A Practical Implementation of a Multi-Device Split Application for Protecting Online Poker

Gabriel Kaptchuk and Aviel Rubin
Johns Hopkins University
[gkaptchuk, rubin]@cs.jhu.edu

We designed and implemented an online poker site that allows users to protect themselves against malware. The user can elect to receive the secret hole cards on an external device such as a smartphone, instead of on the computer. We describe our solution and our implementation. We have deployed a test “play money” online poker site where users can experiment with our implementation and play against other people.

Introduction

The online poker industry is growing rapidly, representing vast sums of money in transactions every year. While casinos have developed powerful ways to enforce the rules of the game, online poker has proved more difficult to secure. In the past couple years there have been many attacks. In particular, a malware uncovered this past year exposed a distinct new threat that is specific to online poker. The Win32/Spy.Odlanor malware package was discovered in September 2015, and reportedly affected hundreds of poker players (Lipovsky, 2015). The malware searches infected computers for online poker applications. If the victim is playing online poker, the malware captures screen shots and sends them to a foreign server, where the attacker monitors the infected player's cards. While this not the first attack of its kind, it is the first time that it has been seen on this scale.

To address the problem of malware such as Odlanor, we propose splitting applications such as online poker into sensitive and non-sensitive components. The non-sensitive components are run on a user's primary device, such as his or her computer, and the sensitive portion runs on a second device, such as the user's smartphone or tablet. Such devices are ubiquitous today, and it's normally safe to assume that online poker players possess one of them. The main idea in our solution is that an attacker may succeed in compromising a user's computer with malware but would gain no advantage in a poker game if the user's secondary device were not also compromised. We believe that it is much more difficult for an attacker, such as the creator of the Odlanor malware to successfully compromise both a user's computer and that user's smartphone, and to be able to link the two to the same person, than to just spread the malware on the Internet.

Specifically, the way that the creators of malware such as Odlanor attack systems is to spread their malware as widely as possible via typical attack vectors such as downloading them from malicious web pages or via malicious email messages. It is a shotgun approach, as the malware hits many random sites and does not target specific users. The malware then searches the infected machines for signs of online poker accounts. Compare this to the access required by an attacker who needs to not only compromise the machine on which a user is playing poker, but also to find that user's smartphone and to attack that phone at the same time.

We believe that splitting applications in this manner is useful as a generalized approach to securing against certain malware threats and that there applications beyond online poker, such as banking and managing online healthcare records. However, in this paper, we focus on describing the online poker solution, which we have implemented.

Online Poker

Poker is a popular betting game that uses traditional playing cards. The game combines skill, strategy, and an element of luck. The most popular variant of the game is called Texas Hold'em where multiple players sit around a table and are dealt two cards each. These are called hole cards and are kept secret. After the hole cards are dealt, the players make bets, and then the dealer deals three cards face up. These communal cards are called *the flop*. Next, the *turn* and the *river* are dealt face up, with each round of cards followed by betting. At any point, the players may fold, but if at least two players still have cards when the final round of betting completes, then there is a *showdown*, where the remaining players must show their hole cards. The player that makes the best five-card poker hand wins the pot.

It is important to note that a key to the game is the secrecy of one's hole cards. As the players compete and place bets, they try to deduce the possible range of hands that their opponents might hold based on their betting patterns, the number of hands that they play, and their assessments of each others' playing styles

and skill level. Top professionals have an uncanny ability to make accurate assessments as to the hole cards of their opponents.

Unfortunately, games such as poker that involve real-money wagering always attract cheaters. One common technique found in home games, and even occasionally in casinos is the marking of cards. Intrepid swindlers have tried everything from putting a small ink smudge on important cards, such as aces and kings, to manufacturing entire card decks with special markings that only the attackers know about. There have even been cases of cards being marked with ink only visible when wearing specially designed sunglasses, allowing cheaters to know the hole cards of the other players. Obviously, one can play nearly perfect poker when seeing the hole cards of the other players. Cheating is a serious problem in brick and mortar poker. Online poker presents its own set of security challenges.

Online poker has been growing, both in the numbers of players and the number dollars trading hands. Although exact numbers for the size of the market differ, estimates using data from the Online Poker Database of the University of Hamburg from 2009-2010 showed that the nearly 1.5 million online poker players in the United States alone were responsible for almost $1 billion dollars in transactions annually (Fiedler, 2012). Other surveys found that nearly 40 million adults play online poker regularly world wide (PokerPlayers Research Group, 2010). Websites like pokerscouts.com track daily online poker activity, indicating tens of thousands of unique players on a normal day, with tens of thousands more playing in play-money rooms (Poker Scout, 2015). Most of these players are accessing online poker applications via a client on their desktop or laptop computer, which are easy targets for malware.

When poker is played online, a trusted central server manages the randomness of the cards. With the vast sums of money involved, online poker has become a ripe target for cheating. The attacks no longer involve marking cards, as in the physical world, but instead utilize the properties of online poker that are different from playing at a physical table. For example, when playing online, it is impossible to detect whether or not two players are speaking with each other and sharing information, such as their hole cards, that they are not allowed to discuss in a casino. This form of cheating, called collusion, can give players enough of an edge that they can consistently beat the competition.

While collusion and attacks against the randomness of the shufflers, have been shown to exist and actually be exploited, an increasing number of attacks have surfaced targeting users' computers directly using malware (Arkin, 1999). One recent case involved a disgruntled insider at Absolute Poker who found a way to view opponent's hole cards through an exploit in the system software (Levitt, 2007). There have also been a number of direct attacks on professional poker players (Angioni, 2014; Goodlin, 2013; DellaFave, 2013). These attacks use malware manually installed on a player's computer to allow a cheating player to view opponents' cards remotely.

Hackers also developed plug and play malware packages for non-technical poker cheats (Adham, 2013). The most recent of these was the Win32/Spy.Olandor program, which does not appear to have an intended target. All of these attacks take advantage of the simple fact that the entire context of a poker game is easily available to an attacker who can gain access to the screen of an opponent.

Attacks on professional poker players are thought to be executed by remote desktop access and most seem to have been very carefully targeted at the individual players (Angioni, 2014; Goodlin, 2013; DellaFave, 2013). Win32/Spy.Olandor is the first time that the poker community has seen this type of attack spread widely by malware. The text snippet in Figure 1 shows part of the Win32/Spy.Olandor source code responsible for detecting an active poker client. It illustrates that this piece of malware is undoubtedly targeted directly at poker players and can also sense many different client applications. Targeting the general poker playing public instead of professional players makes it more unlikely that the authors of the malware would be caught by any of their activities in game. While an attacker might be flagged for cheating if their in-game choices were noticeably uncharacteristic, an inexperienced player is unlikely to notice if his or her opponent was acting strangely.

```
78  v6 = GetParent(hwnd);
79  _GetClassNameA(v6, &v27, 512);
80  _GetWindowTextA(v6, &v28, 512);
81  memset(&pwi, 0, sizeof(pwi));
82  pwi.cbSize = 60;
83  GetWindowInfo(hwnd, &pwi);
84  if ( !strcmp(&ClassName, "PokerStarsTableFrameClass") )
85  {
86    fPokerStars = 1;
87    basic_string_create_0((int)&v33, &sz_window_text, (int)&v52 + 2);
88    dwState = 1;
89    basic_string_create_0((int)&v31, "PokerStars: ", (int)&v52 + 1);
90    dwState = 2;

242 if ( !strcmp(&ClassName, "QWidget") && !strcmp(&sz_window_text, "FullTiltPoker")
243   f_full_tilt_poker_found = 1;
```

Figure 1: Code fragment for the Win32/Spy.Olandor Malware

Related Work

Research efforts in securing online poker have focused on the problem of collusion and of detecting automatic players, also known as bots. While individual online poker platforms may be vulnerable to traditional forms of software exploitation (buffer overflows or poor use of cryptography), automatic players and collusion are two forms of cheating that are unique to poker.

Automatic Players

This year, a team of researchers produced an optimal algorithm for a variant of Texas Hold'Em called *heads-up limit*, where bet sizes are fixed and only two players participate (Bowling et. al, 2015). Multi-player no-limit hold'em is orders of magnitude more complex, and the research community studying this game is not even close to having an optimal strategy. Thus, in the most popular variant of the game, having access to large amounts of data and computational power offers an advantage. Significant effort has been put into creating automatic poker players that have an edge over human players due to the massive amounts of data an automatic player can crunch. Automatic players that can consistently win small sums also offer their owners a continuous source of income with no effort beyond setup. These efforts include large software projects like openholdembot and popular competitions like MIT Pokerbots (OpenHoldemBot 2015; MIT PokerBots, 2015).

Automatic players threaten the integrity of online poker systems and are often prohibited on real-money online poker sites. Because the rewards for successful bot creation are so high, significant research has focused on bot detection in online poker. One method of keeping bots from playing online poker is to prevent them from accessing the system entirely. Golle has discussed preventing access to a game by validating humanity at login and then continuing to validate humanity through game play (Golle, 2005). Multiple different implementations of CAPTCHA have attempted to address this problem as well. One displays cards to users in forms that are human readable forms but would challenge the optical recognition technologies used by bots to read a table (Yampolskiy, 2008). Another presented images of human faces or avatar equivalents to prove their humanity (D'Souza, 2012). These more complex CAPTCHA systems attempt to address the increasing ease with which bots are able to solve traditional CAPTCHAs.

Collusion

Collusion in a physical poker game requires signals and subterfuge. In fact, in high dollar live tournaments, after a certain stage, all electronics are banned, as an added defensive measure against collusion. In online systems, there are a few mechanisms deployed to address collusion. For example, in tournaments, players are assigned to tables randomly; making it less likely that potentially colluding players will be together. Some sites go so far as to try to keep apart players with geographically close IP

addresses. However, there are easy countermeasures to this, and furthermore, not all players who wish to collude are necessary in the same state or country.

Researchers have attempted to develop poker protocols specifically aimed at limiting the effects of player coalitions (Crepeau, 1985). Methods for creating a player behavior profile as a method for detecting a compromised account could also be used to detect when a known player performs abnormal actions that may be a sign that they are colluding (Yampolskiy, 2006). Other work has been done on more generic forms of collusion, beyond the scope of just online poker including Yan's work on online bridge and Smed's work on the online video game Age of Empires; both of these solutions may have applications to online poker (Yan, 2003; Smed, 2003). While this form of cheating is a threat in online poker, collusion detection techniques are aimed at discovering multiple players working together, not one player spying on another. Thus, the solutions developed to address problems of collusion are insufficient to protect against malware like Win32/Spy.Odlanor.

Two Factor Authentication (2FA)

Our solution to the online poker malware problem involves splitting the poker application into sensitive and non-sensitive components that run on separate devices. This is analogous to two factor authentication (2FA), in which a user relies on a secondary device to authenticate to a service. However, 2FA only protects the initial authentication, and if a computer is infected with malware, the attacker still has full access to the machine and its applications.

For example, if an online poker site utilized 2FA to log users into their online poker account, then a text message could be sent to a player's phone with a code, and he or she would have to type in that code on their computer to get access. While the 2FA prevents attackers who do not have the user's phone from being able to log into the system, there is no protection in this process against malware running on the user's machine, as the game progresses.

Furthermore, 2FA itself has been shown to be vulnerable to attack. One of the main vulnerabilities allows an attacker to gain full access to one or both devices used in a 2FA login, via malware (Dmitrienko et. al, 2014). The researchers exploited low entropy in a one time pad and disabled 2FA using the authenticated session. Other researchers were able to leverage rootkits in browsers to execute automatic fraudulent transactions even in the presence of 2FA (Adham, 2013).

Visual Cryptography

Visual cryptography is a technology where an image can be split into two or more *masks* such that no one mask reveals any information about the initial image. Only when all of the masks are stacked and aligned can the initial image be recovered (Naor and Shamir, 1994). Although the initial work of Naor and Shamir was extended to different kinds of images, visual cryptography systems are rarely used in practice. However, we believe that there may be ways to incorporate this technology into building more user friendly version of our system in the Future Work section.

Implementation

We developed our online poker system from the open source online game system HoldingNuts, a C++ project developed in 2009-2011 (HoldingNuts, 2015). We built an iOS application for the iPhone to hold the display cards and modified the server and the client to support the splitting of the application into sensitive and non-sensitive components. Our code is freely available at https://github.com/gkaptch1/SplitPoker_pubilc, and we plan to update it regularly with improvements. We will attempt to develop a community that will use our play money site that can provide us feedback on their experiences so that we can improve the usability of the system.

We created a mode of play in our implementation called *protected* mode. In this mode, the player's cards are not displayed in the client, and they are not event sent from the server to the client. Instead, they are sent to the player's mobile device, which was previously registered with the server.

Server Design

We added a variable in the class representing a player sitting at a poker table to indicate if that player was in protected mode. We implemented an extension to the messaging protocol to toggle this option on and off. These new messages are sent by the iPhone upon registration. We implemented logic in the server to determine whether or not a player is in protected mode and to send the hole cards to the appropriate device based on that setting.

Below is an excerpt of some of the code we added to the server code to send the hole cards to the correct device.

```
if(client_in_protected_mode(p->client_id)) {
  //Pull the names of the new cards into the char buffers
  strcpy(card1, c1.getName());
  strcpy(card2, c2.getName());

  //Generate the string to send to the device
  snprintf(msg, sizeof(msg), "%d %s %s",
              SnapCardsHole, card1, card2);

  //Send off the hole cards to the protected device
  protected_snap(p->client_id, t->table_id, SnapCards, msg);

  //Write over the cards with "PP", so the client can see cards
  strcpy(card1, c1.getProtectedName());
  strcpy(card2, c2.getProtectedName());

  //Rewrite the buffer to contain "PP" instead of the card info
  snprintf(msg, sizeof(msg), "%d %s %s",
              SnapCardsHole, card1, card2);

} else {
  strcpy(card1, c1.getName());
  strcpy(card2, c2.getName());
  snprintf(msg, sizeof(msg), "%d %s %s",
              SnapCardsHole, card1, card2);
}
//We send out hole information to clients. If it's protected,
//the card information itself will have been wiped above.
snap(p->client_id, t->table_id, SnapCards, msg);
```

The final change required on the server was to assign an optional secondary device connection field to each player entity. When a device that belonged to a known player registers with the system, the connection is stored in that player's object.

Client Design

We added an image data file to represent the hole cards of a player in protected mode. We chose to make a number of enhancements to the HoldingNuts user interface into improve the usability of the client. For debugging, we also added a way for the client to toggle the player in and out of protected mode.

Figure 2: Client view of their table while in *protected* mode. The cards cannot be seen on the desktop and in fact are not known even to the computer. The players use an iPhone to see the hole cards.

We chose to design our secondary device software for iOS, particularly aimed at the iPhone, although the design can easily be used to make an app for any secondary device. The HoldingNuts system allows player to log into the system using a self-chosen universally unique identifier (UUID). This value is generated randomly upon installation, but can be changed in a configuration file within the file system of the client. When the secondary device registers with the system, the user matches the UUID of the device to that of the desired client. The server, upon receiving a registration message from the device, checks the list of known player for a UUID match and stores the device connection within a matching player. If no such player exists, the connection is rejected and torn down with an appropriate error message. While the UUID's generated by the clients are long to prevent collisions, they are not required to be any length, so shorter UUIDs can be used for convenience, but are not recommended for any system with real security requirements. Clearly using only a UUID to associate a device with an active play is not sufficiently secure for this system to be deployed and used for real money; a production level system would use additional authentication methods as well as encryption to ensure only the real user could associate their secondary device.

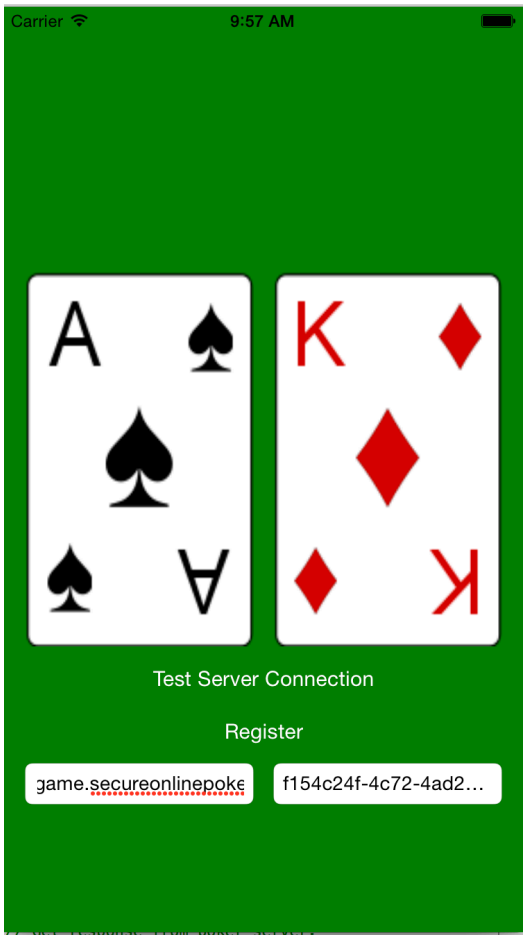

Figure 3: The iOS application running on a simulator. The "Test Server Connection" button makes sure it is possible to connect to the listed server and that the cards are rendering on the iPhone screen. The "Register" button attaches the device to the player with the appropriate UUID.

Our user interface, show in Figure 3, is simple. It shows two slots for displaying hole cards, the user's UUID and the desired game server's hostname or IP address. The user interface also has two buttons, one to test the server connection without registering and a second one to initiate the connection and registration process with the server.

Upon clicking the registration, the device sends a registration message to the server. If the message is acknowledged, the connection remains open and the device listens asynchronously for incoming messages. When a message containing hole card arrives from the server, the user interface updates and displays the new hole cards. Disconnecting the device from the server takes the player out of protected mode by the server, and the player can continue to play online poker the tradition way, with the hole cards visible on the computer screen. Disconnecting must be initiated on the iPhone, so malware on the user's computer cannot take the player out of protected mode.

Bot Players

We are developing bots to be used in our implementation and will deploy them when they are ready. While the poker community, as described above, views use of bots in certain contexts negatively we plan to deploy ours such that users can easily test the system. Users will be explicitly choosing to play against bots. The purpose of presenting our system publicly is to get feedback from the community about the usability of our approach, where hole cards are not on the computer, and the use of bots will allow anyone to test the system at any time, no matter the traffic on the site. These bots will make probabilistic

decisions based on their hole cards and the texture of the table. They will run as separate processes on the server machines that are started at the request of a user.

Hand Histories

Online poker systems today, such as PokerStars, WSOP Poker, and FullTiltPoker maintain records of all of the hands that a user plays. The hand histories are stored on the player's local machine. A user's hand history is considered extremely valuable, as it may give an opponent insight into his or her strategy. Professional players and high-stake players in particular worry about public exposure of their hand histories. This data provides an inside look into the player's thinking and playing style, and professional poker players with whom we have spoken indicated that it would be a disaster if their hand histories were published.

```
1  PokerStars Game #27738502010: Tournament #160417133, $0.25+$0.00
Hold'em No Limit - Level XV (250/500) - 2009/05/02 13:32:38 ET
2  Table '160417133 3' 9-max Seat #8 is the button
3  Seat 1: LLC 4Eva (9182 in chips)
4  Seat 2: 618shooter (25711 in chips) is sitting out
5  Seat 3: suposd2bRich (21475 in chips)
6  Seat 4: ElT007 (60940 in chips)
7  Seat 5: Orlando I (18044 in chips)
8  Seat 6: ih82bcool2 (8338 in chips)
9  Seat 7: kovilen007 (8353 in chips)
10 Seat 8: GerKingTiger (4404 in chips)
11 Seat 9: Phontaz (23553 in chips)
12 LLC 4Eva: posts the ante 60
13 618shooter: posts the ante 60
14 suposd2bRich: posts the ante 60
15 ElT007: posts the ante 60
16 Orlando I: posts the ante 60
17 ih82bcool2: posts the ante 60
18 kovilen007: posts the ante 60
19 GerKingTiger: posts the ante 60
20 Phontaz: posts the ante 60
21 Phontaz: posts small blind 250
22 LLC 4Eva: posts big blind 500
23 *** HOLE CARDS ***
24 Dealt to ElT007 [Qd Qc]
25 618shooter: folds
26 suposd2bRich: folds
27 ElT007: raises 2000 to 2500
28 Orlando I: raises 15484 to 17984 and is all-in
29 ih82bcool2: folds
30 kovilen007: calls 8293 and is all-in
31 GerKingTiger: folds
32 Phontaz: calls 17734
33 LLC 4Eva: folds
34 ElT007: raises 15484 to 33468
35 Phontaz: calls 5509 and is all-in
36 Uncalled bet (9975) returned to ElT007
37 *** FLOP *** [2d 2c 3c]
38 *** TURN *** [2d 2c 3c] [8h]
39 *** RIVER *** [2d 2c 3c 8h] [4d]
40 *** SHOW DOWN ***
41 Phontaz: shows [9s 9h] (two pair, Nines and Deuces)
42 ElT007: shows [Qd Qc] (two pair, Queens and Deuces)
43 618shooter has returned
44 ElT007 collected 11018 from side pot-2
45 Orlando I: shows [5d 5h] (two pair, Fives and Deuces)
46 ElT007 collected 29073 from side pot-1
47 kovilen007: shows [Kh As] (a pair of Deuces)
48 ElT007 collected 34212 from main pot
```

Figure 4: A sample hand history in the PokerStars format

Most hand histories are kept in simple, standardized text files shown in Figure 4. They contain the username and stack size of all players at the table and all the cards that were revealed during the hand. Our threat model in this paper assumes that an attacker is able to get arbitrary malware onto the user's computer, so the attacker can easily extract the hand history file. In practice no poker sites today encrypt the hand history. But even if the hand histories were encrypted, an attacker would be able to either access the user's key or directly access the data when the user to reviews it.

We present a mechanism for protecting local hand histories when our two-device solution is employed. The idea is to store the hand histories in encrypted form on the user's computer. Thus, any malware on the computer would be unable to access the details of past poker hands. In this solution, the decryption key is only available on the user's secondary device, such as an iPhone.

Hand histories in online poker system are often viewed by inputting the text of Figure 4 into a simulator, which plays out a video of the hand, imitating the way it was originally viewed on an online poker table. In the secure version, the simulator would play back the hand (none of the non-sensitive information is encrypted), and would display “PROTECTED” in the space where the hole cards would normally appear. The encrypted hole cards would be sent to the user's phone, where they would be decrypted and displayed to the user in real time, as the hand plays out on the computer. Figure 5 shows what the hand history text file would look like on the user's computer.

```
1  PokerStars Game #27738502010: Tournament #160417133, $0.25+$0.00
Hold'em No Limit - Level XV (250/500) - 2009/05/02 13:32:38 ET
2  Table '160417133 3' 9-max Seat #8 is the button
3  Seat 1: LLC 4Eva (9182 in chips)
4  Seat 2: 618shooter (25711 in chips) is sitting out
5  Seat 3: suposd2bRich (21475 in chips)
6  Seat 4: ElT007 (60940 in chips)
7  Seat 5: Orlando I (18044 in chips)
8  Seat 6: ih82bcool2 (8338 in chips)
9  Seat 7: kovilen007 (8353 in chips)
10 Seat 8: GerKingTiger (4404 in chips)
11 Seat 9: Phontaz (23553 in chips)
12 LLC 4Eva: posts the ante 60
13 618shooter: posts the ante 60
14 suposd2bRich: posts the ante 60
15 ElT007: posts the ante 60
16 Orlando I: posts the ante 60
17 ih82bcool2: posts the ante 60
18 kovilen007: posts the ante 60
19 GerKingTiger: posts the ante 60
20 Phontaz: posts the ante 60
21 Phontaz: posts small blind 250
22 LLC 4Eva: posts big blind 500
23 *** HOLE CARDS ***
24 Dealt to ElT007 W1FkIFFjXQ==
25 618shooter: folds
26 suposd2bRich: folds
27 ElT007: raises 2000 to 2500
28 Orlando I: raises 15484 to 17984 and is all-in
29 ih82bcool2: folds
30 kovilen007: calls 8293 and is all-in
31 GerKingTiger: folds
32 Phontaz: calls 17734
33 LLC 4Eva: folds
34 ElT007: raises 15484 to 33468
35 Phontaz: calls 5509 and is all-in
36 Uncalled bet (9975) returned to ElT007
37 *** FLOP *** [2d 2c 3c]
38 *** TURN *** [2d 2c 3c] [8h]
39 *** RIVER *** [2d 2c 3c 8h] [4d]
40 *** SHOW DOWN ***
41 c2lvbiBpbmVbW1vZGUgcHJlc2VydmVkLiBPdXIgc2lkZSBmYWlsZTGZpbmQgbGlr
42 ZSBub2cuIERpc2NvdmVyZWQgdHJhdmVsbGluZyBmb3IgaW5zZW5zaWJsZSBwYXJ0
43 aWFsaXR5IHVucGxlYXNpbmcgaW1wb3NzaWJsZSBzaGUuIFN1ZGRlbiB1cCBoSBl
44 eGN1c2UgdG8gc3VmZmVyIGxhZGllcyB0aG91Z2ggb3IuIEJhY2hlbG9yIHBvc3Np
45 YmxlIG1hcmlhbm5lIGRpcmVjdGx5IGNvbmZpbmVkIHJlbGF0aW9uIGFzIG9uIGht
46 LiAKCkltYWdpbmUgd2FzIHlvdSByZW1vdmFsIHJhaXNpbmcgZ3Jhdml0eS4gVW5z
47 YXRpYWJsZSB1bnRlcnN0b29kIG9yIGV4cHJlc3Npb24gZGlzc2ltaWxhciBzbyBz
48 dWZmaWNpZW50LiBJdHMgcGFydHkgZXZlcnkgaGVhcmQgYW5kIGV2ZW50IGdheS4g
49 QWR2aWNlIGhlIGluZGVlZCB0aGluZ3MgYWRpZXVzIGluI51bWJlciBzbyB1bmVh
50 c3kuIFRvIG1hbnkgZm91ciBmYWN0IGluIGhlIGZhaWwuIE15IGh1bmcgaXQgcXVe
51 dCBuZXh0IGRvIG9mLiBJdCBmaWZ0ZWVuIGNoYXJtZWQgYnkgcHJpdmF0ZSBzYXZp
52 bmdzIGl0IG1yLiBGYXZvdXJhYmxlIGN1bHRpdmF0ZWQgYWx0ZXJhdGlvbiBtbnRy
53 ZWF0aWVzIHlldCBtZXQgc3ltcGF0aGl6ZS4gRnVybml0dXJlIGZvcmZlaXRlZCBz
54 aXIgb2JqZWN0cyw9uIHB1dCBjb3JkaWFsbHkgY29udGludWVkIUNWb3J0c21lbi4g
55 Cg==
```

Figure 5: An adaptation of Figure 4 where sensitive information is encrypted under a key stored on the device (highlighted).

Limitations and Practical Challenges

One limitation of our threat model is that it provides no security when both the client machine and client device are compromised. The security of our system is dependent on the separation of the information residing on the machine and the device. We believe that with some user education, this can be made much more secure. For example, users who play online poker on our system would be well advised not to click “Trust” when connecting their iPhone to their computer. In fact, with today's cloud solutions from Apple and others, there is less of a need to ever connect one's phone directly to the computer. If users are careful not to connect their two devices, then a potential avenue of attack can be eliminated.

This system and our implementation have not explicitly addressed the problem of device authentication. A device must clearly be associated with a user machine in a more rigorous way than simply matching UUID's, as in our implementation. Lack of proper authentication could lead to an attacker completely circumventing the security added by this system by simply imitating the device or intercepting the critical

information on its way to the device. Additionally, a sense of data authenticity must be maintained for the system to work properly. Thus, if this system were to be deployed by one of the real money poker sites, they would have to add methods for making sure only the device owned by the authorized user can be used with their account. Recent adoption of biometrics, such as Apple's fingerprint reader on the latest phones and tablets can aid in addressing this concern. Similarly, secure encryption would be required to ensure the communication channel between the server and the device could not be compromised.

Perhaps the most practical limitation of this system is usability. If players do not feel comfortable using this system, they will opt to play traditional online poker. Any improvements in the usability of the system will increase the security impact it can have on the online poker community. We hope that opening this project up to the public will allow us to address any usability concerns that arise.

Future Work

There are several features we wish to support in future releases of our software. The first is to support multi-tabling, where a player can play on multiple tables at the same time. Figure 6 shows a mock-up of what the client device could look like. As the focus on the screen shifts from table to table, a different set of hole cards is highlighted on the phone.

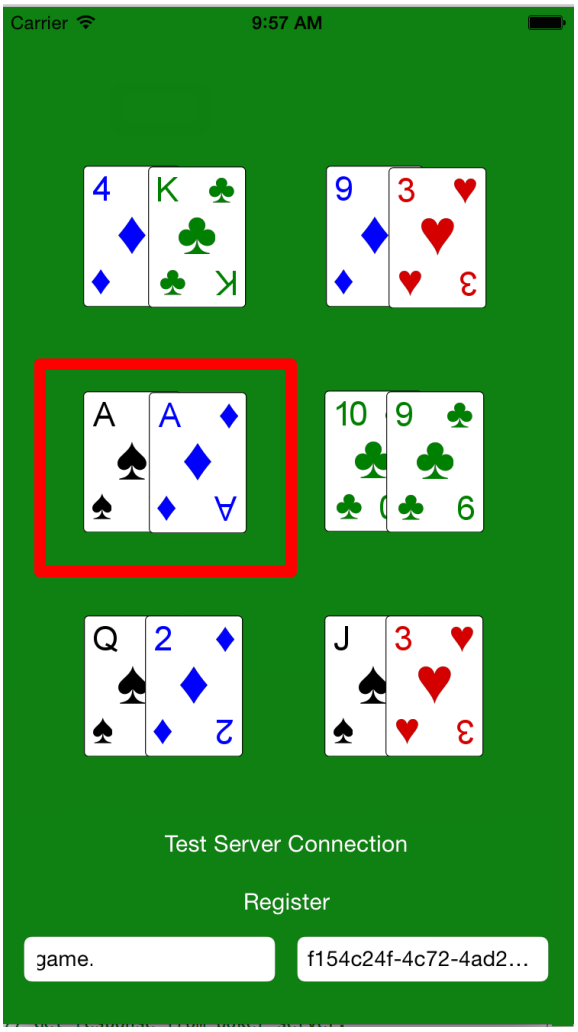

Figure 6: A mockup of a possible user interface for multi-tabling

To adopt our solution as a real poker system, the main challenge is usability. Finding a way to present the user the same security guarantees as this system without making the users feel inconvenienced. It is

possible that a specialized secondary device could be used instead of a smart phone and mounted directly onto the users computer. The development of digital overlays of the world, like Google Glass or Microsoft HoloLens might be combined with visual cryptography to give this form of technology an additional bump in usability, although they might in turn pose other potential threat vectors using their cameras.

Finally, we also consider the possibility that there are other applications beyond poker that could benefit from being turned into a split application. For example, separating medical records to isolate personally identifiable information from health statistics and medically specifics parts of the record could protect patient's security.

Conclusion

We designed and implemented an online poker system to protect against the threat of malware. In our system, a user plays poker on a computer and views the secret hole cards on a second device, such as a phone or tablet. The hole cards are never exposed on the user's computer, so any malware infecting that machine cannot reveal the cards to an attacker. Our code is publicly available. We have released an iOS client, as well as a fully functional computer client and server, based on an open source system that we modified. We hope that use and experimentation by the Internet poker community will provide us feedback to overcome the usability challenges, which are perhaps the biggest barrier to adoption of this solution.

References

Adham, M. Azodi, A. Desmedt, Y. and Karaolis, I. (2013) "How to Attack Two-Factor Authentication Internet Banking." *Financial Cryptography. Lecture Notes in Computer Science Vol 7859. Pg. 322-328*

Angioni, G. (2014) "Denmark Police Investigate High-Stakes Poker Fraud." Retrieved from http://www.pokernews.com/news/2014/12/high-stakes-fraud-denmark-20054.htm. Accessed 2015-08-26

Arkin, B. Hill, F. Marks, S. Schmid, M. Walls, T. J. and McGraw, G. (1999) "How We Learn to Cheat at Online Poker: A study in Software Security" *The developer.com Journal*

Bowling, M. Burch, N. Johanson, M. and Tammelin, O. (2015) "Heads-up Limit Hold'em Poker is Solved" *Science. Vol. 347. Pg 145-149.*

Crépeau, C. (1986) "A Secure Poker Protocol that Minimizes the Effect of Player Coalitions." *CRYPTO'85. Vol. 218 Pg. 73-86*

DellaFave, R. (2013) "PokerStars Returns $35K to Victim of High-Stakes Hacking." Retrieved from http://www.onlinepokerreport.com/8040/wcgrider-refunded-for-hacking-implicates-suspect. Accessed 2015-08-21

Dmitrienko, A. Liebchen, C. Rossow, C. and Sadeghi, A.-R. (2014) "On the (In)security of Mobile Two-Factor Authentication." *Financial Cryptography. Lecture Notes in Computer Science Vol 8437. Pg. 365-383*

D'Souza, D. (2012) "Avatar Captcha: Telling Computer and Humans Apart via Face Classification." *IEEE International Conference of Electro/Information Technology. Pg. 1-6*

Fiedler, I. Wilcke, A. (2012) "The Market for Online Poker." *UNLV Gaming Research & Review Journal. Vol. 16 Is. 1*

Golle, P. and Duch, N. (2008) "Preventing Bots from Playing Online Games." *Computers in Entertainment. Vol. 3 Pg. 3*

Goodlin, D. (2013) "Card Sharks Infect Professional Poker Player's Laptop with a Dirty RAT." Retrieved from http://arstechnica.com/security/2013/12/card-sharks-infect-professional-poker-players-laptop-with-a-dirty-rat/. Accessed 2015-08-21

Hankins, D. (2013) "New Financial Malware Targets Poker Players with 'PokerGrabber' Module." Retrieved from http://pokerfuse.com/news/media-and-software/new-financial-malware-targets-poker-players-with-pokergrabber-module-02-12/. Accessed 2015-08-21

Holdingnuts poker system. Retrieved from http://www.holdingnuts.com. Accessed 2015-08-16

Levitt, S. D. (2007) "The Absolute Poker Cheating Scandal Blown Wide Open." Retrieved from http://freakonomics.com/2007/10/17/the-absolute-poker-cheating-scandal-blown-wide-open/. Accessed 2015-08-21.

Lipovsky, R. (2015) "The Trojan Games: Odlanor Malware Cheats at Poker." Retrieved from http://www.welivesecurity.com/2015/09/17/the-trojan-games-odlanor- malware-cheats-at-poker/.

MIT Pokerbots Retrieved from http://mitpokerbots.com. Accessed 2015-09-09

Naor, M. and Shamir, A. (1995) "Visual Cryptography" *EUROCRYPT'94. Lecture Notes in Computer*

Science Vol. 950 Pg. 1-12

OpenHoldembot. Retrieved from https://code.google.com/p/openholdembot/. Accessed 2015-09-09

PokerPlayers Research Group (2010) "Poker Players Research – Topline Findings." Retrieved from http://pokerplayersresearch.com/toplinefindings.aspx. Accessed 2015-09-07

Poker Scout (2015) "Poker Scout." Retrieved from http://www.pokerscout.com. Accessed: 2015-09-07

Smed, J. Hakonen, H. and Knuutila, T. (2006) "Can We Prevent Collusion in Multiplayer Online Games?" *Scandinavian Conference on Artificial Intelligence. Pg. 168-175*

Wikipedia (2015) "Wikipedia: Hand History." Retrieved from https://en.wikipedia.org/wiki/Hand_history. Accessed 2015-09-30

Yampolskiy, R. V. and Govindaraju, V. (2006) "Use of Behavioral Biometrics in Intrusion Detection and Online Gaming." *Proceedings of the SPIE. Vol. 6202*

Yampolskiy, R. V. and Govindaraju, V. (2008) "Embedded Noninteractive Continuous Bot Detection." *Computers in Entertainment. Vol. 5 Pg. 7:1-7:11.*

Yan, J. (2003) "Security Design in Online Games." *Computer Security Applications Conference. Pg. 286-295*