Drive-By Pharming

Sid Stamm[1], Zulfikar Ramzan[2], and Markus Jakobsson[1]

[1] Indiana University, Bloomington IN, USA
[2] Symantec Corporation, Mountain View CA, USA

Abstract. This paper describes an attack concept termed Drive-by Pharming where an attacker sets up a web page that, when simply viewed by the victim (on a JavaScript-enabled browser), attempts to change the DNS server settings on the victim's home broadband router. As a result, future DNS queries are resolved by a DNS server of the attacker's choice. The attacker can direct the victim's Internet traffic and point the victim to the attacker's own web sites regardless of what domain the victim thinks he is actually going to, potentially leading to the compromise of the victim's credentials. The same attack methodology can be used to make other changes to the router, like replacing its firmware. Routers could then host malicious web pages or engage in click fraud. Since the attack is mounted through viewing a web page, it does not require the attacker to have any physical proximity to the victim nor does it require the explicit download of traditional malicious software. The attack works under the reasonable assumption that the victim has not changed the default management password on their broadband router.

1 Introduction

Home Networks & Drive-by Pharming. Home broadband routers are becoming more popular as people wish to share broadband Internet access with, or provide wireless access to, all computers in their homes. These routers typically run a web server, and configuration of the router is done through a web-management interface. People assume that this internal network is safe from outside attackers since home routers are usually configured by default to reject all incoming connection requests.

However, we show that it's possible to construct a web page that, when simply viewed, can manipulate its visitors' home routers, changing its settings. The attacker can then selectively siphon off the victim's internet traffic to an attacker-controlled server, leading to the theft of sensitive credentials and identity information. The attack, which we call *Drive-by Pharming* can also enable spread of malware, target phishing attacks, or starve the visitor from critical security updates. The attacks do not require the attacker to have any physical proximity to the victim's machine. Also, the attack methodology applies equally to wired and wireless routers. The attack only assumes that the victim is running a JavaScript-enabled browser and that the default management password on the router has not been changed. Moreover, many standard protection mechanisms

for wireless networks (e.g., encrypting the traffic through WPA), do nothing to stop these attacks.

In more detail, the paper describes a web-based automated method to detect routers on a victim's internal network and then change the router settings using JavaScript-generated host scans and a cross-site request forgery (using HTTP requests). We then describe attacks stemming from this internal network scanning, delving into the effects of changing the DNS values on home routers and how this can be used by attackers to perform more successful and difficult to detect phishing scams. We also present ways for this attack to become self-sustaining and spread in a viral fashion between routers using human users as a trigger for its spread.

Combining the results of an informal survey that found that 50% of home users use a broadband router with default or no password [11] and a formal study that shows 95% of home users allow JavaScript in their browsers [13], we estimate that 47.5% of all home users (hundreds of millions of users [8]), are potentially susceptible to the attacks we describe.

1.1 Phishing and Pharming.

Phishing is a prevalent scam in which attackers masquerade as an authority in an attempt to obtain identity credentials from victims. It is a significant industry, but Gartner estimates that approximately 3% of a phishing attack's targets will fall victim [15]. Violino explains how scammers can dramatically increase their yield by spoofing DNS records for a victim domain [12]. When DNS records are spoofed, instead of going to the "correct" web site corresponding to a web site such as `bank.com`), victims will navigate to a fraudulent site that appears to be legitimate. The browser will even display `bank.com` in its address bar. The scammer can stealthily usurp all web traffic directed at the victim domain. These DNS spoof attacks, or Pharming attacks, are harder to detect than ordinary fraudulent web sites since the address bar on the browser displays the domain of the spoofed site. There is no need to lure victims to a phishing site when they will find a pharmed site on their own—removing the possibility that someone will catch an attack based on the lure. Pharming can easily be accomplished when an attacker can change settings on home routers. Traditional techniques for pharming include directly compromising a DNS server, poisoning its cache, or even compromising the HOSTS file on an end-user's PC. This paper demonstrates a different way to engage in pharming—namely, by compromising the DNS settings on the end-user's home broadband router. The paper further shows a relatively easy-to-carry-out mechanism for achieving this aim through the use of a specially crafted web page.

1.2 Attacking a Home Router.

Most end users create a small internal network at home by purchasing a consumer router (such as a Linksys, Belkin, Netgear, D-Link, to name a few). For the

entirety of this paper, we will discuss attacks on consumer or home routers—those purchased for use in homes and small businesses, not commercial-grade routers. Any reference in this paper to a "router" indicates the consumer routers and not commercial grade ones.

Assumptions can be made about the internal IP address of a deployed consumer router; alternatively, one can guess the visitor's internal IP address range (or detect it using a simple Java Applet [6]) and initiate a JavaScript-based host scan via the victim's browser to detect consumer routers with HTTP-based administration. Once a router is identified, the malicious (external) web site can use the victim's browser as a conduit to take control over the router on the victim's (internal) network. This leads to many attack scenarios, like modifying the router's DNS settings or changing its firmware, which we propose and describe in detail. More details and proposed countermeasures are discussed in our related Technical Report [5].

Overview. Section 2 describes related work. In Section 3, we describe how an internal network is identified, and what types of attacks emerge from control over a home router. We continue by discussing how attempts to attack a router from inside an internal network can be accomplished quickly and quietly. Section 4 describes the technology and techniques used to discover and attack an internal home router from an external web site, describing some of the JavaScript code utilized. In Section 5 we discuss the different types of attacks that can be mounted by controlling a home router and how these may spread in a socio-viral fashion.

2 Related Work

Internal Net Discovery. Kindermann has written a Java Applet that discovers a host's internal (i.e., NAT'ed) IP address [6]. Simply because this detection is accomplished via a Java Applet, and 94% of people on the Internet leave Java enabled [13], his method of internal IP discovery can be considered quite reliable. He also describes ways to prevent sites from using his technique to determine a host's internal IP: disable ActiveX, Java, and all other plug-ins.

Grossman [3] shows that once an internal IP is obtained, host scanning using JavaScript is easy by attempting to load images or scripts from a host on various ports. Likewise, scanning for web-serving hosts on a network is simple, and a list of web-serving IPs can be quickly identified. SPI Labs [17] show that existing image names and dimensions combined with the default password used to access the router can provide a "fingerprint" giving away the type of router. We use this technique combined with knowledge of default router passwords and default router DHCP schemes to quickly identify routers on internal networks — then reconfigure them. Tsow et al .[11] show how router firmware can be changed by accessing its configuration web page.

Building on these works, we illustrate stealthy attacks on internal networks that manipulate a router's settings or completely takes control by replacing

the router's firmware. The attacks are both difficult for service providers and victimized consumers to detect, and also exhibit high success rates.

Bad Security Assumptions. According to Microsoft, 3.5 million Windows computers are infected with back-door trojans [1, 16]; many infections are caused by web pages that automatically trigger a download of an executable or ActiveX control which the targeted user must authorize by clicking "run" when prompted. People blindly authorize these drive-by, possibly malicious executables because of repeated prompts (authorizing is the only way to stop them) or because they are unaware what the prompt means.

Drive-by virus infections are amplified when internal networks are not secured. Oppliger [9] suggested that many people who employ firewalls as a security measure might gain a false sense of complete security on the internal network. He claims they assume it will keep all bad traffic out of the internal network, thus eliminating the need for internal network protections. We show this is not the case, since an external web site set up by the attacker can attack the victim's internal network, using the browser as a conduit; the intrusion is accomplished through HTTP originating from inside the internal network — access allowed by nearly all firewall configurations. We stress that our proposed attacks do not deal with drive by personal computer infections; instead, the infection is targeted at the home routers and uses standard Web technologies (which have legitimate uses).

JavaScript Malware. It has been proposed that distributed denial of service (DDoS) attacks can be mounted from a set of clients visiting an attacker's site [7]. Using JavaScript (or similar technologies) a web site can send instructions to all of its visitors to create traffic at a victim's web site. We extend this DDoS idea by using compromised routers (which are more likely to remain in an attacker's control) as well as corrupt DNS records to create a large amount of unwanted traffic to a victim's site.

3 Intuition

Figure 1 shows how an internal network can be discovered and attacked, changing the configuration of a home router. Related work has exposed steps 1–4 of the attack shown in Figure 1. This paper explains how to accomplish step 5 (changing settings on a router) and what types of attacks this enables.

3.1 Attack Scenarios.

Access to a home router from the inside can lead to its complete compromise, making it a zombie performing actions at an attacker's will. This threat is significant since most zombified hosts are personal computers, which may be restarted or removed from a network frequently in the case of notebook computers. A home router is sedentary, and often left powered on, or unattended, for months

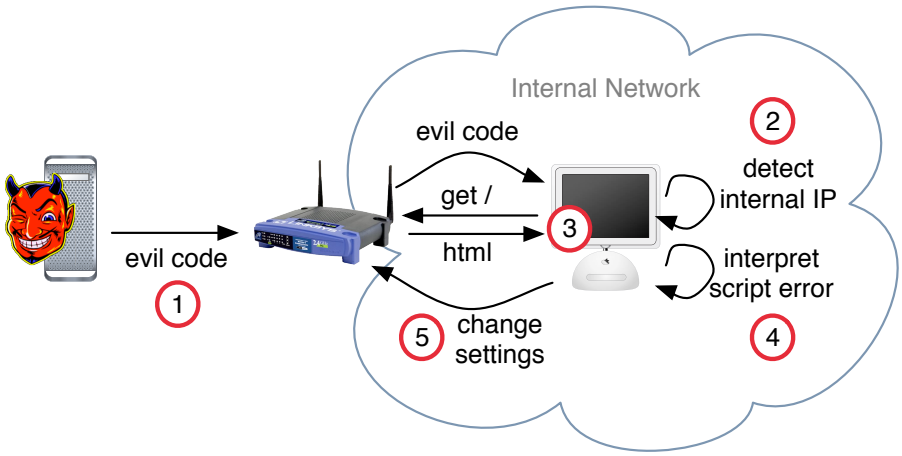

Fig. 1. How a home network's routers are attacked with Internal Net Discovery. (1) A client loads requests a page from the attacking server through the home router. The page is rendered and (2) the client's internal IP address range is either guessed or an Applet is run to detect the client's internal IP. (3) JavaScript requests resources from hosts on the network, which (4) throws JavaScript errors which the client-run page interprets to discover and fingerprint the victim's router. (5) The script attempts to change the discovered router's settings.

at a time, resulting in a zombie with a persistent Internet connection that more reliably responds to its controller. Additionally, home router compromise can lead to subversive DNS spoofing where DNS records are compromised on victims' local networks causing them to visit malicious sites though they attempt to navigate to legitimate ones such as `http://www.securebank.com`.

Security Patch DoS. An attacker could redirect requests for Windows updates or antivirus updates through his own server, which can starve affected clients from some critical patches: this leaves victims vulnerable to security flaws even after the are patched.

High-Yield Phishing. An attacker can redirect victims to his own copies of web sites that seem hosted at legit domains (like `bank.com`) with intent to harvest their passwords; the victims will most likely be oblivious of such a change.

High-Yield Malware. Using DNS spoofing through the compromised router, an attacker can pose as an authority (such as an antivirus website) suggesting that victims install malware that is advertised as critical software. To identify especially vulnerable targets, the attacker can record IP addresses of routers he compromises using Drive-By Pharming, and share this list with other people who have nefarious plans.

3.2 Internal Net Discovery

Since it is assumed that a network behind a firewall is safe from intruders [9], most commercial home network routers (including wireless routers used for sharing a broadband connection) are pre-configured out of the box to *disallow* administration features over the Internet, or Wide-Area Network (WAN) interface but allow administration over the internal network or Local Area Network (LAN) interfaces.

But as we describe, an attacker can still access the LAN-side configuration page from the WAN port due to the methods employed by many home users to make their single broadband connection accessible to their whole family. Most often, people purchase an inexpensive personal router/switch device to provide WiFi access to the Internet or to share a single broadband Internet connection with multiple computers. These devices usually include a NAT firewall and a DHCP server so connected computers do not have to be manually configured. Thus IP addresses are distributed to computers on the LAN from a reserved private IP space of `10.*.*.*` or `192.168.*.*`. Internet traffic is then routed to and from the proper computers on the LAN using a Network Address Translation (NAT) technique. Because of the employment of NAT, an attacker cannot simply connect at will to a specific computer behind the router — the router's forwarding policy must be set by the network's administrator in anticipation of this connection, thus preventing malware from entering the network in an unsolicited fashion. If a piece of malware were able to run on one of the computers behind the router, it would more easily be able to compromise devices — especially if it knows the IP addresses of other devices on the network. This is possible because it is often wrongly assumed that the router (or its firewall) will keep all the "bad stuff" out, so there is no dire need for strict security measures inside a home network.

3.3 Identifying/Configuring Routers

Once the internal IP of a victim has been identified, assumptions about the addressing scheme of the internal network can be made. For example, if Alice's internal IP is `192.168.0.10`, one can assume that all of the computers on the internal network have an IP starting with `192.168.0`. This knowledge can be used to scan the network for other devices, such as the router (steps 3, 4, 5 in Figure 1).

Using JavaScript (or similar logic developed with HTML and CSS), a malicious web page can "ping" hosts on the internal network to see which IP addresses host a live web-based configuration system.[3] More JavaScript can be used to load

[3] Most off-the-shelf routers are pre-configured to be the lowest address in the range they serve. For example, if Alice has internal IP `192.168.0.10` an attacker can comfortably assume the router has internal IP `192.168.0.1`. This *greatly reduces* the number of addresses that need to be checked before attempting to compromise a router; though it is not always accurate, this assumption should be acceptable in most cases.

images from these servers — images that will be unique to each model of router, giving the malicious software a hint about how to re-configure the host.

When a router's model is known, the malicious scripts can attempt to access configuration screens using known default username/password combinations for that specific router model. By transmitting requests in the form of a query string, the router's settings can easily be changed. The preferred DNS servers, among other settings, can be manipulated easily if the router is not protected by a password or if it uses a default password.

Owners of these routers are not required to set a password! Since administration via the WAN port (the Internet) is turned off by default, some manufacturers assume no administration password is needed. Membership of a router's internal network is not sufficient to determine that a person is attempting to change the settings of a router: it could instead be JavaScript malware as described.

4 Attacking a Network

An attacker who can detect a victim's internal network has the ability to attack the router controlling the network, and thus control any data going through the compromised router. To take control, first an attacker must discover the internal IP address of the victim's router. Next, the attacker must determine the make or model of the router in an effort to understand its configuration scheme, and then eventually accesses the router and manipulates its settings from the victim's computer. All of this can be done in an automated fashion, swiftly in most cases—when routers are configured with default passwords.

4.1 Router Discovery

Many home routers have a standard internal IP address (e.g., `192.168.1.100`). In other cases, the malicious web-site can deploy a very simple Java Applet [6] to detect the *internal* IP. Given the internal IP address of a host (e.g., `192.168.0.10`), other IP addresses that are likely to be on the internal network are enumerated (e.g., `192.168.0.1, 192.168.0.2, ..., 192.168.0.254`). Some JavaScript code then executes to append off-site `<script>` tags to the document resembling the following:

<**script src**="http://192.168.0.1"></**script**>

These tags tell the browser to load a script from a given URL and are commonly used to load off-site scripts with many purposes. One example of commonplace use of this is advertisement tracking: a web site embeds a script from `http://adsformoney.com` in order to display advertisements specified by the `adsformoney.com` web service. The script must be loaded from the advertisement company's site and not the publisher's site so that the advertisement company can verify the integrity of the script that is served. The effect of using script tags in this way is that a web-based request can be sent to an arbitrary server (or router) from a client's browser. Requests can thus be sent to another host on a victim's internal network through that victim's browser.

It is expected that all of these `<script>` elements will fail to load and generate a JavaScript error — the key is that they will fail in different ways. If the specified URL is a valid web server, the browser will fetch the root HTML page from that server and fail since the root HTML page is not valid JavaScript. If the specified URL is *not* serving web pages, the request will time out.

Leveraging error information, one can also determine the router's make and model. In particular, the router has a web server which might host an image (e.g., the manufacturer's logo). Using an `<IMG>` tag with `onload()` or `onerror()` handlers, one can detect the presence or absence of an image to determine which router is used.

4.2 Manipulating Routers

Routers with web-based configuration rely on HTML forms to obtain configuration data from a user. While most utilize the HTTP POST method to send data from the web browser to the router, many routers will still accept equivalent form submissions via HTTP GET. This means that form data can be submitted in the URL or query string requested from the router.

For example, the D-Link DI-524 allows configuration of the DMZ host through a web form. A DMZ or demilitarized zone host is a host on the internal network that is sent all incoming connection requests from the WAN. The form contains the input variables `dmzEnable` and `dmzIP4`. When sent the query string "`/adv_dmz.cgi?dmzEnable=1&dmzIP4=10`", the DI-524 enables DMZ and sets the host to `192.168.0.10`. Similar query strings can be constructed for other configuration forms.

Swift Attack Scenario. Additionally, it is important to note that all of these seemingly sequential attack stages can be accomplished in one step. Consider a web site whose only aim is to set the DMZ host to `192.168.0.10` on all networks using DI-524 routers with default passwords (the DI-524 has a null administrator password by default). The author of the site could embed this script tag in his HTML to attempt this attack:

<**script src** = ''http://<ip>/adv_dmz.cgi?dmzEnable=1&dmzIP4=10''></**script**>

This attack will only fail if the owner of the victim network has set a password or is not using a DI-524. Following is another plausible example that specifies a default username and password for a router:

<**script src** = ''http://root:pwd@<ip>/apply.cgi?DNS_serv=p.com''></**script**>

5 New Attacks

We have shown how routers' IP addresses can be discovered and their configurations can be changed using JavaScript. This leaves networks that are vulnerable to Internal Network Detection open to DNS spoofing, or Pharming, as well

as complete router control or zombification. Additionally, compromise of home routers has the potential to spread in a viral fashion, by turning routers into sources of the exploit, then advertising their existence.

5.1 Pharming

We implemented a DNS-configuration change by compromising a D-Link DI-524. By accessing a website with simple Java and JavaScript (as documented in Section 3.2) to detect our internal network we found the router's IP address. Additionally, the JavaScript was able to identify our router model by successfully loading an image. *We stress that this image was available from the router without authenticating. Only web page requests required authentication on this model.* Once the router model was identified (by IP address and image loaded), a request to change the DNS server settings was sent to the router:

<**script src** = ''http://192.168.0.1/h_wan_dhcp.cgi?dns1=w.x.y.z''></**script**>

This changed the DNS server address distributed by the router to `w.x.y.z`, one *other* than the one specified by our service provider. We set up a test (rogue) DNS server at this location, and included fraudulent DNS records for some popular web sites. When we attempted to access these web sites, the DNS requests were directed to the *new* DNS server specified by our exploit, and IP address to which the requests were resolved directed us to a fake page. We developed similar proof of concepts for the Linksys WRT54GS and NetGear WGR614 routers. We remark that when DNS server settings change, the browser will do a fresh look-up for each domain, so measures like DNS pinning [4] will fail to provide protection.

5.2 Growing Zombies

The DNS-server address-changing attack relies on the attacker controlling a DNS server. Another method of attack would be to modify the router's software to contain persistent false records [11]. The malicious firmware can be preconfigured to serve bad DNS data itself. Alternatively, the firmware can "phone home" to an attacker's server and identify itself as a compromised "zombie." This can be done much in the fashion that malware currently "zombifies" computers. These zombies can be configured to perform DDoS attacks or to allow the attacker to change the set of spoofed DNS records at any time.

A victim, who visits the attacker's web site, becomes vulnerable to internal network discovery, and thus router is compromised. Next, the malicious web page tells the router to enable "WAN port administration" so that an arbitrary Internet host can configure it. The victim's browser then contacts the attacking server to begin "updating" the router's firmware. The attacker's server, easily detecting the *external IP* of the router (which is the same as the external IP of the victim's computer) then accesses the router over the WAN port. The server uploads new firmware to the router, which is then configured to behave in any way the attacker desires. If desired, the attacker can ensure the router

will behave as it did before compromise, but with subtle modifications such as remote control. The victim's router is now a zombie under the control of the attacker.

Proof of Concept We implemented this router firmware modification by compromising a D-Link DI-524 as described. A client within the DI-524's internal network accessed our malicious web page. The web page targeted the DI-524, which was configured with default settings (no password). The web page caused the router's WAN port configuration to become enabled. Next, the web page sent a request-based message back to its' host server which used the remote IP of the request (the victim's and his router's external IP) to access the router with the default password (blank). Finally, the malicious server transmitted new firmware to the router, changing the version of the firmware on the router.

5.3 Viral Spread

Zombifying routers by replacing firmware can be deployed through a web page executing JavaScript on one of the router's internal network hosts. A router compromised in this fashion is open to a staggeringly large variety of purposes. There is nothing to say that the new firmware may contain web serving software and content *including the malicious scripts themselves.* Effectively, a compromised router could be transmogrified into a router that also serves the virus that compromised it.

An infected router could be instructed to use search engines to locate web-based bulletin boards, and post its address to lure readers into viewing its content. This mechanism would draw unwitting victims to infect their own networks—resulting in a spread from router to router via human-initiated transmission. This socio-viral spread, much like the social spread of other malware [10], will depend on the content of the viral site to spread itself.

This viral spread mechanism is hard to "shut down" since there are presumably many infected routers. In other words, there is no single source that can be turned off. In contrast, if a single web site hosts malicious code, then the site's owner can potentially be contacted for a takedown.

6 Conclusions

This paper described a new attack concept, termed *Drive-by Pharming*, that provides an alternate (and in our opinion easy-to-carry-out) method by which a pharming attack can be mounted. The attacker creates a web page, that simply when viewed by the victim, changes the DNS settings on the victim's home broadband router. From then on, when the victim navigates to his usual web sites, his data can be siphoned off to an attacker. The victim will likely be unaware that this change has taken place since even the address bar on his browser will indicate that he is viewing a legitimate web page. The attack requires that the victim run a JavaScript-enabled browser and use the default password on

their home broadband router – both of which happen reasonably often enough to make the attack's implications widespread.

We developed proof-of-concept code to demonstrate the Drive-by Pharming attack on three popular home broadband routers. We also explained how this drive-by modification of router settings can be used to help ease the spread of viruses by denying automated security upgrades and patches to victims.

The attacks demonstrate that even with well-defined security policies, extensive firewall and IPS rules, as well as the growing enforcement of the same-origin policies on web browsers, attacks based on Internal Network Detection are still very much possible. These attacks use the victim's browser as a conduit to their internal network. As the complexity of both the web and web browsing environment grow, we expect that many similar attack concepts will be discovered in the future.

7 Acknowledgments

We thank Alex Tsow for insight on how common home routers behave.

References

1. Matthew Braverman, "Windows Malicious Software Removal Tool: Progress Made, Trends Observed" Microsoft Antimalware Team Whitepaper, 10 November, 2006.
2. Mona Gandhi, Markus Jakobsson, Jacob Ratkiewicz, "Badvertisements: Stealthy Click-Fraud with Unwitting Accessories". Anti-Phishing and Online Fraud, Part I Journal of Digital Forensic Practice, Volume 1, Special Issue 2, November 2006
3. Jeremiah Grossman and TC Niedzialkowski, "Hacking Intranet Websites from the Outside: JavaScript malware just got a lot more dangerous." Black Hat Briefings, Las Vegas, NV USA 2006.
4. Mohammad A. Haque. DNS: Spoofing and Pinning. `http://viper.haque.net/~timeless/blog/11/`.
5. Sid Stamm, Zulfikar Ramzan and Markus Jakobsson, "Drive-By Pharming", Indiana University Computer Science Technical Report 641. December 13, 2006.
6. MyAddress Applet by Lars Kindermann, 2003. `http://reglos.de/myaddress/MyAddress.html`
   Page accessed October 17, 2006.
7. V.T. Lam, S. Antonatos, P. Akritidis, K. G. Anagnostakis, "Puppetnets: misusing web browsers as a distributed attack infrastructure." Proceedings of the 13th ACM Conference on Computer and Communications Security (CCS06), 2006.
8. Mary Madden, "Internet Penetration and Impact." Pew Internet and American Life Project Memo, 26 April 2006.
   `http://www.pewinternet.org/PPF/r/182/report_display.asp`
9. Rolf Oppliger, "Internet security: firewalls and beyond," Communications of the ACM Volume 40 issue 5. May. 1997, pp 92-102.
10. Sid Stamm, Markus Jakobsson and Mona Gandhi, "Social Propogation of Malware." `http://www.indiana.edu/~phishing/verybigad/`
11. Alex Tsow, Markus Jakobsson, Liu Yang and Susanne Wetzel, "Warkitting: the Drive-by Subversion of Wireless Home Routers." The Journal of Digital Forensic Practice, 2006.

12. Bob Violino, "After Phishing? Pharming!" CSO Magazine, October 2005. http://www.csoonline.com/read/100105/pharm.html
13. TheCounter.com Statistics, Jupitermedia Corporation. April 2007. http://www.thecounter.com/stats
14. "Survey Reveals the Majority of U.S. Adult Computer Users Are Unprotected from Malware" ESET software press release. 17 July 2006. http://eset.com/company/article.php?contentID=1553
15. "Gartner Says Number of Phishing E-Mails Sent to U.S. Adults Nearly Doubles in Just Two Years," Gartner, Inc. 9 November 2006 Press Release. http://www.gartner.com/it/page.jsp?id=498245
16. "Microsoft says Half of Windows Computers Have Trojans," Internet News, 26 October 2006.
http://www.internetnews.com/security/article.php/3640216
17. "Detecting, Analyzing, and Exploiting Intranet Applications using JavaScript," SPI Labs Research Brief. Accessed October 17, 2006. http://www.spidynamics.com/spilabs/education/articles/JS-portscan.html
18. The Symantec Internet Security Threat Report, Symantec Corporation. Volume 10, September 2006. http://www.symantec.com/enterprise/threatreport/index.jsp