THE BANKING MALWARE MESS

A *Ciphers By Ritter* Page

Terry Ritter

A = ritter B = ciphersby
A@BA.com

2010 July 14

INTRODUCTION

Hackers are stealing money through customer computers during on-line banking. Most computers are easily infected by sophisticated remote-controlled robot programs which put the attacker inside the customer machine. â€œBotsâ€ in customer computers can defeat virtually all known banking authentication schemes, including one-time 2-factor systems with an external dongle, public key certificates, out-of-band schemes and multiple authorization. Small businesses can be grievously hurt, because the bank generally will not make them whole. Serious protection is available, even free, but does require users and owners to learn to do things much differently.

CRIMINAL MALWARE

The fundamental problem is criminal malware which installs a bot infection in the customer computer for profit. â€œMalwareâ€ is short for â€œmalicious softwareâ€ installed without permission from the owner. â€œBotâ€ is short for â€œrobot,â€ a hidden program with more power than the user, remotely controlled by an attacker. An â€œinfectionâ€ is malware which has changed data on the hard drive, so the bot will run on each and every computer session. Often malware is installed by a â€œTrojanâ€ email attachment or .PDF file which appears benign, but actually contains a dangerous bot and infects the computer.

Malware

Almost all malware is designed for Microsoft Windows and will not run in other operating systems. Windows is targeted because 91 percent of browsing occurs under Windows, not because the operating system is flawed: All large, complex systems have security flaws.

Malware is effectively â€œthrown to the winds,â€ and will find itself on a Windows machine 91 percent of the time, and on a Linux machine only 1 percent of the time. If the malware was designed for Windows, it has a 91 percent chance of running. If the malware was designed for Linux, it has a 1 percent chance of running. Attackers design malware for Windows instead of Linux, not because Linux is stronger, but because only running bots make money for attackers. To avoid malware, it is important to not run Microsoft Windows online, especially when banking, and that is the first change which must be made.

Infection

An even more serious problem is infection, since an infected computer will remain infected, session after session, until the operating system is reinstalled. Infection occurs on the system hard drive, and no antivirus scanner will necessarily detect it. Without a tool to certify a computer as clean for online banking, the only way to avoid a hidden infection is to reinstall the operating system. Frequently. A better approach is to use an operating system which boots from a CD or DVD on every session, and that is the second change which must be made.

Solutions

Many recommended solutions do not work: Security software is too little, too late, and cannot clean an infected machine. Laws and law enforcement are a decade behind. Banks are limited because the problem occurs inside customer computers (although banks may be able to delay suspicious transfers). Market forces issue false and contradictory statements which keep users running to buy the latest new solution which will not work. A solution that does work is a free Linux "LiveCD."

The Banking Drama

There are four main actors in the malware online banking drama:

1. The Attacker,
2. The Bank,
3. The Customer, and
4. The Customer Computer.

Each can be blamed, but other than the attacker, blame can be less obvious than it may at first seem. Keep your eye on that last category.

BLAME THE ATTACKER

The Attacker is, of course, the ultimate source of online banking problems. The goal is profit by whatever means possible, typically criminal theft. The tool of greatest menace is the bot:

Bots

A â€œbotâ€ is a program that contacts the remote attacker, and essentially puts that attacker inside the customer computer. Bots hide, and often cannot be found by antivirus scanners.

Typically, a bot has access to everything on the machine, including everything typed or seen by the user, and can send it back to the attacker. That includes everything â€œrememberedâ€ (as in checking the browser â€œremember my passwordâ€ box), all keystrokes (including login usernames and passwords), and even screenshots (to defeat on-screen keyboards). With a bot around, nothing on the computer can be private. A bot can even get between the Web and the browser to add questions to an existing page (originally from the bank but changed by the attacker), or modify information displayed (to hide account changes).

(See: â€œZeuS on the Huntâ€ by Dmitry Tarakanov at: http://www.securelist.com/en/analysis/204792107/ZeuS_on_the_Hunt).

A bot can send messages to a bank which are impossible to distinguish from valid customer orders. Banks are limited in what they can do to stop this, first because the bot is in the customer machine, and then because bots can defeat almost every type of authentication, including one-time 2-factor external hardware dongles, public keys and out-of-band phone or cell-based schemes. Some examples:

> â€œThe theft happened despite [the] use of a one-time password, a six-digit code issued by a small electronic device every 30 or 60 seconds.â€

(From â€œModern banker malware undermines 2-factor authenticationâ€ by Dancho Danchev at: http://www.zdnet.com/blog/security/modern-banker-malware-undermines-two-factor-authentication/4402).

> â€œTrojan-based, man-in-the-browser attacks are circumventing strong two-factor authentication, enabled by one-time password (OTP) tokens. Other strong authentication methods, such as those using chip cards and biometric technology that rely on browser communications, can be similarly defeated.â€
>
> â€œOut-of-band authentication using voice telephony is also being circumvented by fraudsters using call forwarding so that the fraudster, rather than the legitimate user, is called by the service provider performing the authentication.â€

(From â€œWhere Strong Authentication Fails and What You Can Do About Itâ€ by Avivah Litan at: http://www.gartner.com/DisplayDocument?id=1245013).

> â€œ[The] Owner said her financial institution...normally notified her by e-mail each time a new wire was sent out of the companyâ€™s escrow account. But the attackers apparently disabled that feature before initiating the fraudulent wires.
>
> â€œThe thieves also defeated another anti-fraud measure: A requirement that two employees sign off on any wire requests. ...a few days before the theft, she opened an e-mail informing her that a UPS package she had been sent was lost, and urging her to open the attached invoice. Nothing happened when she opened the attached file, so she forwarded it on to her assistant who also tried to view it. The invoice was in fact a Trojan horse program that let the thieves break in and set up shop and plant a password-stealing virus on both [her] computer and the PC belonging to her assistant, the second person needed to approve transfers.â€

(From â€œe-Banking Bandits Stole $465,000 From Calif. Escrow Firmâ€ by Brian Krebs at: http://krebsonsecurity.com/2010/06/e-banking-bandits-stole-465000-from-calif-escrow-firm/).

Malware Realities

Malware is written and distributed for profit, but is limited by reality in ways which may not be readily apparent.

Malware consists of computer programs designed for particular systems. But malware is generally distributed to anyone browsing infected pages or email attachments, and then must exploit whatever system it finds. Some of the malware distribution will thus encounter the right system and will run, for potential profit. The rest of the distribution will encounter a wrong system and will fail, for a complete loss. Almost all malware is written for Windows, which means that computers which do not run Windows can avoid almost all malware.

Currently, about 91 percent of browsing occurs under Microsoft Windows (with 5 percent for Macs, and 1 percent for Linux). Yet almost all (probably more than 99.9 percent or 999 out of 1000) malwares are designed for Windows.

(See, for example: â€œMalware beyond Vista and XP,â€ by Magnus Kalkuhl and Marco Preuss at http://www.securelist.com/en/analysis?pubid=204792070).

Now, perhaps Windows does have more security problems than other systems, but not 999 times more. So why is malware not exploiting the other targets to any significant extent? We have to look beyond security problems for the answer.

Malware for Macs and Linux does exist, but if attackers distribute those, they will run only 5 percent and 1 percent of the time. Windows malware is thus 18 times more likely to be profitable than Mac malware, and 90 times more likely to be profitable than Linux. The optimum profit occurs with *all* Windows malware and *no* other malware at all. Since the Windows target is vastly more profitable, the lack of malware targeting other systems should be no surprise. An attacker would have to be nuts to target any other operating system for profit, no matter how easy it would be to defeat. Malware is about market share, not operating system weakness.

We can of course expect a few, small, malware distributions targeting other systems, for development, market testing, intelligence and propaganda. For example, if attackers can convince users that other systems also have malware problems, users may not switch away from Windows, which keeps things simple for the attackers.

It may be possible to prepare malware for multiple targets, but doing so would be twice the work and increase profit by only 5 percent at best.

Malware attacks are designed for *the single* most expected environment. Even though Windows will be found only 91 percent of the time, attackers will prepare for Windows almost 100 percent of the time, because that will give them vastly better chances of success than any other alternative.

BLAME THE BANK

It may be tempting to blame the bank, but that may not be easy:

> â€œArticle 4A of the UCC [Uniform Commercial Code] has been interpreted to absolve a bank of liability in cases where an agreed-upon security procedure is in place and a theft occurs that can be traced to a compromised PC controlled by the business customer.â€

(From: â€œCybercrooks stalk small businesses that bank onlineâ€ by Byron Acohido at http://www.usatoday.com/tech/news/computersecurity/2009-12-30-cybercrime-small-business-online-banking_N.htm).

Blaming the bank also may not be fair: When a robber takes money from a teller inside a bank, the bank itself loses money and presumably insurance pays. But nobody blames the bank when people are mugged at an ATM. So is online banking more like being a bank employee, or a customer using an ATM?

Banking Authentication

A common refrain is that if only the bank had such-and-such authentication, money would have not have been lost no matter what was in the customer computer. Unfortunately, there is no system like that.

The reality is simple:

1. On customer computers *without* a bot, banking authentication works.
2. On customer computers *with* a bot, banking authentication fails.

Currently, there is no system that can offer provably secure banking transactions in the presence of a live bot. Many dramatic failures of serious authentication give substantial reasons to doubt that security can survive the presence of a live bot. Even if technically possible, the result would be unacceptable: the bot

would still expose all customer data, including every bank login account ID and password. Even a secure SSL connection to the bank could not hide anything from the bot. The problem is not a failure of banking authentication, the problem is the bot in the customer computer.

The situation is thus:

- On the one hand we have *a dream* of banking authentication that might someday function despite a bot in a customer computer. Of course, that computer would be profoundly insecure, exposing everything viewed, everything typed, every password used, and every banking login.
- On the other hand we have *the reality* of a Linux â€œLiveCDâ€ to prevent bot infection, which is available now, but requires learning some new computer skills.

For some, a very tough choice apparently.

Disallow Microsoft Windows for Online Banking

Instead of trying to build a system to avoid an existing bot, a more responsible course might be for banks to disallow Microsoft Windows for online banking.

Unfortunately, knowing what operating system a user is using, in the context of a live bot, will be harder than one might think. Doing this well, if possible at all, would at least require development beyond what we now have.

BLAME THE CUSTOMER

Many attacks are â€œTrojans,â€ which function by getting the user to accept an apparently innocuous document with hidden dangerous content. Most Trojans are specifically invited in by user action, which means managers may be happy to â€œblame the user.â€ There is a lot of talk about a lack of computer education, which presumably would fix everything.

Computer Training

Malware infections often are the result of user error, so it is tempting to blame the computer operator. But how practical is that really?

Here is a BY NO MEANS EXHAUSTIVE list of secure computer configuration and operation tips, some of which come from hard-won personal experience. It does not get into the operational details of the operating system, browser or add-ons. It does not even start on the complex and ultimately hopeless mess of trying to configure and use Microsoft Windows securely.

- Copy each important user file to 2 other independent storage devices.
- Always re-install the operating system after finding even one malware.
- It is no longer acceptable security to â€œcleanâ€ an infection by simply deleting files, even if antivirus software says otherwise.
- Use an external hardware router and disable Universal Plug-n-Play (UPnP) and WAN remote management.
- Use wired (CAT5) internet wherever possible.
- Wireless routers MUST use AES-CCMP (usually WPA2), (also called WPA2 PSK AES) at all times, to protect both owner and user.
- Assign a unique wireless network ID (SSID) and 63-character random pre-shared key (PSK) (use https://www.grc.com/passwords).
- Always use a software firewall even with an external router firewall.
- Use a password manager such as LastPass, with a long secret keyphrase.
- Make passwords have at least 15 random characters, whenever possible.
- Set up a different long random password for every device and every site and every account.
- Do not email plaintext passwords to anyone, including yourself.
- Delete all on-line emails containing plaintext passwords.
- Put secret passwords and instructions in LastPass Secure Notes.
- Email encrypted password file to partner(s) for emergencies and backup.
- Use Puppy Linux from DVD, preferably without a hard drive.
- Use OpenDNS (208.067.222.222 and 208.067.220.220) where possible.
- Use the Firefox browser with security add-ons, including: LastPass, NoScript, Safe, SSLPasswdWarning, Certificate Patrol, URL Tooltip, WOT, BetterPrivacy and Google Docs Viewer.
- Always update to the newest browser version as soon as possible.
- Never let a browser save your passwords (use LastPass).
- Assume the whole neighborhood is on the same sub-net and can see your every packet.
- Network privacy requires SSL (**https://**), a corporate VPN, or a personal VPN service (like WiTopia).
- Avoid entering a password on a web page until SSL has been established.
- If forced to enter a password without SSL, consider the account public.
- Never do banking or purchasing without first establishing SSL.
- Never approve a new SSL certificate, especially in a "Free Wi-Fi" coffee shop or other open Wi-Fi hotspots.
- It is OK to use an existing SSL certificate that is somewhat out of date.
- It is OK to use an SSL certificate for a subdomain (e.g., www.site.com instead of site.com).
- Keep private data on an external drive or a machine with no networking.
- Use web email, preferably GMail, which has full-session SSL.
- Read email attachments on line, especially .PDF files.
- Avoid the most popular PDF reader (try Foxit).
- Never download unexpected email attachments, since a malware email can pretend to be from someone you know.
- Never click on a link in unexpected email. Either call up the supposed page directly, or copy the address, paste it to the browser, and inspect it well before going.
- Always mouse over links and examine the address before you click.
- Never click on an email link to a financial account.
- Never supply or confirm User ID, Password, or any private data via email.
- Never download browser toolbars.
- Any alert that claims your system has malware probably is itself malware.
- Any page which wants you to download and install something may be distributing malware.
- If you need an update or a player, go to the manufacturer's page and download it from there.
- Even respected companies have their pages invaded and used to distribute malware.
- The NoScript add-on is our friend even in Linux since JavaScript code will run on any browser.
- Avoid using Java online. (JavaScript is not Java.)

Do we actually expect users to study lists like this until they can do everything right? Do we actually expect to avoid all human error?

Education is fine. Who can possibly be against education? But no matter how much education or training users get, it will still be impossible to avoid every risky online action. Even when the correct response is known, a human will eventually make a mistake. When even a single mistake can allow malware to infect the system, we have a system problem, not a human problem. Education cannot prevent human error.

Not using a LiveCD when banking online may be the worst of the user errors.

BLAME THE CUSTOMER COMPUTER

The problem is a bot in the customer computer, which the bank cannot detect or kill. So the bank is not responsible. But the customer also cannot detect the bot, so they are not responsible either! The only player left standing is the customer computing hardware and software.

If operating a computer is like driving a car, we ought to be somewhat familiar with manufacturing problems and recalls. If a car, in its normal environment, suddenly became undetectably dangerous to operate, would we blame the driver, or would we blame the car and the manufacturer?

Can the Microsoft Windows product, when operating in its expected PC environment, be considered "fit for the purpose" of secure online banking?

Despite loud cries to the contrary, I believe Microsoft Windows cannot be hardened enough, by any means whatsoever, to provide secure online banking for ordinary users. This is not due to poor program quality, since all large, complex systems have errors. This is due to market share. As long as *any error whatsoever* can be found in Windows, that will be exploited to leverage the 91 percent Windows market share for criminal attacks.

Infection

Infection has been around for decades, and not nearly enough system design effort has gone into preventing it. Unlike endless patching that never seems to make a difference, absolutely stopping infection is a real possibility.

Infection occurs when malware can modify the operating system startup processes so that the bot is started on each session. If we protect the startup and operating system files so they cannot be modified, they cannot be infected. Unfortunately, operating system protection is insufficient, because malware subverts the operating system. Hardware protection is needed.

Infection can occur from a single user error, and after infection, a bot will be watching and waiting until the user finally does some banking. The basis for most malware infection is a writable boot drive (normally a hard drive, but possibly a USB flash drive). To avoid infection, users can boot a LiveCD when online.

What Could Be Done?

- Microsoft should supply a tool to certify their OS as not infected and clean for secure online banking.
- Microsoft should provide tools to support easy user re-installation of the operating system.
- Microsoft should supply a Windows â€œLiveâ€ DVD for online banking. Normal operation requires the ability to update the DVD with new or changed or deleted files without re-configuring the operating system and browser.
- Microsoft and Intel need to re-visit the PC hardware design. Boot media needs to offer hardware protection for operating system files, protection which software cannot provide when it has been penetrated by malware. BIOS flash also needs serious write protection.
- The Web needs a new protocol to authenticate the responsible owner or source of each and every piece of executable code downloaded into a browser, before it is executed. Every JavaScript and Flash and every other executable would have to authenticate before it could run. Every document that could hold executable code would have to authenticate before it would display.
- The Web needs to improve secrecy by quickly transitioning to "use SSL everywhere and all the time."

Consequences

Users will screw up, no matter how much training they have had. If a user avoids taking the Trojan bait, there is no problem. But if the user bites, the consequences vary widely, and the difference is due to the system, not the user:

- On a conventional system, the user error leads directly to a hidden bot infection on the boot hard drive which remains active, session after session, until the operating system is reinstalled. The result is a loss of secrecy and possible financial consequences.
- On a LiveCD or DVD system, the user error does not infect the boot CD, and the system still starts clean on every session. The LiveCD also runs Linux, so most likely no bot ever runs at all. There are no financial consequences.

There you have it: A LiveCD prevents bot infection while a conventional system does not. A LiveCD gives the ordinary user some welcome freedom to be human.

SOLUTIONS

Everybody has an idea about what to do. The most popular course seems to be doing nothing at all, but that will not prevent bad guys from stealing your money.

Some people argue that ordinary users simply cannot learn to boot Linux from DVD, so the whole idea is hopeless. But those who might benefit most are small businesses who have to deal with harsh reality all the time. If change is what is needed, change can happen. The result should be a less risky environment and considerably less anxiety.

The conventional approach has been to â€œhardenâ€ the PC with complex configuration, antivirus software, and extensive user training. Unfortunately, no part of that approach can be trusted to work enough to prevent malware from getting in and dropping a bot infection.

Demanding that the bank do something ignores the fact that the problem is in the customer computer and there is little the bank can do if the customer has a bot. The much vaunted banking authentication solutions simply cannot be trusted when an attacker is inside the customer computer.

New laws surely will take years, and they will get it wrong anyway. But there are some real possibilities:

The Dedicated PC

Some articles claim the American Banking Association and FBI recommend using a â€œdedicated PCâ€ for online banking.

(See, for example: â€œFeds Warn Small Businesses to Use Dedicated PC for Online Bankingâ€ by Kim Zetter at http://www.wired.com/threatlevel/2009/12/feds-warn-small-businesses/).

Searching the sites aba.com and fbi.gov yields no such recommendation, but we can still consider the proposition.

(See: â€œTracing an FBI Warningâ€ by Chris Larsen at http://www.bluecoat.com/blog/tracing-fbi-warning).

Presumably, the idea is that if a PC never browses any site other than the bank, and never does email, it cannot get infected. Of course, that only helps if the PC is not *already* infected, which means it must be new, or have an operating system reinstall. The idea also works only as long as banking pages are never infected.

(See: â€œSeveral Banks Have Their Websites Defacedâ€ by Lucian Constantin at http://news.softpedia.com/news/Several-Banks-Have-Their-Websites-

Defaced-147128.shtml).

There is nothing more secure about a dedicated PC than a normal PC, except the claim that it is used only for banking. But all it takes is a moment of inattention to infect it for the foreseeable future, and we cannot expect to detect the infection.

If a dedicated PC has ever been used for general browsing or email we cannot know, and if there is a bot in place we are unlikely to detect it. That makes the dedicated PC dangerously deceptive and a security disaster just waiting to happen.

BEST PRACTICES

Users can learn some best practices to improve online security:

- Do not use Microsoft Windows online.
- Boot a Linux â€œLiveCDâ€ for online browsing and banking, such as:
  - Puppy Linux (http://www.puppylinux.com/), which supports incremental DVD+RW updates. See PC Security for Banking on my site for download and configuration details.
  - Puppy Linux supports surprisingly convenient operation without any hard drive at all, which may be ideal for laptop security.
  - Lightweight Portable Security (http://spi.dod.mil/lipose.htm) is a US military LiveCD version, but does not seem to support DVD updates.
  - Ubuntu (http://www.ubuntu.com/) is a popular Linux, but generally expects to be installed to a hard drive.
  - Many other Linux LiveCD distributions are available.
- Use the Firefox browser with security add-ons.
- Use the LastPass password manager.
- Use SSL (**https://**) whenever possible, and make sure SSL is established before entering passwords.
- Avoid running Java.

Firefox Security Add-Ons

No other browser has anywhere near the security features available to Firefox with security add-ons, including:

- "LastPass" is a secure password manager add-on for both Windows and Linux which allows the user to make and easily use a different long random password for every device and every site and every account.
- "NoScript" converts the usual and dangerous "run any scripts from any page" to "only run scripts from specific trusted pages."
- â€œURL Tooltipsâ€ pops up a display of the underlying URL when the mouse is on a link.
- "Safe" puts a colored border around each page where SSL (**https://**) has been established, thus clearly announcing snooping protection.
- "SSLPasswdWarning" warns when submitting passwords without having established an SSL (**https://**) secure connection.
- "Perspectives" provides a way to check web page authentication and expose SSL man-in-the-middle attacks.
- â€œCertificate Patrolâ€ warns of changes in security certificates, to also expose SSL attacks.
- "Google Docs Viewer" supports viewing various file types online, especially .PDF files, thus avoiding infection.
- "Better Privacy" deletes invisible and otherwise non-deletable "super cookies."
- "WOT" helps expose phishing sites.

Until another browser has these features, the choice for secure browsing must be Firefox.

ALSO SEE

- PC Banking Security Q&A (18K)
- PC Security for Banking (46K)
- Simplified PC Security (26K)
- Basic PC Security (150K)