps: (1) Select Games, (2) Develop Mod, (3) Start Cloud Gaming, (4) Execute Mod, and (5) Exploit Resources. As such, adversaries are able to harvest the abundant resources provided by cloud gaming services for malicious purposes. Here, we explain each exploitation step in further detail.

Step ❶: Select Games. The first step in the preparation phase of the exploitation is to select a computer game used to mount the attack. In general, adversaries can choose any games on the Steam platform as long as they support modding. Adversaries also need to select the games that are supported by cloud gaming services.

We first comprehensively explore the modding support on the Steam platform. We develop a crawler script to obtain the entire game database on Steam. In total, we collect 32,311 computer games in the Steam library as of Aug 2021 and among which, 1,506 games (4.7%) support mods through the Steam Workshop. In addition, we find 1,244 games that support third-party mod managers such as Nexus [16] and ModDB [14]. Figure 3 shows the number of games with mod supports with respect to their release date. The overall trend is obvious that the modding capabilities have become increasingly popular among the latest released games. We anticipate that game mods would continue its trend and more developers would incorporate modding features in their released games.

Also, Figure 4 illustrates the CDF with respect to the total number of mods in a single game, showing a vigorous modding community where 50% of games contain more than 100 mods in Steam Workshop. The game with the largest number of mods (Garry's Mod) has more than 1.6 million customized mods hosted on Steam. Such a big modding community could cause significant management costs and issues for Steam, as adversaries can stealthily upload malicious game mods to Steam Workshop.

Step ❷: Develop Mods. The second step is that adversaries need to develop malicious mods for the selected games. The developed mods can contain either malicious programs (Section 5.1) or malicious URLs (Section 5.2) to inject exploitation content.

We also observe that various programming languages can be used to develop game mods, providing adversaries with rich opportunities to choose their favorable languages to construct malicious mods. We manually inspect 314 computer games that we have access to and identify 124 games with modding capabilities. Figure 5 shows the distribution of programming languages that are supported by each computer game. The most popular language used in game mods is Lua scripting, which is supported by 73 games. Other major programming languages include C#, Python, and C/C++. Especially, we find 7 games that support more than one programming languages to be used for modding.

When the mod is fully developed, adversaries need to upload the malicious mods to Steam Workshop. The uploaded mods can be configured as private to keep the mods from the general public.

Step ❸: Start Cloud Gaming. Adversaries can then subscribe to the malicious mods that they upload and execute the mods within cloud gaming platforms. Adversaries first need to run the computer game that they selected previously. This automatically triggers Steam Workshop to download the malicious mods that they constructed. Adversaries can verify the mod download through the Steam download page.

Step ❹: Execute Mod. Once the malicious mods are downloaded to the cloud gaming services, adversaries need to execute the malicious programs or URLs in the game mods. We reveal three methods to execute malicious content: (1) In-game Subroutine, (2) Game Option, and (3) Web browser. Specifically, In-game Subroutine is used to execute malicious code that can run within a game session. Game Option can be utilized to execute standalone malicious programs as separate processes. Section 5.1 presents both approaches in greater detail. On the other hand, when adversaries click the malicious URLs, a web browser is launched to visit the corresponding web page. We uncover three types of web browsers that may be built into the cloud gaming services, including VM built-in browser, Steam built-in browser, and in-game browser. We further explore the characteristics of these web browsers in Section 5.2.

Step ❺: Exploit Resources. Finally, with the malicious mod executed, adversaries can exploit the abundant resources provided by cloud gaming services. In this study, we primarily focus on exploiting the top-tier GPUs as well as high-bandwidth network connections. Section 6 shows four proof-of-concept studies, including crypto-mining, machine learning model training, Command & Control, and anti-censorship. We demonstrate that adversaries are able to gain a considerable amount of financial benefits from exploiting cloud gaming services.

5 EXPLOITATION

In this section, we investigate the feasibility of exploiting cloud gaming services through game mods. Specifically, we demonstrate that adversaries can intentionally inject malicious programs and URLs through game mods. Adversaries can execute malicious programs using in-game subroutine and game launch options, while

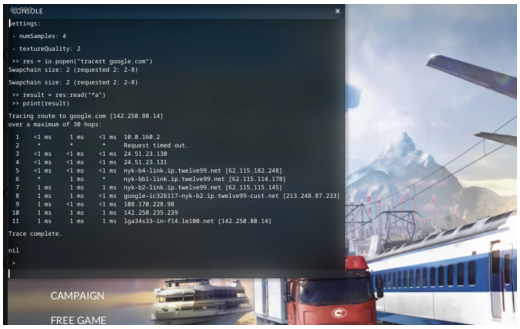

Figure 6: Transport Fever 2 and its debug mode running traceroute.

malicious URLs can be accessed by various web browsers built inside the game instance.

5.1 Malicious Program

Internal Functions. The intention of providing modding capability in computer games is to allow players to add or modify the games to their own taste. Among many add-on components that game mods contain, a special category of mods aims to modify the gaming mechanisms fundamentally. Such mods bring external pieces of code to the game and they are executed along with the original game. From the adversaries' point of view, this provides an opportunity to inject and execute malicious code through game mods.

Malicious code can be executed within a game session since the code itself is treated as a part of the game. Also, since malicious code is injected by the adversaries themselves, they would know the exact procedure on how to access the malicious function. In addition, adversaries can further exploit the mod debug tool, which is often built into many games by the game developers as an extended feature, enabling the modding community to easily test their code in a game session. Adversaries can utilize the mod debug tool to quickly locate the malicious code in their mod and execute them directly via the debug tool.

Standalone Program. In addition to the malicious code that is integrated as a function or subroutine, another method is to include well-developed programs into the mod itself. This can be achieved since a mod can contain many types of files, including external dependencies like library files and configuration files. Since Steam Workshop does not restrict the types of files included in a mod, adversaries can take such an advantage and package the entire malicious program inside a mod. This also reduces the technical requirements for adversaries because they do not need to develop or write any code.

Since these programs must be executed in separate processes, adversaries need to find other methods to launch standalone programs inside the cloud gaming environment. We identify that one practical approach to execute a standalone program in cloud gaming services is through game launch options. The game launch option is supported by Steam for players to enable special and extended features that are otherwise hidden from the game. Such features may include game cheating, debug mode, in-game console, *etc.* Our investigation suggested that the game launch option can be bypassed to run any standalone programs that adversaries want. In doing so, one can run "`full_program_path %command%`" with Steam game launch option. The above command can override the original launch command, resulting in Steam launching the program located at "`full_program_path`". Moreover, the "`full_program_path`" can also be replaced by built-in tools and programs of the system. For example, using the "`CMD %command%`" as a game launch option, the Steam platform would launch the built-in CMD prompt in Windows OS instead of the computer games. With the command prompt, adversaries can perform all types of actions, including browsing the file system, executing malicious programs, and modifying any system configurations.

Another method to run external standalone programs in cloud gaming services is via debug mode. Particularly, we test this functionality in Transport Fever 2. In debug mode, the in-game console of Transport Fever 2 supports Lua scripting and Lua commands/functions. The Lua commands and functions can be executed directly inside the console. In our exploration, we find that adversaries can run any executable programs using the *io.popen* function in the Lua module *io*. The semantic of *io.popen* is similar to the cmd prompt in Windows. To test its capability, we use the command "`io.popen("tracert google.com")`" to perform a traceroute from Nvidia Geforce Now server to Google server. Figure 6 shows the result of the traceroute command. With this functionality, adversaries can run any program in the game host and exploit the server for malicious purposes.

5.2 Malicious URLs

URL Injection. The injection of malicious URLs into cloud gaming services through game mods is straightforward. One of the simple methods is to include the malicious URLs in the description of their constructed mods. A mod description is essential for players to distinguish different mods that they subscribe to. Some game mods even contain designated sections for mod developers to include URLs such as the personal page of the mod developer and an external web page containing a comprehensive summary of the mod. Adversaries can take advantage of the mod descriptions to inject malicious URLs. That is, adversaries can construct an empty game mod that contains only the description of the mod with malicious URLs. The adversaries subscribe to the game mod, which is automatically downloaded to the cloud gaming services by the start of the game.

One special requirement for exploiting cloud gaming services with malicious URLs is that the injected URLs must be hyperlinked to the corresponding web pages. Some computer games can achieve this because of the embedded automatic URL detection feature in the mod description. For other games that do not contain such a feature, adversaries would need to utilize other tools. Our investigation suggests that adversaries can take advantage of the game chatrooms to convert raw URLs into hyperlinks. Many chatrooms incorporate this feature to identify the URLs in a message and automatically create a hyperlink to the corresponding web page. To comprehensively understand the feasibility of injecting malicious URLs through game chatrooms, we manually study the 314 computer games in our Steam game collection. Particularly, we reveal that 42 out of 314 (13.4%) computer games contain in-game chatrooms. These computer games consist of 7 single-player games and

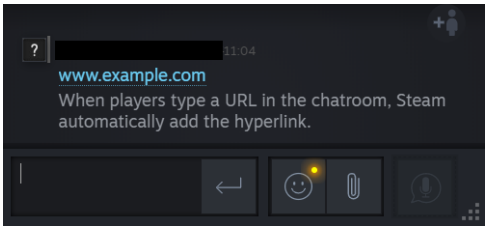

Figure 7: Example of Steam built-in chatroom with injected www.example.com **domain.**

35 multiplayer games, which account for 16.7% and 83.3%, respectively. The 7 single-player games are categorized as single-player by Steam, but the chatroom feature is embedded in multiplayer gameplay. Within the 42 computer games, 39 in-game chatrooms are equipped with the URL detection feature, which automatically generates hyperlinks if it detects URLs in the messages. The only three chatrooms that do not have the URL detection feature are all single-player games. With the URL detection feature, adversaries can easily generate hyperlinks for the injected malicious URLs. The Steam built-in chatrooms can also be used for this purpose. Figure 7 shows an example of injecting URLs using Steam's built-in chatroom. By generating hyperlinks for the injected URLs, adversaries can visit the corresponding web pages and execute malicious code and scripts.

Another method of injecting malicious URLs is to utilize the Wiki pages, forums, and blogs of computer games. Many game developers compose these pages in order for players to obtain information about game mechanisms, discuss potential issues, and share their game experiences. Within the 314 games that we examine, we find 231 games with official or third-party Wikis, forums, and blogs. Among them, 86 games contain at least one external website that is accessible from within a game session. To make these sites easily accessible to players, many developers embed them directly into their games using buttons or icons. When they are clicked, the game automatically launches a web browser to display the corresponding websites. However, our exploration suggests that, while game developers are dedicated to improving and polishing their computer games, the contents hosted on Wikis, forums, and blogs are often neglected. This provides adversaries with an opportunity to inject malicious URLs. Taking advantage of wikis, forums, and blogs, adversaries inject URLs into these sites by posting messages. Adversaries can manually create hyperlinks to their malicious websites. To access the malicious websites, adversaries can simply click the site icons in a game session and search for their self-injected URLs to visit malicious websites.

Execution. Once the malicious URLs are injected into the cloud gaming services, adversaries can simply click the hyperlinks in order to access the web pages. To open the web pages, our investigation shows three types of browsers game instances may open: (1) VM built-in browser, (2) Steam built-in browser, and (3) in-game browser. Here, we investigate each of the browsers in detail to understand the feasibility for adversaries to carry out malicious activities in cloud gaming services.

The virtual machine serves as a fundamental layer in a game instance, running on an operating system able to host other applications such as Steam software and computer games. Our investigation shows that all three cloud gaming services provide the players with a Windows VM. By default, the Windows OS contains a built-in browser of Microsoft IExplorer (MSIE) or Microsoft Edge, depending on the version of the Windows OS. In addition to the Windows built-in browsers, game hosts in Nvidia Geforce Now and Shadow.tech also contain Google Chrome browsers in their VMs. In both systems, Google Chrome is configured as the default browser. Adversaries may use these browsers to visit malicious websites to compromise the game hosts.

Furthermore, the Steam platform is supported by all three cloud gaming services that we investigate in this study. With the intention of aiding players to browse online content without the need to pause or exit the game, Steam integrates its own web browser into the application. The Steam built-in browser is a variant of Google Chrome with additional steam commands. However, some browser functionalities are strictly limited, such as Google account management, content downloading, and browser extensions. Therefore, the Steam built-in browser may not be able to execute exploitation on game hosts that require these aforementioned features. The Steam built-in browser can also be accessed in a game session. Adversaries simply press *SHIFT+TAB* on the keyboard to open the Steam console. Then, at the bottom of the screen, adversaries click the "WEB BROWSER" button to launch the browser. From here, adversaries may visit malicious websites to exploit the cloud gaming services.

In addition, our investigation reveals that some game engines, such as Unity [26], support in-game browser functionality. Unfortunately, none of the 314 games in our Steam collection support native in-game browser functions for us to investigate. However, we realize that 2 of them have game mods developed by the players' community to enable in-game browsing. Specifically, Gary's mod has a dedicated mod in the Steam Workshop that explicitly implements web browsers in the game. Minecraft, a well-known PC game in recent years, has a mod published online, which supports full browsing functionality in the games. To this end, adversaries can utilize these in-game browser mods to exploit cloud gaming services through malicious URLs.

6 CASE STUDIES

Taking advantage of the security vulnerabilities of game mods, adversaries can easily exploit game hosts in cloud gaming services for malicious purposes. In this study, we demonstrate four cases of misuse in cloud gaming services from the perspectives of GPU Exploitation and Bandwidth Exploitation: (1) Crypto-mining, (2) Training Machine Learning Model, (3) Establishing Command & Control, and (4) Censorship Circumvention. We show that both GPU and bandwidth exploitation can lead to considerable benefits for adversaries even though we conduct only proof-of-concept attacks due to ethical considerations.

Each cloud gaming service adopts different policies in terms of session length. This limits the total number of hours a player can access each game session. NVidia allows 6 hours of a gaming session, and the system automatically disconnects a player after the session expires. However, this does not affect the overall

Service	Instance	CPU Config		GPU Config			Network	Price/Month
		# of vCPU	Memory(G)	# of GPU	GPU Type	Memory(G)	Bandwidth	
Cloud Computing	t4g.large	2	8	-	-	-	5G	$33.73
	t4g.xlarge	4	16	-	-	-	5G	$64.54
	a1.2xlarge	8	16	-	-	-	10G	$96.81
	g4dn.xlarge	4	16	1	NVIDIA T4	16	25G	$241.63
	g4ad.2xlarge 8	8	32	1	AMD Radeon Pro V520	8	10G	$248.89
	p2.xlarge	4	61	1	NVIDIA K80	12	N/A	$448.22
	p3.2xlarge	8	61	1	NVIDIA Tesla V100	16	10G	$1,524.24
Cloud Gaming	Geforce Now	8	16	1	NVIDIA Tesla P40	24	4G	$0 - $9.99
	LoudPlay	8	16	1	NVIDIA RTX-6000P	6	2G	$10.12
	Shadow.tech	8	12	1	NVIDIA P5000	16	2G	$29.99

Table 1: Hardware and subscription fee comparison between cloud computing and cloud gaming.

profitability of our exploitation, since adversaries can immediately launch another game session to continue misusing the NVidia platform. Shadow.tech does not employ any restrictions on game session length. Adversaries could execute malicious programs on Shadow.tech for an unlimited duration to maximize profitability. LoudPlay adopts a different strategy in which they charge players based on the number of game hours. Therefore, as long as adversaries continuously pay the hourly subscription fee, they can exploit the LoudPlay system without limitations.

6.1 GPU Exploitation

The temptation for adversaries to abuse cloud gaming services is due to the significant price differences. Table 1 lists the computing hardware provided as well as the monthly subscription fees for cloud gaming services compared to Amazon EC2 instances. The EC2 t4g.xlarge instance, which contains a similar CPU and memories but without GPU, is estimated a monthly subscription fee of $64.54. If users request an additional GPU in their instance, the subscription fee raises significantly. Compared to t4g.xlarge, the g4dn.xlarge instance provides the same number of vCPUs and memory capacity, but with a dedicated Nvidia T4 GPU. The subscription for g4dn.xlarge raises to $241.63 per month, which is 274.4% higher than the t4g.xlarge instance. In comparison, the majority of cloud gaming services provide similar hardware to players as g4dn.xlarge, with 8 vCPUs, 16G memory, and a dedicated GPU; but the monthly subscription fee is considerably low, with $9.99, $10.12, and $29.99 for Nvidia Geforce Now, LoudPlay, and Shadow.tech, respectively. Undoubtedly, this brings significant financial benefits to adversaries if the GPUs in cloud gaming services are exploited.

6.1.1 ***Crypto-mining.*** Recent years have shown many security incidents of crytpojacking in which adversaries abuse the victim's computing resources to mine cryptocurrencies [33–35, 38, 46]. Inspired by these previous works, we attempt to conduct similar cryptojacking activities in cloud gaming services. We demonstrate that crypto-mining programs can be successfully executed inside all three of the cloud gaming services. Leveraging the abundant computing power, especially the provided top-tier GPU, adversaries can potentially make a considerable profit from exploiting cloud gaming services to mine cryptocurrencies.

Device	Algorithms	Nvidia	Shadow	LoudPlay
GPU	DaggerHashimoto	32.73M	21.71M	27.55M
	ZHash	-	21.12	35.57
	KAWPOW	16.35M	14.57M	12.67M
CPU	RandomXmonero	1,452	1,306	1,635

Table 2: Hashrate comparison across all three cloud gaming services. (Unit: Hash/s)

In this study, we use two existing crypto-mining programs. NiceHash [17] is a standalone mining program that can be executed directly on Windows OS. We use NiceHash to demonstrate the potential full-scale exploitation of cloud gaming services with GPU supports. In comparison, we also investigate crypto-mining exploitation using WebMinePool [27]. This program enables us to embed a JavaScript mining program into a web page, allowing us to carry out our exploitation through malicious URLs. We demonstrate that adversaries can still gain benefits from mining cryptocurrencies by abusing only the CPU, even though the profit is not as significant as GPU mining.

Methodology. We launch attacks on cloud gaming services including Nvidia Geforce Now, LoudPlay, and Shadow.tech. Our attack methodology follows the same procedure as shown in Figure 2. The first step in the preparation phase of the attack is to select a carrier game with modding capabilities. In this study, we choose Transport Fever 2 as it is one of the games in our Steam collection that is supported by the three cloud gaming services. Next, we construct our own game mod for Transport Fever 2. Specifically, we repackage the entire NiceHash software in our created mod. This allows us to execute the program without the need for any software installation. For the CPU exploitation, we construct a web page containing the official mining script provided by WebMinePool using our own domain. We include the URL of this web page in the description section of our game mod. This web page displays a simple user interface that allows us to select the number of CPU cores as well as the overall CPU usage that the mining script can consume. It also displays the start/stop buttons so that we can control the duration of the attack. We host the web page with mining scripts on an Amazon AWS server.

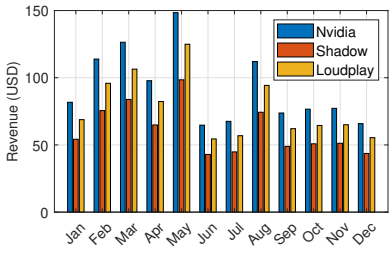

Figure 8: Monthly profit of crypto-mining using NiceHash

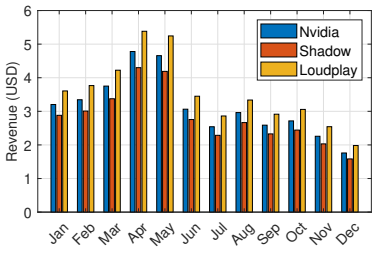

Figure 9: Monthly profit of crypto-mining using WebMinePool

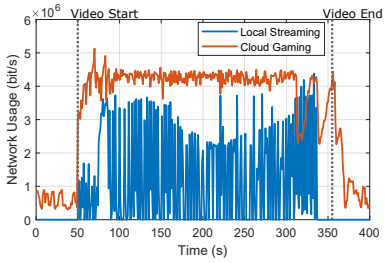

Figure 10: Network usage between local streaming and cloud gaming

We upload our game mod to Steam Workshop. We also subscribe to our own mod, which enables Steam Workshop to automatically download the mod when the game starts. For ethical concerns, we configure the mod as private so that the mod cannot be downloaded by other players.

After preparing the malicious content, we move to the injection phase in which we act as a regular player. We act as a regular game player and obtain an account for each cloud gaming service. We start the Transport Fever 2 game on all three cloud gaming services. This triggers Steam Workshop to download our mod to the game instances in cloud gaming services. To run the NiceHash program, we enable the debug mode of Transport Fever 2. This allows us to utilize the in-game console as shown in Figure 6. Using the *io.popen* command, we launch the NiceHash program in the game hosts as a separate process. We configure NiceHash to run with 90% CPU usage, but we do not set any limits on GPU usage. To access the web page containing the WebMinePool script, we simply click on the URL in the mod description. The Steam built-in browser is automatically launched and the web page is opened in the browser. Similar to the NiceHash software, we configure this WebMinePool mining script to execute all available CPU cores with 90% usage.

Finally, we begin our investigation by executing both programs in all three cloud gaming services. We evaluate the computing power of cloud gaming services by using the built-in benchmark tool. We benchmark each cloud gaming service five times and we use the average of the five hashrates as our evaluation results. This can stabilize the benchmark results, leading to a more realistic estimation. Based on the hashrates, we can estimate the actual revenue in US dollars based on the profitability chart [12]. We subtract the revenue from the subscription fee of the cloud gaming services, and the results are the expected profits due to crypto-mining exploitation.

Result and Analysis. Our exploitation attempts are executed successfully in all cloud gaming services. To establish our baseline, we first evaluate the computing power of the game hosts by running a series of mining algorithm benchmarks. These benchmarks provide us the hashrate of each system which largely determines the profitability of mining operations. Table 2 lists the hash rate for each cloud gaming service. Among the GPU benchmarks, we find that DaggerHashimoto has the highest hashrate among the algorithms in all cloud gaming services. Therefore, our profit estimation is based on the hashrates of DaggerHashimoto mining Ethereum (ETH). For CPU mining, the benchmark only supports the RandomXmonero algorithm for mining Monero (XMR).

Our profit estimation shows that adversaries executing the mining operations in cloud gaming services by NiceHash can generate a substantial amount of profit. This is largely due to the GPU mining support embedded in NiceHash. Figure 8 shows the profitability of mining cryptocurrencies through NiceHash. It is obvious that the total revenue gained from crypto-mining significantly exceeds the cost of subscriptions in all three cloud gaming services. At the highest profit point, adversaries are able to gain $148.42, $98.45, and $124.93 per month from Nvidia Geforce Now, Shadow.tech, and LoudPlay, respectively. Taking the subscription fees into consideration, the overall profit from GPU mining in Nvidia Geforce Now becomes $138.43, making it the most profitable cloud gaming service to exploit. In LoudPlay, adversaries can potentially make $94.94 per month from GPU mining. While Shadow.tech demands the most expensive subscription fee of $29.99, adversaries can still generate a profit of $88.33 per month by exploiting the cloud gaming service for crypto-mining.

Figure 9 shows the profitability of mining cryptocurrencies using WebMinePool. Since cloud gaming services virtualize the CPU to support multiple players, it is anticipated that the revenue gained from CPU-based mining is fairly low. In general, our results indicate that mining cryptocurrencies through only CPUs in cloud gaming services cannot compensate the subscription fee for cloud gaming services. The mining operations in LoudPlay generate the highest revenue of $5.24 per month in April 2021. However, because of the subscription fee of $10.12, mining operations could lose $4.48 per month. One viable method to generate profit from CPU mining is to use the free tier service provided by Nvidia Geforce Now. With the cost of subscriptions down to zero, adversaries are able to gain the full revenue from the mining operations. At the highest profit point, adversaries can potentially gain a profit of $4.78 per month. This profit could be further expanded if adversaries register a large number of accounts to mine cryptocurrency simultaneously. In such a case, the overall profit could be significant even with CPU mining only.

6.1.2 ***Training Machine Learning Model.*** Machine learning has been widely used in a large variety of applications. As machine learning models require high-performance computation to train, high-end computer hardware has been invested for this purpose. Especially, the model training in the development of machine learning applications demands top-tier GPUs for efficient data processing. In this case study, we demonstrate that adversaries can exploit

the GPUs provided by cloud gaming services for machine learning model training.

As shown in Table 1, all three cloud gaming services provide Tesla GPUs by Nvidia. To train machine learning models using Nvidia GPUs, the CUDA toolkit is required to be installed in cloud gaming services. For the installation process, we utilize the official CUDA installation program provided by Nvidia. We construct a game mod for Transport Fever 2 and include the installation program inside the mod. We inject the mod to the cloud gaming services and execute the program using the in-game console of Transport Fever 2. We confirm that the CUDA toolkit is successfully installed in cloud gaming services by executing several CUDA sample programs.

We then demonstrate the possibility of training a machine learning model in cloud gaming services. We conduct a proof-of-concept experiment by developing our own Python script for image classification model training. For ethical considerations, our aim is not to train a full-scale machine learning model in cloud gaming services. In our experiment, we only train one layer of the model using 10 labeled images. The script and training data are injected into the cloud gaming services along with our game mod. We confirm that the Python script can indeed produce an image classification model, and the training time is under 5 minutes.

The exploitation of cloud gaming services for machine learning model training can considerably benefit the adversaries by saving their investment on powerful computing hardware. As shown in Table 1, cloud gaming services provide powerful GPUs with significantly lower subscription fees. While the monthly usage fees range from $241.63 to $1,524.24 in Amazon cloud computing services, adversaries can use a GPU for less than $30 per month by exploiting cloud gaming services.

6.2 Bandwidth Exploitation

Besides powerful computing hardware, cloud gaming services also provide players with high-bandwidth low-latency network connections. The usage fee of this network connection is included in the monthly subscription; no other charges are collected from the players. In comparison, AWS EC2 services also charge their customers based on network usage, with a data transfer price as high as $0.09 per GB [11]. If adversaries abuse the cloud gaming services to perform network-demanding tasks, they would avoid paying a fortune to the cloud computing services for using the network.

6.2.1 ***Establishing Command & Control.*** Many computer games, especially multiplayer games, rely on peer-to-peer (P2P) UDP connections to exchange essential gaming data across multiple players. Such a requirement inevitably prohibits cloud gaming services from blocking UDP communications in/out of the game hosts due to security concerns. From the adversaries' point of view, this provides a practical opportunity to exploit the cloud gaming services for malicious data communications. In this case study, we demonstrate that adversaries can establish Command and Control (C&C) servers which cause severe security problems including botnet [31, 68] and domain generation algorithm (DGA) [32] attacks. Whereas cryptomining operations exploit the abundant computing resources in cloud gaming services, C&C abuses the high Internet bandwidth provided by cloud gaming services. Using UDP hole punching and an external relay server, adversaries can expose unoccupied UDP ports to the Internet. Any clients who know the port number and the IP address of the game host can establish communications to transmit and receive data. To investigate the feasibility of such exploitation, we establish an external relay server in Amazon AWS, and we intentionally inject a UDP hole punching script in our own game mod using Transport Fever 2. The system architecture of our established C&C server is illustrated in Figure 11. We demonstrate that we can successfully establish UDP connections with external clients and communicate with them using full network bandwidth.

Methodology. In the preparation phase of the exploitation, we first establish a relay server on Amazon AWS to accept UDP connections from the Internet. We record the IP address and the UDP port used by the relay server. Then, we inject a UDP hole punching script inside the same mod that we develop for crypto-mining exploitation. We configure our hole punching script to communicate with the relay server using the IP address and the port number that we record in the previous step. We upload our mod to Steam Workshop.

In the exploitation stage, we launch the Transport Fever 2 game in cloud gaming services. We execute our malicious mod in the game session, which establishes a UDP connection to the relay server. The relay server records the UDP connection (i.e., IP address and port number) received from the game host. Then, we generate 10 virtual machines using Google Cloud Service as external clients. We configure these clients to first contact the relay server and obtain the IP address and the port number used by the game host. We further verify that all 10 clients can successfully establish simultaneous UDP communications with the cloud gaming services.

Discussion. Using the aforementioned methodology, adversaries can easily exploit cloud gaming services for hosting C&C servers. The C&C server takes full advantage of the network bandwidth provided by cloud gaming services to exchange data with external clients. Since cloud gaming services do not demand any charges on network usage, adversaries can potentially save a large amount of money by establishing C&C server in cloud gaming services, instead of general cloud computing services. For example, while adversaries need to spend $900 on 10TB of network usage in Amazon AWS [11], such a cost will be diminished to zero by exploiting cloud gaming services.

Cloud gaming services may have to relax their network security policies to accommodate communication requirements for supporting all games. The network firewalls have to be configured to allow data transmissions in/out of the game hosts. Therefore, while we demonstrate the feasibility of establishing C&C server in cloud gaming services, such a relatively open network environment could potentially result in many other vulnerabilities exploitable by adversaries for malicious purposes.

6.2.2 ***Censorship Circumvention.*** Internet censorship is used to control the accessibility of Internet content. While censorship is often placed by governments or ISPs to block specific content, different ways have been leveraged to circumvent censorship, including the use of VPN and Tor browsers. In this case study, we demonstrate that adversaries can utilize the built-in web browsers in cloud gaming services to access blocked content on the Internet.

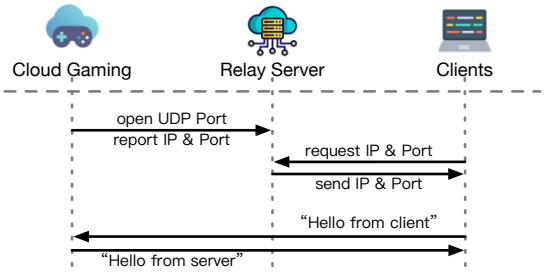

Figure 11: Establishing C&C with cloud gaming service.

Since the display frames transmitted from cloud gaming services to the clients are encoded and encrypted, network censors cannot retrieve or monitor the content in the display frames. Moreover, in order to deliver display frames with a constant resolution and refresh rate, the network connections between cloud gaming services and the clients should also be kept at high usage. This results in similar behaviors as network traffic obfuscation [42], making some censorship techniques such as network traffic analysis futile.

We explore the feasibility of circumventing censorship in cloud gaming services using the censorship dataset published by Jin *et al.* [55]. Specifically, we identify 20 domains from the dataset that are blocked by India and South Korea governments. We use commercial VPNs to position us in both countries. We verify that visiting these domains from a web browser results in displaying block pages constructed by the governments. We begin our exploitation by launching the Transport Fever 2 game in all three cloud gaming services. We manually type the domain URLs into the chatroom built inside Steam (example shown in Figure 7). By clicking the automatically generated hyperlinks, we confirm that the censored domains can be accessed and that the corresponding web pages are properly displayed in cloud gaming services.

Furthermore, we investigate the behavior of network traffic obfuscation when cloud gaming services are used. We select a video from YouTube that supports 1080p resolution at 60fps and play it using our lab servers and cloud gaming services. We record the network usage for both scenarios using WireShark [28]. Figure 10 plots the network bandwidth demanded by local video streaming and video streaming over cloud gaming services. As shown in the figure, the network usage for local video streaming consists of many spikes. This is due to the video buffering feature in which the browser pre-downloads the video contents before displaying them to the users. Once the buffer is filled, the download process will halt until some parts of the buffer are cleared. By contrast, the network connection for video streaming over cloud gaming services shows a comparably constant usage, despite two fluctuations near the end of the video due to packet losses. Such constant usage can be exploited as network traffic obfuscation to circumvent Internet censorship.

7 DEFENSE

7.1 Mitigation Practices

In order to mitigate the resource misuse of cloud gaming services, we present several defense practices. These methods aim to reduce the occurrence of malicious programs running in cloud gaming services, hence protecting their valuable assets from exploitation.

The infrastructure and ecosystem of cloud gaming are complex, involving cloud gaming services, game developers, and game management platforms such as Steam. A comprehensive defense may require the collaboration of all parties, which usually represent different entities. Furthermore, there are no standard implementations for cloud gaming, making a one-size-fits-all defense mechanism impractical to build. Thus, the countermeasures presented here only cover preliminary and primitive defensive practices limited to cloud gaming services' administrators, who should comprehensively examine their systems and implement proper defense mechanisms that best fit their system architectures and business needs.

Practice 1: *Process Monitoring.* In the cloud gaming infrastructure, a game instance contains three major components: VM, Steam software, and a game session. Processes that do not belong to any of these three categories should be considered suspicious activities that require further investigation. Therefore, to mitigate potential misuse, a process monitoring program should be implemented in the game instance (*e.g.* cryptojacking detection [29, 57, 71]), which can identify malicious processes and block them from running in the game instance. This will prevent abusive programs that need to run as standalone processes, such as NiceHash in crypto-mining, from being injected into cloud gaming services.

Moreover, in the case of a benign computer game running in cloud gaming services, the majority of the CPU and GPU usage should be associated with the game itself. If there are other processes with high CPU or GPU utilization, they are likely malicious programs. To this end, the process monitoring program can also be used to oversee the CPU and GPU usage of each process. With such straightforward monitoring, exploitation such as mining cryptocurrency using a built-in web browser can be mitigated.

Practice 2: *URL Request Redirection.* The web browsing feature in cloud gaming services provides players with a convenient way of accessing the Internet. However, browsing activity may also be conducted on the player's client-side, instead of on the game host side. To this end, cloud gaming services could implement a defense strategy to intercept all URL requests from the game hosts and redirect them to the player's client. Once the redirected URL requests are received at the player's side, a browser on the client can be automatically launched to visit the corresponding websites. This prevents the case in which cloud gaming services are misused to visit malicious content on a web page.

Practice 3: *Adaptive Network Policies.* Computer games may demand distinct network needs. While single-player games should not require external connections at all, multi-player games may need to communicate with some external servers in order to coordinate across all players. In order to accommodate the requirements for all computer games, cloud gaming services have not enforced any restrictions on network policies. This leads to the misuse of cloud gaming services for establishing C&C server and censorship circumvention. To mitigate such exploitation, cloud gaming services could carefully and comprehensively examine the supported games to identify their network demands. A primitive approach to accomplish this could be to leverage a sandbox to run each computer game and actively analyze the network connections requested by a game. The result can then serve as the baseline of network rules to restrict network access of each game. Cloud gaming services

allow external communications only to meet the network demands requested by each computer game and to block other unrecognized network connections. Moreover, we encourage both game developers and mod developers to self-report the network demands of their released products. Based on this information, cloud gaming services can implement more comprehensive and adaptive network policies for computer games supported by their platforms.

7.2 Limitation and Future Work

Our work reveals that adversaries can exploit cloud gaming services through game mods and malicious URLs for malicious purposes. We examine three popular cloud gaming services to demonstrate that such services can be exploited by adversaries for malicious activities. In the future work, we plan to explore additional security vulnerabilities and evaluate their security risks. These vulnerabilities and risks may involve computer game development, gaming engines, and game-server communication protocols. We also intend to investigate more cloud gaming services on the market in order to gain more insights into the scale and severity of such vulnerabilities.

So far, we only present four proof-of-concept attacks to validate the feasibility of exploiting cloud gaming services through our exposed vulnerabilities, including crypto-mining, machine learning model training, C&C, and censorship circumvention. Yet, other possible exploitation may also exist in cloud gaming services, leading to more severe damages. In the future work, we aim to conduct further research on the discovery of more sophisticated attack vectors in cloud gaming services, and more importantly, on the development of more effective defense mechanisms for protecting their valuable assets.

8 RELATED WORK

8.1 Cloud Gaming

While cloud gaming is a relatively new paradigm in the gaming ecosystem, it has received extensive research attention over the past decades. In 2009, Ross [66] first introduced the cloud gaming as a promising gaming delivery technique. Based on the cloud gaming architecture proposed in Ross's paper, pioneers have proposed and implemented many cloud gaming platforms such as OnLive [40] and GamingAnywhere [49, 50]. Ojala *et al.* [64] presented their investigation on the business model of cloud gaming services. Meanwhile, researchers have devoted countless efforts to cloud gaming in terms of improving performance [48, 60], reducing latency [30, 39, 58, 76], and enhancing the quality of service [47, 61]. Lee *et al.* [59] and Suznjevic *et al.* [70] investigated player experiences for different types of computer games in cloud gaming. More recently, Domenico *et al.* [41] analyzed network requirements for newly emerged cloud gaming services, suggesting that wired or WiFi networks could deliver a smooth gaming experience to players while the lossy and much slower cellular networks may not be sustainable.

Our work differs from existing studies as we investigate the security aspects of cloud gaming services. To the best of our knowledge, we are the first to study the feasibility of exploiting cloud gaming services and misusing their resources through computer games. The uncovered attack vectors shed lights on the security risk of cloud gaming services and will promote the development and deployment of effective countermeasures against resource misuse and service exploitation.

8.2 Cloud Security

Cloud gaming inherits similar fundamental infrastructures from traditional cloud computing [36]. For years, the security aspects of cloud computing have long been studied. Ristenpart *et al.* [65] first discovered the vulnerability of co-residency in cloud infrastructure. Later works expand the co-residency detection using both side channels [78–80] and covert channels [56, 62, 69, 73, 77]. Varadarajan *et al.* [72] proposed resource-freeing attacks (RFAs) so that attackers can gain more resources by modifying the workload of the neighboring VMs. A similar attack approach has been improved by Huang *et al.* [51], who proposed cascade attacks to overwhelm system's hardware resources. In addition, power attack [52–54, 75] and thermal attack [44, 67] have also been proposed to throttle the performance or even shut down the cloud services due to insufficient power delivery and cooling capacity. Moreover, previous works have covered other special types of cloud services, including exploit as a service [45], impersonation as a service [37], reputation escalation as a service [74], and everything as a service [43].

Our work complements existing studies by investigating the security aspect of cloud gaming, a special type of cloud service known as Gaming-as-a-Service (GaaS). We demonstrate that it is feasible and profitable for adversaries to exploit gaming services offered in the cloud environment, imposing a serious threat to GaaS.

9 CONCLUSION

In this paper, we conduct an in-depth study on the security vulnerabilities of cloud gaming services. Due to their lower cost and richer computing/network resources than the traditional cloud computing solutions, cloud gaming services could easily become an attractive target of adversaries and thus vulnerable to resource misuse and service exploitation. We reveal that adversaries can exploit cloud gaming services and misuse their resources by injecting malicious programs/URLs via game mods for malicious purposes. To demonstrate the serious security threats posed by such vulnerabilities, we conduct four proof-of-concept attacks including crypto-mining, training machine learning model, establishing Command & Control, and censorship circumvention. Finally, we present effective defense mechanisms to protect cloud gaming services against malicious resource misuses.

ACKNOWLEDGEMENT

We would like to thank the anonymous reviewers for their detailed and insightful comments, which help to improve the quality of this paper. This work is partially supported by the National Science Foundation (NSF) Grants CNS-1815650 and CNS-2054657, the Army Research Office (ARO) Grant W911NF1910049, the Commonwealth Cyber Initiative, and an Internet Freedom Fund from the Open Technology Fund (OTF). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding agencies.

REFERENCES

[1] 2018. Valve removes Steam game after allegations of hidden cryptocurrency miner. https://arstechnica.com/gaming/2018/07/valve-removes-steam-game-after-allegations-of-hidden-cryptocurrency-miner/.

[2] 2020. COVID-19 Has Increased Interest in Cloud Gaming Services. https://www.pcmag.com/news/covid-19-has-increased-interest-in-cloud-gaming-services.

[3] 2020. Report: Gaming Industry Value To Rise 30%. https://www.forbes.com/sites/mattgardner1/2020/09/19/gaming-industry-value-200-billion-fortnite-microtransactions/.

[4] 2021. Cloud Gaming Market - Growth, Trends, COVID-19 Impact, and Forecasts. https://www.reportlinker.com/p06101217/Cloud-Gaming-Market-Growth-Trends-COVID-19-Impact-and-Forecasts.html.

[5] 2021. GPU Availability and Pricing Update: November 2021. https://www.techspot.com/article/2369-gpu-pricing-2021-update/.

[6] 2021. Nvidia GeForce Now could be an answer to the GPU shortage. https://www.techadvisor.com/news/game/geforce-now-gpu-shortage-3803897/.

[7] 2021. Nvidia Is Doubling Down on a Massive Opportunity. https://www.nasdaq.com/articles/nvidia-is-doubling-down-on-a-massive-opportunity-2021-10-27.

[8] 2021. Random Bans for "Crypto mining". https://forum.shadow.tech/activation-account-billing-31/random-bans-for-crypto-mining-3382.

[9] 2021. Shadow Cryptocurrency Ban Situation. https://forum.shadow.tech/activation-account-billing-31/shadow-cryptocurrency-ban-situation-3062.

[10] 2021. Steam just reached 50,000 total games listed. https://www.pcgamesn.com/steam/total-games.

[11] 2022. Amazon EC2 On-Demand Pricing. https://aws.amazon.com/ec2/pricing/on-demand/.

[12] 2022. Ethereum, Monero Mining Profitability historical chart. https://bitinfocharts.com/comparison/mining_profitability-eth-xmr.html.

[13] 2022. LoudPlay. https://www.loudplay.ru/.

[14] 2022. ModDB. https://www.moddb.com/.

[15] 2022. Mods on Steam: Community-made Content For Your Favorite Games. https://store.steampowered.com/about/communitymods/.

[16] 2022. Nexus Mods and Community. https://www.nexusmods.com/.

[17] 2022. NiceHash. https://www.nicehash.com/.

[18] 2022. Nvidia Geforce Now. https://www.nvidia.com/geforce-now/.

[19] 2022. Shadow Cloud Computing. https://shadow.tech/.

[20] 2022. Steam. https://store.steampowered.com/.

[21] 2022. Steam Breaks Record For Most Concurrent Users With Nearly 28 Million Players Online. https://www.gamespot.com/articles/steam-breaks-record-for-most-concurrent-users-with-nearly-28-million-players-online/1100-6499277/.

[22] 2022. The best gaming PC in 2022. https://www.pcgamer.com/best-gaming-pc/.

[23] 2022. The best gaming PCs in 2022. https://www.tomsguide.com/us/best-gaming-pc,review-2219.html.

[24] 2022. Transport Fever 2. https://www.transportfever2.com/.

[25] 2022. Transport Fever 2 Upload a Mod. https://www.transportfever2.com/wiki/doku.php?id=modding:publishing.

[26] 2022. Unity. https://unity.com/.

[27] 2022. WebMinePool. https://webminepool.com/.

[28] 2022. WireShark. https://www.wireshark.org/.

[29] Amit Seal Ami, Nathan Cooper, Kaushal Kafle, Kevin Moran, Denys Poshyvanyk, and Adwait Nadkarni. 2022. Why Crypto-detectors Fail: A Systematic Evaluation of Cryptographic Misuse Detection Techniques. In *IEEE Symposium on Security and Privacy (S&P)*.

[30] Maryam Amiri, Hussein Al Osman, Shervin Shirmohammadi, and Maha Abdallah. 2016. Toward Delay-Efficient Game-Aware Data Centers for Cloud Gaming. *ACM Transactions on Multimedia Computing, Communications, and Applications* (2016).

[31] Manos Antonakakis, Tim April, Michael Bailey, Matt Bernhard, Elie Bursztein, Jaime Cochran, Zakir Durumeric, J Alex Halderman, Luca Invernizzi, Michalis Kallitsis, et al. 2017. Understanding the Mirai Botnet. In *USENIX Security Symposium*.

[32] Manos Antonakakis, Roberto Perdisci, Yacin Nadji, Nikolaos Vasiloglou, Saeed Abu-Nimeh, Wenke Lee, and David Dagon. 2012. From Throw-Away Traffic to Bots: Detecting the Rise of DGA-Based Malware. In *USENIX Security Symposium*.

[33] Weikang Bian, Wei Meng, and Mingxue Zhang. 2020. MineThrottle: Defending against Wasm In-Browser Cryptojacking. In *Proceedings of the Web Conference (WWW)*.

[34] Hugo LJ Bijmans, Tim M Booij, and Christian Doerr. 2019. Inadvertently Making Cyber Criminals Rich: A Comprehensive Study of Cryptojacking Campaigns at Internet Scale. In *USENIX Security Symposium*.

[35] Hugo LJ Bijmans, Tim M Booij, and Christian Doerr. 2019. Just the Tip of the Iceberg: Internet-Scale Exploitation of Routers for Cryptojacking. In *ACM Conference on Computer and Communications Security (CCS)*.

[36] Wei Cai, Min Chen, and Victor CM Leung. 2014. Toward Gaming as a Service. *IEEE Internet Computing* (2014).

[37] Michele Campobasso and Luca Allodi. 2020. Impersonation-as-a-Service: Characterizing the Emerging Criminal Infrastructure for User Impersonation at Scale. In *ACM Conference on Computer and Communications Security (CCS)*.

[38] Domhnall Carlin, Jonah Burgess, Philip O'Kane, and Sakir Sezer. 2019. You Could Be Mine(d): The Rise of Cryptojacking. *IEEE Security & Privacy* (2019).

[39] Mark Claypool and David Finkel. 2014. The Effects of Latency on Player Performance in Cloud-based Games. In *Annual Workshop on Network and Systems Support for Games*.

[40] Mark Claypool, David Finkel, Alexander Grant, and Michael Solano. 2012. Thin to Win? Network Performance Analysis of the OnLive Thin Client Game System. In *Annual Workshop on Network and Systems Support for Games*.

[41] Andrea Di Domenico, Gianluca Perna, Martino Trevisan, Luca Vassio, and Danilo Giordano. 2021. A Network Analysis on Cloud Gaming: Stadia, GeForce Now and PSNow. *Network* (2021).

[42] Lucas Dixon, Thomas Ristenpart, and Thomas Shrimpton. 2016. Network Traffic Obfuscation and Automated Internet Censorship. *IEEE Security & Privacy* (2016).

[43] Yucong Duan, Guohua Fu, Nianjun Zhou, Xiaobing Sun, Nanjangud C Narendra, and Bo Hu. 2015. Everything as a Service (XaaS) on the Cloud: Origins, Current and Future Trends. In *IEEE International Conference on Cloud Computing (CLOUD)*.

[44] Xing Gao, Zhang Xu, Haining Wang, Li Li, and Xiaorui Wang. 2018. Reduced Cooling Redundancy: A New Security Vulnerability in a Hot Data Center. In *ISOC Network and Distributed System Security Symposium (NDSS)*.

[45] Chris Grier, Lucas Ballard, Juan Caballero, Neha Chachra, Christian J Dietrich, Kirill Levchenko, Panayiotis Mavrommatis, Damon McCoy, Antonio Nappa, Andreas Pitsillidis, et al. 2012. Manufacturing Compromise: The Emergence of Exploit-as-a-Service. In *ACM Conference on Computer and Communications Security (CCS)*.

[46] Geng Hong, Zhemin Yang, Sen Yang, Lei Zhang, Yuhong Nan, Zhibo Zhang, Min Yang, Yuan Zhang, Zhiyun Qian, and Haixin Duan. 2018. How You Get Shot in the Back: A Systematical Study about Cryptojacking in the Real World. In *ACM Conference on Computer and Communications Security (CCS)*.

[47] Hua-Jun Hong, De-Yu Chen, Chun-Ying Huang, Kuan-Ta Chen, and Cheng-Hsin Hsu. 2014. Placing Virtual Machines to Optimize Cloud Gaming Experience. *IEEE Transactions on Cloud Computing* (2014).

[48] M Shamim Hossain, Ghulam Muhammad, Biao Song, Mohammad Mehedi Hassan, Abdulhameed Alelaiwi, and Atif Alamri. 2015. Audio-Visual Emotion-Aware Cloud Gaming Framework. *IEEE Transactions on Circuits and Systems for Video Technology* (2015).

[49] Chun-Ying Huang, Kuan-Ta Chen, De-Yu Chen, Hwai-Jung Hsu, and Cheng-Hsin Hsu. 2014. GamingAnywhere - The First Open Source Cloud Gaming System. *ACM Transactions on Multimedia Computing, Communications, and Applications* (2014).

[50] Chun-Ying Huang, Cheng-Hsin Hsu, Yu-Chun Chang, and Kuan-Ta Chen. 2013. GamingAnywhere: An Open Cloud Gaming System. In *Proceedings of ACM Multimedia Systems conference (MMSys)*.

[51] Qun Huang and Patrick PC Lee. 2013. An Experimental Study of Cascading Performance Interference in a Virtualized Environment. *ACM SIGMETRICS Performance Evaluation Review* (2013).

[52] Mohammad A Islam and Shaolei Ren. 2018. Ohm's Law in Data Centers: A Voltage Side Channel for Timing Power Attacks. In *ACM Conference on Computer and Communications Security (CCS)*.

[53] Mohammad A Islam, Shaolei Ren, and Adam Wierman. 2017. Exploiting a Thermal Side Channel for Power Attacks in Multi-Tenant Data Centers. In *ACM Conference on Computer and Communications Security (CCS)*.

[54] Mohammad A Islam, Luting Yang, Kiran Ranganath, and Shaolei Ren. 2018. Why Some Like It Loud: Timing Power Attacks in Multi-tenant Data Centers Using an Acoustic Side Channel. *Proceedings of the ACM on Measurement and Analysis of Computing Systems (SIGMETRICS)*.

[55] Lin Jin, Shuai Hao, Haining Wang, and Chase Cotton. 2022. Understanding the Practices of Global Censorship through Accurate, End-to-End Measurements. In *Proceedings of the ACM on Measurement and Analysis of Computing Systems (SIGMETRICS)*.

[56] Beom Heyn Kim and David Lie. 2015. Caelus: Verifying the Consistency of Cloud Services with Battery-Powered Devices. In *IEEE Symposium on Security and Privacy (S&P)*.

[57] Radhesh Krishnan Konoth, Emanuele Vineti, Veelasha Moonsamy, Martina Lindorfer, Christopher Kruegel, Herbert Bos, and Giovanni Vigna. 2018. MineSweeper: An In-depth Look into Drive-by Cryptocurrency Mining and Its Defense. In *ACM Conference on Computer and Communications Security (CCS)*.

[58] Kyungmin Lee, David Chu, Eduardo Cuervo, Johannes Kopf, Yury Degtyarev, Sergey Grizan, Alec Wolman, and Jason Flinn. 2015. Outatime: Using Speculation to Enable Low-Latency Continuous Interaction for Mobile Cloud Gaming. In *ACM International Conference on Mobile Systems, Applications, and Services (MobiSys)*.

[59] Yeng-Ting Lee, Kuan-Ta Chen, Han-I Su, and Chin-Laung Lei. 2012. Are All Games Equally Cloud-Gaming-Friendly? An Electromyographic Approach. In *Annual Workshop on Network and Systems Support for Games*.

[60] Xiaofei Liao, Li Lin, Guang Tan, Hai Jin, Xiaobin Yang, Wei Zhang, and Bo Li. 2015. LiveRender: A Cloud Gaming System Based on Compressed Graphics Streaming. *IEEE/ACM Transactions on Networking* (2015).

[61] Yuhua Lin and Haiying Shen. 2016. CloudFog: Leveraging Fog to Extend Cloud Gaming for Thin-Client MMOG with High Quality of Service. *IEEE Transactions on Parallel and Distributed Systems* (2016).

[62] Ramya Jayaram Masti, Devendra Rai, Aanjhan Ranganathan, Christian Müller, Lothar Thiele, and Srdjan Capkun. 2015. Thermal Covert Channels on Multi-core Platforms. In *USENIX Security Symposium*.

[63] Faraz Naseem Naseem, Ahmet Aris, Leonardo Babun, Ege Tekiner, and A Selcuk Uluagac. 2021. MINOS: A Lightweight Real-Time Cryptojacking Detection System. In *ISOC Network and Distributed System Security Symposium (NDSS)*.

[64] Arto Ojala and Pasi Tyrvainen. 2011. Developing Cloud Business Models: A Case Study on Cloud Gaming. *IEEE Software* (2011).

[65] Thomas Ristenpart, Eran Tromer, Hovav Shacham, and Stefan Savage. 2009. Hey, You, Get Off of My Cloud: Exploring Information Leakage in Third-Party Compute Clouds. In *ACM Conference on Computer and Communications Security (CCS)*.

[66] Philip E Ross. 2009. Cloud Computing's Killer App: Gaming. *IEEE Spectrum* (2009).

[67] Zhihui Shao, Mohammad A Islam, and Shaolei Ren. 2019. A First Look at Thermal Attacks in Multi-Tenant Data Centers. *ACM SIGMETRICS Performance Evaluation Review* (2019).

[68] Brett Stone-Gross, Marco Cova, Lorenzo Cavallaro, Bob Gilbert, Martin Szydlowski, Richard Kemmerer, Christopher Kruegel, and Giovanni Vigna. 2009. Your Botnet is My Botnet: Analysis of a Botnet Takeover. In *ACM Conference on Computer and Communications Security (CCS)*.

[69] Dean Sullivan, Orlando Arias, Travis Meade, and Yier Jin. 2018. Microarchitectural Minefields: 4K-Aliasing Covert Channel and Multi-Tenant Detection in IaaS Clouds. In *ISOC Network and Distributed System Security Symposium (NDSS)*.

[70] Mirko Suznjevic, Justus Beyer, Lea Skorin-Kapov, Sebastian Moller, and Nikola Sorsa. 2014. Towards Understanding the Relationship Between Game Type and Network Traffic for Cloud Gaming. In *IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*.

[71] Ege Tekiner, Abbas Acar, A Selcuk Uluagac, Engin Kirda, and Ali Aydin Selcuk. 2021. SoK: Cryptojacking Malware. In *IEEE European Symposium on Security and Privacy (EuroS&P)*.

[72] Venkatanathan Varadarajan, Thawan Kooburat, Benjamin Farley, Thomas Ristenpart, and Michael M Swift. 2012. Resource-Freeing Attacks: Improve Your Cloud Performance (at Your Neighbor's Expense). In *ACM Conference on Computer and Communications Security (CCS)*.

[73] Zhenyu Wu, Zhang Xu, and Haining Wang. 2012. Whispers in the Hyper-Space: High-Bandwidth and Reliable Covert Channel Attacks Inside the Cloud. In *USENIX Security Symposium*.

[74] Haitao Xu, Daiping Liu, Haining Wang, and Angelos Stavrou. 2015. E-commerce Reputation Manipulation: The Emergence of Reputation-Escalation-as-a-Service. In *Proceedings of the Web Conference (WWW)*.

[75] Zhang Xu, Haining Wang, Zichen Xu, and Xiaorui Wang. 2014. Power Attack: An Increasing Threat to Data Centers. In *ISOC Network and Distributed System Security Symposium (NDSS)*.

[76] Roy D Yates, Mehrnaz Tavan, Yi Hu, and Dipankar Raychaudhuri. 2017. Timely Cloud Gaming. In *IEEE International Conference on Computer Communications (INFOCOM)*.

[77] Anil Yelam, Shibani Subbareddy, Keerthana Ganesan, Stefan Savage, and Ariana Mirian. 2021. CoResident Evil: Covert Communication In The Cloud With Lambdas. In *Proceedings of the Web Conference (WWW)*.

[78] Yinqian Zhang, Ari Juels, Alina Oprea, and Michael K Reiter. 2011. HomeAlone: Co-residency Detection in the Cloud via Side-Channel Analysis. In *IEEE Symposium on Security and Privacy (S&P)*.

[79] Yinqian Zhang, Ari Juels, Michael K Reiter, and Thomas Ristenpart. 2012. Cross-VM Side Channels and Their Use to Extract Private Keys. In *ACM Conference on Computer and Communications Security (CCS)*.

[80] Yinqian Zhang, Ari Juels, Michael K Reiter, and Thomas Ristenpart. 2014. Cross-Tenant Side-Channel Attacks in PaaS Clouds. In *ACM Conference on Computer and Communications Security (CCS)*.