Untrusted Platforms

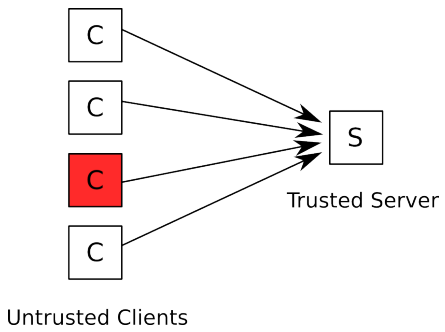
Kurt Thomas

November 23, 2011

In this lecture, we discussed the challenges of guaranteeing confidentiality, integrity, and fairness when executing code on untrusted systems. In particular, we discuss (1) cheating in online games, (2) web applications that receive client-side input, and (3) digital media copyright protection.

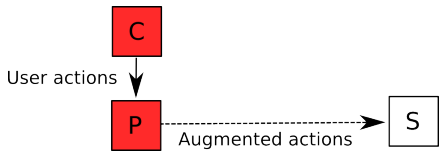
1 Cheating in Online Gaming

In online gaming, a player that cheats can potentially impact the in-game experience of other players. For first person shooters, this might entail auto-aiming or seeing through walls to gain an information advantage, while for an MMORPG it might be a bot that farms gold and mobs, marginalizing the effort put in by other players. These types of gaming advantages generalize into three categories of attacks: **augmentation, confidentiality, and integrity**. For each of these categories, we consider a scenario of multiple untrusted clients connecting to a single server where any or all of the clients may be cheating. The goal of the server is to identify any forms of cheating.



1.1 Augmentation

Augmentation describes any situation where computer-aided decision making grants a player an unfair advantage. Examples include aim bots which provide perfect targeting and instantaneous reaction time as well as fully-automated bots that control all aspects of a client's interaction with the server. The simplest method of augmentation is to replay actions or inject new packets into a client's communication with the server, typically through an application-level proxy.



Potential defenses include:

- Computing a CRC of packets to make tampering more difficult
- Encrypting the channel between the client and server
- Running a separate program alongside the client to identify tampering or known bot processes, similar to anti-virus

However, because the client has full access to the underlying game code and control over the operating system, there is only a technical challenge to reverse engineering a game's mechanics; not a security challenge. In the end, games that computers play better than humans will be susceptible to augmentation attacks.

1.2 Confidentiality

Real-time strategies and first-person shooters rely on asymmetric information, where the server is privy to all actions that transpire in a game, but only certain actions are revealed to clients. However, due to network latency, servers will often provide every action to a client and rely on the client's software to selectively render events. A prime example is fog of war, where an opponents actions are not rendered unless a player is within range to observe them. An attack on confidentiality occurs when a misbehaving client accesses information relayed by a server that was meant to remain secret, such as an opponents movements off screen.

Potential defenses include:

- Removing reliance on clients to perform selective rendering, with a potential cost to latency
- Honey tokens; servers providing false information to clients that benign clients would not operate on, but cheaters would. (E.g. announcing a player is behind a wall; a regular user would not render the character, while a cheater would believe there was an opponent nearby)

1.3 Integrity

A final set of attacks includes tampering with client-side state such as character position, health, or ammunition levels. These set of attacks are easily prevented by replicating client-side state on the trusted server and verifying the data provided by clients.

2 Web Applications

Many of the attacks that exist for online gaming also carry over to web applications.

2.1 Augmentation

Augmentation in web applications can appear in the form of sending automated friend requests in social networks, emailing spam messages, or sniping auctions on sites such as eBay.

Potential defenses include:

- Adding in tasks that only humans can solve (e.g CAPTCHAs)
- Leveling the playing field for all participants; providing sanctioned auction bots to remove the unfair advantage provided by augmentation

2.2 Confidentiality

Breaches of confidentiality for web applications may include providing private user data to JavaScript which is selectively rendered. Such data can be intercepted by hooking into a browser.

Potential defenses for this scenario are rather simple:

- Assuming that latency is a non-issue, servers should not provide sensitive data that isn't mean to be displayed or known by a client.

2.3 Integrity

An example of an integrity attack on a web application includes tampering with client-side sanitization of web forms. If an application validates the length of a string or checks the bounds on a value, a misbehaving client can easily forgo these sanitization operations and provide malformed input to the server.

Potential defenses for this scenario are rather simple:

- Perform, or re-perform, all sanitization operations on the server-side; treat all client-side input as untrusted.

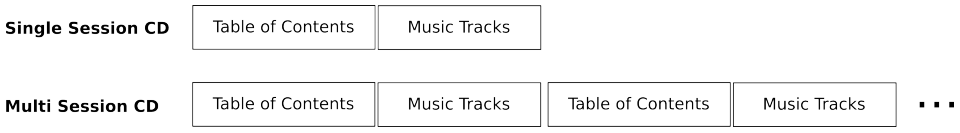
3 Copy Protection

Copy protection aims to prevent unauthorized duplication of multimedia such as movies, games, or music. Most copy protection aims to prevent **wholesale copying** by parties seeking to sell counterfeit copies, as opposed to **casual copying** used to create personal backups. However, torrents have blurred this distinction; now anyone can participate in the distribution of content with little or no loss of quality. While there are many forms of copy protection, lecture specifically covered CDs and DVDs.

3.1 CDs

Sony Rootkit In order to prevent CDs from being ripped, at one point Sony installed a rootkit on a user's computer. As soon as the CD was inserted, an autorun program would install a program that intercepted all requests to the CD-ROM, actively preventing requests performed by ripping software.

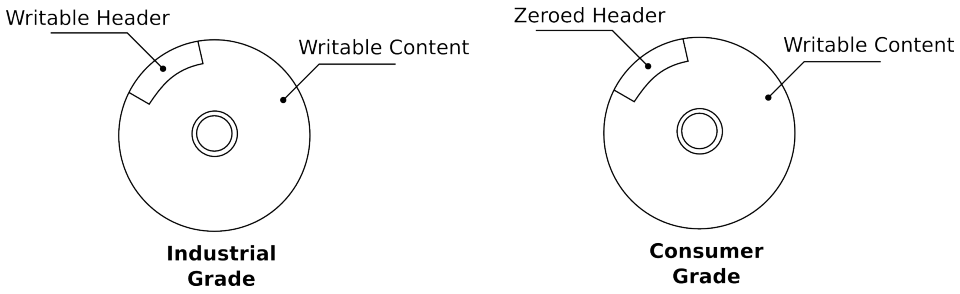
Malformed Headers A second approach to copy protection was to abuse a discrepancy between the way CD-ROMs read CDs compared to personal or vehicle CD players. In particular, the attack leveraged multi-session CDs. For a single session CD, a table of contents provides an index for future tracks, but the CD can only be written once. For multi-session CDs, future tracks can be appended by simply appending a new table of contents and the associated data, shown below.



By specifically crafting a malformed table of contents, CD producers were able to create multi-track CDs that would cause CD-ROM readers to be unable to read a CD; CD players were able to tolerate the error and simply skip the malformed table of contents and move on to the next valid table of contents. To circumvent this protection, a person could simply take a marker and draw over the CD where the malformed segment was written. This would cause the CD-ROM to correctly interpret the region as an error and silently jump to the next table of contents instead of crashing.

3.2 DVDs

To combat copying DVD content, the film industry came up with DeCSS, an encryption scheme for DVD content. The following DVD figure is used as a reference.



- Content of a DVD was encrypted with a per-disk key K and written to the content portion of a DVD as $Enc_K(m)$
- Manufacturers would embed a player key PK_i inside their DVD players, with a single key existing per manufacturer.
- In order for DVDs to be readable by any official device, the header region of a DVD would include $Enc_{PK_1}(K)$ ... $Enc_{PK_n}(K)$, a list of disk keys encrypted with every player key.

- Individual player keys allowed keys to be revoked on a per-device basis if the key was ever leaked. This threat was used to force manufacturers to make reverse engineering difficult.

- Consumer grade DVDs could not replicate this header region because they were sold pre-written with zeros.

DeCSS was eventually cracked due to a mathematical weakness that allowed an attacker to recover the key from a DVD's ciphertext. Furthermore, only a 40-bit key was used for legal export reasons, which by today standards is susceptible to brute force attacks, had the cryptographic weakness not existed.

While using a custom crypto protocol is frowned on by security experts, there is a legal motivation for doing so.

- Custom algorithm forces DVD manufacturers to license technology. These agreement contains requirements to make reverse engineering devices to recover their player key difficult.

- Manufacturers were also forced to adhere to region codes within DVDs, preventing disks sold in the US from working in other countries. This allowed staged releases.

Blue-ray continues to encrypt all disk content, but with a new protocol, discussed next lecture.