

# Chance and Deception: Attacks That Rely On User Actions

Ian Adams

## Abstract

There are a variety of attacks that rely heavily on user actions to succeed and don't require explicit installs or on the part of the user. We examine 3 basic categories of such attacks: Autorun attacks utilizing physical media such as flash drives and CD-Roms, phishing attacks and web based malware attacks that do "drive-by-downloads".

Through surveying users we found the following: First, many users are vulnerable to autorun based attacks, second, substituting visually similar but semantically different characters in hyperlinks provides a particularly effective way of obfuscating a malicious websites hyperlink, and third there appears to be an interesting logical disconnect whereby web links advertised physically, but still of unknown origin, are deemed more trustworthy than those from an unknown email source.

We also attempted two simple proof of concept experiments to demonstrate the feasibility of using physical advertising through flyers and "promotional" CDs as attack vectors to guide users to malicious websites. Both experiments failed, with a minimal number of hits to the flyer advertised website, and none using the CDs.

## 1 Introduction

With the explosion of personal computer and internet use, attacks on computer systems have gone from the exception to the rule. These range from taking advantage of a flaw in an authentication protocol to gain illicit access to a system [8], or utilizing buffer overflows to write and read to arbitrary memory positions [23], to distributed denial of service attacks involving literally thousands of compromised machines [19].

There are also more user oriented attack methods that require explicit user actions, such as taking advantage of "autorun" functionality on media such as flash drives to covertly insert arbitrary software [24] and malicious web sites that, when visited, take advantage of browser vulnerabilities to run arbitrary commands on the victims computer. It is these attacks that require actions on the part of the victim to succeed that are the focus of this paper.

Exploiting individual users is by no means a new idea, we need look no further than the ubiquitous 419 scams [11]. In these phishing schemes persons are often solicited to give out private data such as bank information with the promise that they will get some large sum of money in return. Though often silly sounding to those who are aware of them, these schemes do work and hundreds of thousands of dollars have been illicitly gained via this scam [11], highlighting the general gullibility of a significant subset of users. Of more concern are schemes that guide unaware users to sites that appear as valid money order or banking web sites and get users to enter their credentials which are effectively stolen in this method and can now be used for malicious purposes. The latter doesn't even require an especially gullible user base to exploit, making it all the more dangerous.

More insidious than phishing, and potentially as dangerous, are attacks that simply require a visit to a malicious web site. In these cases attackers exploit client-side vulnerabilities in web browsers that allow the user's computer to be infected with "trojans" that can exploit further system or browser level vulnerabilities. These may contain various payloads such as keyloggers for stealing personal information or attack software for using the computer in a DDoS attack [25].

On the physical side, there have been many various attacks exploiting autorun functionality of various media when run on Windows operating systems. These range from CDs and DVDs, to USB flash drives and even iPods and digital picture frames [20] to install malware.

In this paper we investigate three types of attacks that rely on user actions to succeed: Autorun attacks, phishing and web based drive-by-downloads. We also examine methods to assist in luring users to malicious web sites. In addition to the literature search, we did a survey to investigate potential user vulnerabilities to such attacks, and found that there appears to be several potential vulnerabilities. First, in relation to web based phishing and malware, many users seem to feel that physical web site advertisement through a flyer or promotional media equates to it being "more secure" than say, an email. Secondly, many users are vulnerable to autorun based attacks, and third, substitution of international characters into hyperlinks can

provide a dangerous assist to web based phishing and malware attacks.

We additionally attempted a simple anecdotal proof of concept attack using flyers advertising a legitimate forum, as well as spreading burnt CDs of an independent band—who approved our use of their music in the project—in an attempt to see if crude physical advertising would yield any hits. Neither of our experiments were particularly successful, only 2 verified hits from the forum advertisements, and none from the autorun “phone home” web link on the CD.

The rest of the paper is organized as follows: Section 2 gives an overview of the attacks we are investigating, section 3 details our survey and failed experiment and their implications, section 4 discusses potential attack mitigations, section 5 is related works, and section 6 concludes.

## 2 Attacks

Here we discuss three types of attacks that are heavily dependent upon user actions, either through explicit visitation of a contaminated or malicious web site, falling for a phishing scheme, or connection of malicious media and attack through autorun functionality. It is worth noting that these attacks are ultimately “silent” in nature. Even though they depend on users being lured to do some action, such as visiting a web site, the actual attack occurs without their knowledge.

### 2.1 Web Based Attacks

Most web based attacks rely on a user having a vulnerable browser and navigating to a malicious web site, which then allows a “drive-by-download” where malicious software such as keyloggers and adware is silently downloaded and installed. In some cases, users can be lured to actively malicious web sites through links posted on forums, mass spam emails, etc. However, the site owner/server however may not be actively malicious themselves. For example, web forums that allow users to post arbitrary material may be exploited by hostile users. An attacker can post a malicious image exploiting the WMF vulnerability—we discuss this shortly in more detail—which will cause a victims computer to silently execute arbitrary commands. Similarly 3rd party advertisements outside the control of the host might be an attack vector [25]. In other cases the web server itself may even be hacked and been turned into an unwitting host for attacks. A particularly poignant example of the latter was when the Bank of India web site was hacked [18] and had a frame tag inserted that silently linked to the hackers’

web site and attacked vulnerable browsers with a variety of trojans, keyloggers and spyware.

The Exploits to accomplish these drive-by-downloads are highly dependent on a variety of factors, ranging from operating system to firewalls and browser versions. However, the use of scripting languages has been a boon to attackers, as they may now gather information about the user such as browser versions, protocols, operating system, and so forth, allowing them to automatically tailor attacks to specific vulnerabilities [25]. To highlight some of the exploits used in the wild, we briefly describe two vulnerabilities that have been used to accomplish drive-by-downloads: The WMF and MDAC exploits.

The Windows MetaFile—WMF—is a Microsoft image file format. The WMF file vulnerability comes from its “Escape” records which extend its functionality. Specifically the SETABORTPROC—set abort process—record allows arbitrary code to be executed when the rendering of an image failed [4]. This was originally intended to deal with issues like print spooling, but was exploited such that specially crafted images would be designed to fail rendering, and then call the commands specified in the SETABORTPROC record. Once the malicious image is viewed, commands to download other software such as a trojan horse can be executed that may in turn exploit other vulnerabilities.

The Microsoft Data Access Components or MDAC vulnerability, is found in ADODB.connection objects execute method . ADODB is a library for PHP and Python used as a database abstraction layer. Utilizing the execute method with malformed parameters can corrupt memory and in turn potentially allow execution of arbitrary code [5]. Like the WMF vulnerability, if the malformed query is crafted in a specific way, this can then be used to download trojans to exploit other vulnerabilities and install malware.

### 2.2 Phishing

Phishing is a technique whereby users are tricked into revealing sensitive information such as bank account numbers. These range from the relatively naive approaches preying on the ignorance of a person, such as the infamous Nigerian money scams [11] to more sophisticated schemes like targeting specific user groups with links to legitimate appearing banking web sites [22] asking them to “confirm” their credentials, when in reality they are providing the attacker with access to their account.

<a href="http://www.google.com">http://www.google.com</a>	<a href="http://www.google.com">http://www.google.com</a>
<a href="http://www.cnn.com">http://www.cnn.com</a>	<a href="http://www.cnn.com">http://www.cnn.com</a>
<a href="http://www.nsa.gov">http://www.nsa.gov</a>	<a href="http://www.nsa.gov">http://www.nsa.gov</a>
<a href="http://www.google.com">http://www.google.com</a>	<a href="http://www.google.com">http://www.google.com</a>
<a href="http://www.CNN.com">http://www.CNN.com</a>	<a href="http://www.CNN.com">http://www.CNN.com</a>

Figure 1: Here are several link pairs with the characters all latin characters on the left, and miscellaneous international characters such as cyrillic and greek substituted in for the highlighted characters on the right.

### 2.3 Homograph Attacks and Link Obfuscation

The idea of a homograph attack is to trick users into believing a malicious source is a legitimate one. In April of 2000, a web site masquerading visually as Bloomberg news service suggested PairGain Technologies would be purchased for twice its market value, causing a subsequent spike and fall in stock prices [13]. This attack succeeded due to the fact the links to the site were posted as raw IP addresses on message boards, rather than the usual string URL, obfuscating its true location. When users visited the site it visually appeared to be the Bloomberg web site [13], and evidently many did not check the address bar of their respective browsers to ensure they really were at the Bloomberg news website. Deceptive tactics as such can cause serious monetary and intangible “trust” damage as the Bloomberg attack illustrated, as well as assisting in phishing and malware attacks as well by making users more likely to believe they are dealing with a legitimate known source.

The inclusion of international character sets into browsers and software can be a particularly insidious assist to such deceptive methods. Many times web links and URLs can be deemed suspicious simply by visual examination, for example, seeing the difference between [www.google.com](http://www.google.com) and [www.g00gle.com](http://www.g00gle.com). International characters make this comparison significantly trickier as some software can render fontsets that make links visually appear nearly identical, but in fact point to different addresses. Figure 1 shows some hyperlinks with the differing characters between each pair highlighted. Further complicating this is a lack of standards in how software and browsers render these characters, if they render them at all.

As an example, Apple’s iWork09 word processor has

<http://www.google.com>



Figure 2: Some programs such as Adobe’s Acrobat Reader provide preview functionality asking users if they really intend to visit the site, as seen here despite visually appearing as Google in the hyperlink, it really links to someplace very different due to the use of unicode characters.

the ability to render many various unicode characters and provides hyperlink insertion. If we generated a PDF, and view it in Adobe Acrobat, a unicode hyperlink that is clicked on prompts the user if they wish to go to a site that is clearly not as it initially appears whereby the non-latin characters are translated to garbled ASCII—see figure 2. The Apple Preview program however, has no such functionality, and a user could easily think they are clicking on a legitimate link, to say Google, and in reality are directed someplace entirely different.

The TinyURL [3] redirection service provides another method of obfuscating a URL. The service takes arbitrary URLs and converts them into usually much shorter ones. For example “<http://www.yourehosed.org>” is turned into “<http://tinyurl.com/nohw4g>”.

TinyURL was recently used in a phishing scheme whereby users of the popular “Twitter” blogging service were linked to a site superficially appearing to be Twitter to log in to [16]. Though they offer a preview function which allows users to see the URL they will be redirected to, it is neither required, nor as we found in our survey, universally used. Furthermore it is still vulnerable to visual similarity attacks as it was in the Twitter attack—“Twitter” was replaced with “Tvvitter” in the URL. It is worth noting however that international characters appear to be translated to garbled ASCII, so for the moment it does not appear vulnerable to the above international character substitution dangers.

### 2.4 Malicious Media-Autorun Dangers

While direct user installation of malicious software is certainly a threat, we are focusing on “silent” attacks that occur automatically. Specifically the possibilities of using autorun functionality to automatically run arbitrary code and commands on a system upon connection or insertion

```
[autorun]
open=browsercall.exe readme.html
```

Figure 3: The contents of a very small autorun.inf file, that when parsed runs browsercall.exe program with the argument readme.html. Both browsercall and readme are in the root directory of the CD

of the malicious media.

The most vulnerable systems to this type of attack are those running versions of the ubiquitous windows operation system which support autorun. On older versions of Windows XP, even when autorun is explicitly disabled it may still be vulnerable to autorun based attacks [2]. Similarly, Apples OS 9 used to support similar autorun functionality, but dropped in the OS X release. Even though Apple’s OS X no longer supports explicit autorun, it does allow media to automatically open to specified folders, and a clever attacker could entice users to execute malicious programs by asking them to install applications, or naming their attack executables as “ReadMe” files.

For Windows based systems, storage media such as CDs and DVDs can have an ASCII text file named “autorun.inf”. This file contains plain text commands that can create special options when a context menu is opened, as well as executing arbitrary code automatically when the autorun.inf file is read. For example the command

```
open=runThis.exe
```

would cause the operating system to search the root directory of the inserted media for the runThis.exe file and then attempt to execute it.

As an illustration of the simplicity of exploiting autorun, we created a CD with an autorun file that opened an HTML file with the default web browser and the HTML file redirected to a web site of our choosing. It took less than 15 minutes to understand the format of the file, find a 3rd party executable used to locate the default browser, burn the CD and test its functionality. See figures 3 and 4 for the contents of two autorun.inf files. This highlights the ease with which an attacker can create a method to silently installs malicious software and run arbitrary commands.

USB drives and devices have similarly been used to spread malicious software. Worms such as Conficker [24], the Obama Worm [21], and many others [14, 15, 27] have been spread using USB in such a fashion. Recently government agencies and the US military have begun banning the use of flash drives due to this threat [6].

Most USB drives—at least in our work with them—don’t execute and parse the autorun.inf file immediately

```
[autorun]
shellexecute="shutdown.exe -r -t 1 -f"
shell\Open\Command="shutdown.exe -r -t 1 -f"
```

Figure 4: Another autorun file, that on windows systems will force an automatic restart

upon connection, but rather upon a user “opening” the device, e.g. double clicking the removable device icon. U3 Smart drives however, in addition to appearing as a generic flash drive to the system, emulate a CD-Rom and thus can execute code completely automatically upon connection [15] as done with CDs and DVDs . We also found and successfully tested free 3rd party tools that allow one to easily modify the emulated CD image, opening the path for simplistic autorun like we tested, to sophisticated tools to automatically search and download encryption keys, records, passwords and a variety of system information [1, 15].

It is worth mentioning that on top of the above, such attacks are not limited just to optical media and USB thumb-drives. They have additionally been accomplished using a variety of USB connectable devices such as iPods and even electronic picture frames [7, 20].

### 3 Survey

We designed a survey that was aimed at examining the potential vulnerabilities to the attacks and misdirection methods we discussed in section 2. Our survey consisted of 19 multiple-choice questions on a variety of topics relating to use of physical media (CDs, flash drives, etc.) and web browsing habits regarding what constitutes safe and unsafe “link” practices. Additionally, on some questions users were able to type in an elaboration on their choice of answer if they wished to do so.

We surveyed 57 persons, of whom 45 completed the survey in its entirety. Our survey was web based, and as such consisted of persons with at least a minimal level of computer literacy. 2 persons identified themselves as having “Technical backgrounds”, 11 as “medical business”, and 44 as “general”. The general background consisted predominantly of recent college graduates as well as those whom are currently enrolled in some form of college.

It is important to note that our survey methods were not random and were based on sources of convenience—peers, friends at various companies, etc.—and as such cannot be considered scientifically or statistically valid. However, it is our belief that they still provide some illuminating, but admittedly anecdotal results.

Below we discuss our more interesting findings and some of their potential implications. Each subsection starts with one or more survey questions and responses we received, followed by a discussion of the results and what we saw as their implications in light of the attacks we examined.

### 3.1 Physical Advertisements-Web Sites

*Assuming its advertised content is of interest to you, would you click on a link from an unfamiliar email source? Provided it's not obviously suspicious e.g. the link is to <http://www.g00gle.com> instead of <http://www.google.com>*

Total Responses	51	100%
Yes	12	23.5%
No	39	76.5%

*If someone left a flyer where you would clearly see it (doorstep, bus stop, etc) and it was on a topic of interest or importance to you (political event, ballot measure, concert, church event, insert topic of choice here) would you visit their website?*

Total Responses	50	100%
Yes	35	70.0%
No	15	30.0%

Unsurprisingly many users are wary of links sent to them via email. Even if the link does not appear outwardly suspicious to the user and sounds “interesting”. However, if the web address was advertised via a physical method, many said they would visit the link. One user commented they would not expect foul play from a physical source, another stated that they felt that “...for the most part these things are legitimate.”

This suggests an interesting hole in the mindset of users and could provide an alternate attack vector from the typical email spam and phishing. It may not be of great use for random infections, but may be ideal for targeted attacks against a particular company, ideological groups, etc. if the advertised material can be well-tailored to pique their interest.

Though it may not be feasible in countries with stricter radio regulations and traffic monitoring, it would be interesting to see if this mindset also applies to radio advertisements as well, perhaps as another method to lure users to malicious websites.

### 3.2 Autorun Vulnerabilities

*Does your computer support autorun functionality? Examples of autorun are when certain media (CD's, thumb-drives, etc) are inserted and automatically a window is opened, a program is run, a 'splash screen' appears, etc.*

Total Responses	57	100%
Yes	44	77.2%
No	9	15.8%
I don't know	4	7.0%

*Do you have autorun explicitly disabled on your machine?*

Total Responses	56	100%
Yes	12	21.4%
No	36	64.2%
I don't know	8	14.2%

*What operating system is running on the machine you are using?*

Total Responses	57	100%
OS X	10	17.5%
OS 9	1	1.7%
Macintosh but unknown	1	1.7%
Windows XP/Vista/95/98	39	68.4%
Other-Unix/Linux/etc.	5	8.8%
I don't know	1	1.7%

Many of the surveyed users run some version of Windows, which is known to have the easily exploitable autorun functionality. Though some users explicitly disabled the autorun feature the disabling may be ineffective depending on the Windows version [2]. A total of 27 users were found to both run a version of Windows, and either did not know if autorun was disabled or explicitly stated it wasn't.

This is dangerous in conjunction with malicious media, such as the USB drive spread worms we mentioned earlier, and may have potential for use in targeted attacks by spreading contaminated 'promotional' media, which we discuss the feasibility of later.

### 3.3 Promotional Media, and Curiosity

*Do you or would you accept AND use free promotional music or software CDs or DVDs provided they were of a relevant and interesting nature to you?*

<b>Total Responses</b>	50	100%
Yes	35	70.0%
No	15	30.0%

Have you ever connected or inserted media devices(USB thumb drive, CD, DVD) of unknown origin into your computer? For example, finding a thumb drive lying on the street, free music CDs, etc.

<b>Total Responses</b>	53	100%
Yes	13	24.5%
No, but I would	13	24.5%
No, never	27	50.9%

The above highlights the possibility of promotional media and advertising as a feasible attack vector, especially in conjunction with many users' autorun vulnerabilities. Users stated that free promotional software or music would be used, some added provisos though of checking if the source seemed legitimate. To aid in appearing legitimate, with relatively low effort and expense one can make nearly professional looking CDs with off the shelf label kits.

Sometimes it may not even be necessary to advertise something as promotional. Some state that they would insert a thumb drive or CD of unknown origin that they simply found on the street out of curiosity. This would likely be an inefficient method of attack, but certainly feasible. One user commented that they would connect a USB drive to their computer out of curiosity, and another stated that they would keep a drive if they could not find the owner. This exploitation of simple curiosity was recently demonstrated when a popular band intentionally left USB flash drives in restrooms with their songs[26].

### 3.4 Unicode Dangers

In our survey, we showed users 10 link pairs consisting of 'major' web sites— CNN, MSNBC, Google, NSA, and The Onion—of which six pairs had international characters substituted in, functionally making a different link. We asked the users to tell whether each link pair was identical or not. Figure 5 shows the link pairs that users were given, and Figure 6 is a break down of the individual link pair identifications. Though not surprising given the similarities between many characters, it confirms our suspicions that unicode URLs can be very deceptive and would be of great assistance to both luring users to sites with web-based malware as well as phishing schemes.

- |  |   |
|--|---|
| 1) <a href="http://www.google.com">http://www.google.com</a>     | <a href="http://www.google.com">http://www.google.com</a>     |
| 2) <a href="http://www.cnn.com">http://www.cnn.com</a>           | <a href="http://www.cnn.com">http://www.cnn.com</a>           |
| 3) <a href="http://www.nsa.gov">http://www.nsa.gov</a>           | <a href="http://www.nsa.gov">http://www.nsa.gov</a>           |
| 4) <a href="http://www.google.com">http://www.google.com</a>     | <a href="http://www.google.com">http://www.google.com</a>     |
| 5) <a href="http://www.CNN.com">http://www.CNN.com</a>           | <a href="http://www.CNN.com">http://www.CNN.com</a>           |
| 6) <a href="http://www.CNN.com">http://www.CNN.com</a>           | <a href="http://www.CNN.com">http://www.CNN.com</a>           |
| 7) <a href="http://www.msnbc.com">http://www.msnbc.com</a>       | <a href="http://www.msnbc.com">http://www.msnbc.com</a>       |
| 8) <a href="http://www.msnbc.com">http://www.msnbc.com</a>       | <a href="http://www.msnbc.com">http://www.msnbc.com</a>       |
| 9) <a href="http://www.theonion.com">http://www.theonion.com</a> | <a href="http://www.theonion.com">http://www.theonion.com</a> |
| 10) <a href="http://www.nsa.gov">http://www.nsa.gov</a>          | <a href="http://www.nsa.gov">http://www.nsa.gov</a>           |

Figure 5: These were the link pairs shown to users in the survey. The lefthand links are only using latin characters. The righthand links are either the same, or if they contain colored characters, have had the corresponding colored characters replaced with other international characters

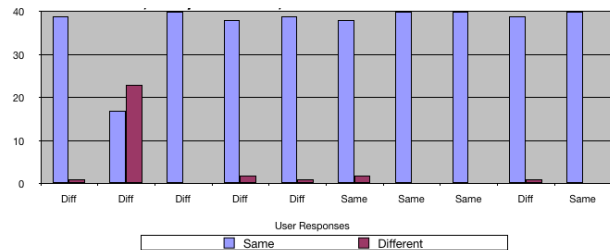


Figure 6: Breakdown of the proportion of users answers to the link pair examinations. Along the x-axis "same" means the link pair was the same "diff" means it was different. The left-most position corresponds to the first link pair above, likewise with the second and so forth.

#### 3.4.1 URL examination

If a website visually appears legitimate, how often (if at all) do you examine the URL address? As an example, if you see the Google search screen, do you check that you are indeed at the Google web address?

<b>Total Responses</b>	56	100%
Every time	9	16.1%
Intermittently	30	53.6%
Not at all	17	30.4%

Though in the case of drive-by-downloads, one quick visit is all that is necessary for the attack, with phishing

the user may need to spend extended amounts of time at the website. Users who frequently check the URL may notice they aren't where they thought they were, but for those who rely solely on the visual look of a site, they may not notice in time, if at all.

### 3.5 TinyURL

*When you are sent a TinyURL do you use the preview function?*

Total Responses	32	100%
Always	5	15.6%
Only if I do not recognize the source	16	50.0%
Never	11	34.4%

In retrospect, we realized this question was poorly phrased, and should have included a "I never click on tinyURL links" option. Additionally, users were told to skip this question if they were unfamiliar with the tinyURL service, leading to the significantly lower number of responses. Regardless, the above still lends some credibility to our speculation that the TinyURL redirection service could be used to obfuscate malicious links. As mentioned earlier it seems to garble international characters, but as we can see, not everyone checks the link source with the preview function. Additionally, email addresses being spoofed or hijacked is not uncommon so a superficially familiar email source may not actually be so.

### 3.6 Proof of Concept Tests

We attempted two separate proof-of-concept tests using physical advertising methods and exploiting user curiosity. The first using flyers to advertise a forum we set up to discuss university issues. The second spreading burnt CDs containing music from an independent band with autorun opening an HTML document pointing to a redirect site that tracked incoming traffic. The CDs and were left at various frequently crowded bus stops and the flyers were left in high traffic locations through out the university campus.

To confirm the traffic was from our sources and not random internet traffic we used a server-side PHP script which checked for specific key values stored in variables in the provided URLs both from the flyer and the autorun-redirect page on the music CD.

Both tests essentially failed. We received no hits from the CD tracker, and only a few confirmed hits from the flyer. There are several possible reasons for the relative failure of our experiment, ranging from poor choice of flyer

locations, to failed advertising that did not catch the eye of passerby, possibly even users being wary. In short, we don't know given the extremely informal methods we employed.

#### 3.6.1 Ethical Considerations

There are potential ethical issues regarding our experiments, but we believe our methods were morally sound. The advertised forum not only existed, but contained the topics advertised and was fully functional. For the music CDs we contacted the band, made them fully aware of the test, and had their permission to use their music. In the case of tracking the band and flyer traffic we actually tracked significantly less than most commercial web sites. We merely maintained a count and nothing else. No IP addresses, browser versions, or any stats whatsoever beyond a traffic count were used.

### 3.7 Overall Implications

Some of the methods and attacks we have discussed such as the unicode obfuscation have great potential for generalized attacks, like constructing botnets and phishing. Exploiting the disconnect between something being physical—like a flyer advertisement, or malicious music or software CDs—is more likely to be useful for tailored attacks targeting a specific group. For example, a company may have standard USB flash drives that are used; an attacker attempting to infiltrate could buy a device of the same model loaded with their attack software, and if the company has vulnerable machines, e.g. with autorun, they can spread a few of the devices around, banking on the chance that a user may find one and plug it in to find who it belongs to.

## 4 Mitigation

For mitigating the danger from malware, the first line of defense is keeping one's operating system, anti-virus software and web browser updated. While there are inevitably going to be "zero-day exploits" where there will be a period of vulnerability, keeping ones software up to date will do much to reduce the possible threats.

Disabling autorun will also help a great deal. While there are bypasses in some situations as we mentioned earlier, it's a simple and effective defense.

Phishing and URL obfuscation are harder to deal with. There has been work in automatically detecting "Unicode Attacks"[12] comparing legitimate URLs to those with

substituted characters. Their methodology compares visual and semantic similarities along with color highlighting to denote characters from various alphabets. There has also been the suggestion of simply restricting a URL to a single alphabet type, but this may not be feasible in practice as there are sites that mix character sets in their URLs [13]. The text coloring method seems promising to us, but as of this date, we have not seen it in common use. Another Anti-Phishing tool is a web browser side bar—called a “Web Wallet”—where sensitive information is entered, and the destination is compared to the source web site. If the destination does not appear to match the web site source or the site is not secured, the user is alerted to potential phishing [28]. Like the text coloring however, the web wallet, or anything similar to it, does not appear to be in widespread use as of this writing.

## 5 Related Work

Provos *et al.* [25] provide a summary of both the depth and nature of web-based malware problem. They highlight both the difficulty in finding web based malware, as well as its pervasive nature. During the course of their investigation they found over 200 thousand unique malware binaries and over 400 thousand URLs that were identified as malicious. Additionally they found that the malicious software had a wide range of capabilities ranging from simple ad pop-ups to exfiltration of sensitive information and full control of an infected machine. Though they only briefly examine methods of “tricking” the user, they still acknowledge it can be a key part of the infection process, but certainly not the only one, as third-party ads, and even forums and blogs, can be used as attack vectors.

A recent innovation in attack methods comes from Chan *et al.* [9] with BootJacker. BootJacker works by interposing itself during a forced restart to install arbitrary malware and then restoring system state quickly, allowing for access to live encrypted communication sessions. BootJacker requires both physical access to the machine and a forced restart to work. It is more tailored to a user who can steal or gain personal access to a machine, but may have potential for the indirect approaches we are examining. As mentioned earlier we found and demonstrated you that can run arbitrary shell commands from and autorun.inf file to force a restart, and if they had USB or CDs as part of the boot path before disk it may be possible to install payloads without even personally being near the machine.

Dhamija *et al.* [10] did a study of why phishing works, and we found that some of our findings roughly corresponded with theirs. In particular they found that with

legitimate-appearing websites, users were more likely to be fooled, and additionally missed the differing web address being subtly different. Even in cases where the address was significantly different, e.g. [www.paypal.com](http://www.paypal.com) versus [www.paypal-signin03.com](http://www.paypal-signin03.com), users could be fooled, even after being ‘primed’ that the test was asking them to look for spoof websites. This was hardly the only spoofing method however, and they found that user ignorance about security features as well as the attention span given to them also contributed to the success of phishing. In all phishing is a surprisingly effective activity, studies have estimated that phishing costs nearly 2.4 billion dollars per year, with nearly 5% of american internet users being successfully targeted [17].

## 6 Future Work & Conclusions

Our work was essentially anecdotal. In the future we would like to have a proper survey as well as more controlled and professional experiments. Garnering assistance from psychology and or marketing professionals to be more enticing in our methods of advertising to pique user curiosity, and would be likely representative of a well thought out targeted attack. Furthermore, provided autorun functionality continues to be used in the wild, we would like to do a more indepth investigation into the possible attacks using this method, in particular, mixing BootJacker with an autorun bootable device.

With our methods, we investigated and discussed 3 basic methods of attack that are dependent upon user actions: Phishing, drive-by-downloads, and malicious media. We also found there are a variety of methods to obfuscate URLs and weblinks. We found through user survey that phishing and luring users to malicious web sites can be assisted through hyperlink misdirection techniques. We also found that many users are potentially vulnerable to malicious media that utilizes autorun to silently execute arbitrary code. We feel that web link obfuscation is particularly dangerous and is an issue that needs more attention to deal with. Lastly, physical media provides both a general attack vector, as evidenced with USB spread worms, as well as great potential for targeted attacks against specific groups.

## Acknowledgments

Thanks to my classmates and fellow SSRC lab members for their valuable feedback and support.

## References

- [1] Gonzor228. <http://gonzor228.com/download/>.
- [2] How to disable the autorun functionality in windows. <http://support.microsoft.com/kb/967715/>.
- [3] Tiny url. <http://tinyurl.com/>.
- [4] Secunia security advisory. <http://secunia.com/advisories/18255/>, February 2006.
- [5] Secunia security advisory. <http://secunia.com/advisories/22452/>, February 2007.
- [6] ALERTBOOT.COM. Military usb memory stick ban: Lack of disk encryption is not the only issue. <http://www.alertboot.com/blog/blogs/>, November 2008.
- [7] BRENNER, B. Infected ipods a threat to corporate networks. [http://searchsecurity.techtarget.com/news/article/0,289142,sid14\\_gci1225559,00.html](http://searchsecurity.techtarget.com/news/article/0,289142,sid14_gci1225559,00.html), 2006.
- [8] BURROWS, M., ABADI, M., AND NEEDHAM, R. A logic of authentication. *ACM Transactions on Computer Systems* 8 (1990), 51–76.
- [9] CHAN, E., CARLYLE, J., DAVID, F. M., FARIVAR, R., AND CAMPBELL, R. H. Bootjacker: Compromising computers using forced restarts. In *ACM Conference on Computer and Communications Security* (2008).
- [10] DHAMIJA, R., TYGAR, J., AND HEARST, M. Why phishing works. In *Conference on Human Factors in Computing Systems* (April 2006).
- [11] EDELSON, E. The 419 scam: Information warfare on the spam front and a proposal for local filtering. *Computers & Security* 22, 5 (2003), 392–401.
- [12] FU, A. Y., DENG, X., WENYIN, L., AND LITTLE, G. The methodology and an application to fight against unicode attacks. In *Second Symposium on Usable Privacy and Security* (2006).
- [13] GABRILOVICH, E., AND GONTMAKHER, A. The homograph attack. *Communications of the ACM* 45, 2 (February 2002), 128.
- [14] GRAY, P. Telstra distributes malware-infected usb drives at auscert. <http://searchsecurity.techtarget.com.au/articles/24758>, May 2003.
- [15] HIGGINS, K. J. Smart usbs gone bad. <http://www.darkreading.com/security/management/showArticle.jhtml?articleID=208804413>, Mar 2007.
- [16] HUMPHRIES, M. Twitter+tinyurl = tvviter scam. <http://www.geek.com/articles/news/twitter-tinyurl-tvviter-scam-20090522/>, May 22 2009.
- [17] JAKOBSSON, M., AND RATKIEWICZ, J. Designing ethical phishing experiments: A study of the (rot13)ronl query features. In *International World Wide Web Committee* (May 2006).
- [18] KEIZER, G. Bank of india site hacked, serves up 22 exploits. <http://www.computerworld.com/action/article.do?command=viewArticleBasic&articleId=9033999>, August 31 2007.
- [19] KIRK, J. Estonia recovers from massive ddos attack. <http://www.computerworld.com/action/article.do?command=viewArticleBasic&articleId=9019725>, May 2007.
- [20] LEMOS, R. Malware hitches a ride on digital devices. *Security Focus* (January 2009).
- [21] MILLS, E. 'obama worm' probably a student prank, experts say. <http://www.zdnetasia.com/news/internet/0,39044908,62050466,00.htm>, January 2007.
- [22] MILLS, E. New phishing attempt targets bank customers. [http://news.cnet.com/8301-1009\\_3-10057180-83.html](http://news.cnet.com/8301-1009_3-10057180-83.html), October 2008.
- [23] PINCUS, J., AND BAKER, B. Beyond stack smashing recent advances in exploiting buffer overruns. *IEEE Security & Privacy* (2004), 20–27.
- [24] PORRAS, P., SAIDI, H., AND YEGNESWARAN, V. An analysis of confickers logic and rendezvous points. Tech. rep., SRI International, 2009.
- [25] PROVOS, N., MCNAMEE, D., MAVROMMATIS, P., WANG, K., AND MODADUGU, N. The ghost in the browser: Analysis of web-based malware. In *HotBots'07* (2007).
- [26] SPEROUNES, S. Paranoia lampooned in discs marketing. *Times Colonist* (April 15 2007).

- [27] TUNG, L. Hp ships usb sticks with malware. *Cnet News* (April 9 2008).
- [28] WU, M., MILLER, R. C., AND LITTLE, G. Web wallet: Preventing phishing attacks by revealing user intentions. In *Symposium on Usable Privacy and Security* (July 2006).