

Attacker Techniques And Motivations

Muhammad Amar Primus Firdaus
11230940000067



Table of contents

01

**How Hackers Cover Their Tracks
(Anti Forensics)**

02

Fraud Techniques

03

Threat Infrastructure



Learning Objects



The Contents of Microsoft Windows Security Principle

How Hackers Cover Their Tracks (Antiforensics)	A. How and Why Attackers Use Proxies (Types of Proxies, Detecting the Use of Proxies, Conclusion) B. Tunneling Techniques (HTTP, DNS, ICMP, Intermediaries, Steganography, and Other Concepts, Detection and Prevention)
Fraud Techniques	A. Phishing, Smishing, Vishing, and Mobile Malicious Code (Mobile Malicious Code, Phishing against Mobile Devices, Conclusions) B. Rogue Antivirus (Following the Money: Payments, Conclusion) C. Click Fraud (Pay-per-Click, Click Fraud Motivations, Click Fraud Tactics and Detection, Conclusions)
Threat Infrastructure	Botnets, Fast-Flux, Advanced Fast-Flux



01

How Hackers Cover Their Tracks (Antiforensics)

How and Why Attackers Use Proxies

Masking one's IP address is a standard practice when conducting illicit activities. A well-configured proxy provides robust anonymity and does not log activity, thereby frustrating law enforcement efforts to identify the original location of the person(s) involved. A proxy allows actors to send network traffic through another computer, which satisfies requests and returns the result. Students or employees can use proxies to communicate with blocked services such as Internet Relay Chat (IRC) and instant messaging, or to browse websites that administrators block. While law enforcement can visit a physical location identified by an IP address, attackers that use one (or multiple) proxies across country boundaries are more difficult to locate (see Exhibit 2-1).

Proxies provide attackers with a way to lower their risks of investigator identification of their true IP address. In the hypothetical attack displayed in Exhibit 2-1, the victim's log file contains only one of the many IP addresses that investigators need to locate the attacker. Attackers operate free proxies or alter a victim's proxy settings because proxies can serve as a monitoring tool. Anon Proxy is one example of a malicious proxy that its authors designed to monitor users and steal information such as social-networking passwords.

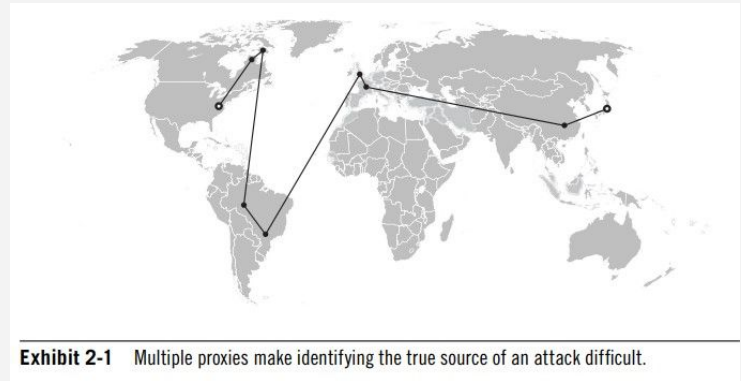


Exhibit 2-1 Multiple proxies make identifying the true source of an attack difficult.

How and Why Attackers Use Proxies



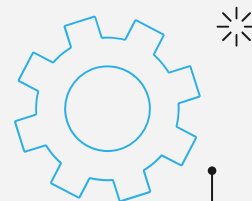
A. Type of Proxies

The most common proxies listen on TCP port 80 (HTTP proxies), 8000, 8081, 443, 1080 (SOCKS Proxy), and 3128 (Squid Proxy), and some also handle User Datagram Protocol (UDP). Attackers who install custom proxies often do not use standard ports but instead use random high ports. Some lightweight proxies are written in scripting languages, which run with an HTTP server and are easier for attackers to modify. Instead of configuring the application to use a proxy, users can tunnel all traffic through the VPN. VPN services usually support strong authentication and are less likely to leak information that could identify the user of a proxy.



Exhibit 2-2 Free and commercial proxies available from web-hack.ru.

Translated from Russian, these free Proxy and SOCKS services are updated every three hours; users can also purchase proxy access through the store. Attackers may prefer proxy services advertised on hacking forums because they are less responsive to abuse requests.



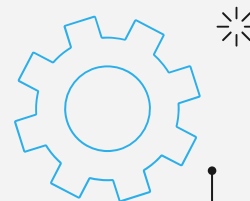
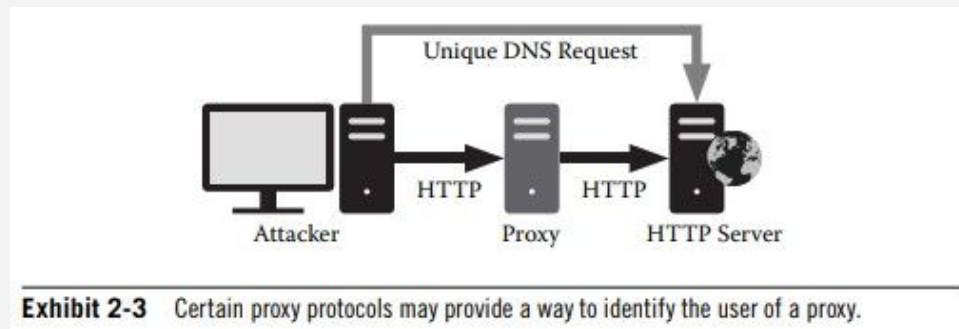


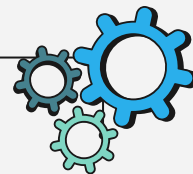
B. Detecting the Use of Proxies

Detecting proxies is difficult and not always reliable. Since many malicious code authors install custom proxies and use encrypted or custom protocols, it is very difficult to detect all proxies. There are techniques to detect common proxies, but such techniques are unlikely to be effective against attackers who use proxies aggressively. To detect proxies on the network with intrusion detection systems (IDSs), organizations may use proxy rules available from emergingthreats.net.³ The domain name system blacklist (DNSBL) is one example of a blacklist that allows administrators to block certain proxies.⁴ Certain proxies do not proxy all traffic. For instance, a Web application can force users to perform unique DNS requests with subdomains (see Exhibit 2-3).

The application links the DNS request to the user's IP address and verifies that the HTTP request originates from the same IP address. If they are not the same, indicating the use of a proxy, the application can determine that the proxy IP address made the HTTP request and that the user's actual IP address made the DNS request. Similarly, some Web plug-ins may query the local information rather than using the proxy address. As an example, decloak.net is a Metasploit project that uses the following application plug-ins to determine the true IP address of a proxy user:

- Word
- Java
- Flash
- QuickTime
- iTunes





C. Conclusion

Free and commercial proxies are very numerous on the Internet and can use standard protocols and ports. Other proxies are more difficult to identify, and administrators can detect the use of proxies through configuration changes, IDSs, or tools like decloak. net. Attackers who want to hide their locations have resources available to them. Since it is difficult to detect all proxy users accurately, proxy tools and services will continue to be useful for attackers.

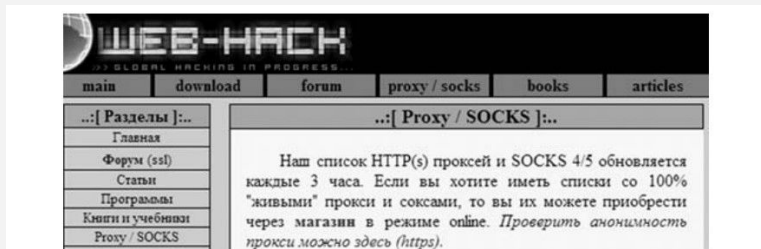


Exhibit 2-2 Free and commercial proxies available from web-hack.ru.

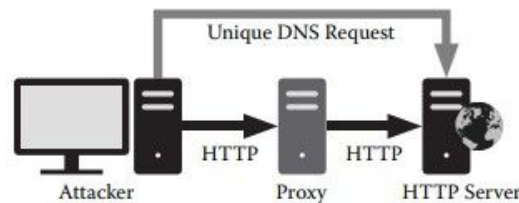
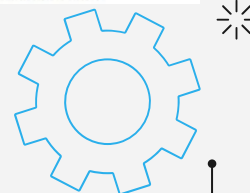


Exhibit 2-3 Certain proxy protocols may provide a way to identify the user of a proxy.



Tunneling Technique

Most enterprise security controls include strong firewalls, intrusion detection systems (IDSs), and user policies, such as proxies and time of-day rules that limit the amount and type of traffic generated on user networks. Tunneling data through other protocols often bypasses these controls and may allow sensitive data to exit the network and unwanted data to enter. It is even possible to extend all networks through these means without ever triggering an alert or log entry. Traffic tunneling in SSH involves proxying a TCP connection over an SSH session, allowing content to flow through the SSH connection. The server or client listens on a specified TCP port, transfers data to the other side, and forwards it to the specified TCP destination. SSH tunneling can be more complex, allowing for reverse tunneling or arbitrary application proxying through protocols like SOCKS.

Exhibit 2-4 shows how an SSH connection can tunnel Telnet connection securely between trusted environments. The example tunnels traffic between two unrelated hosts that have no SSH capability to illustrate the flexibility of the solution.

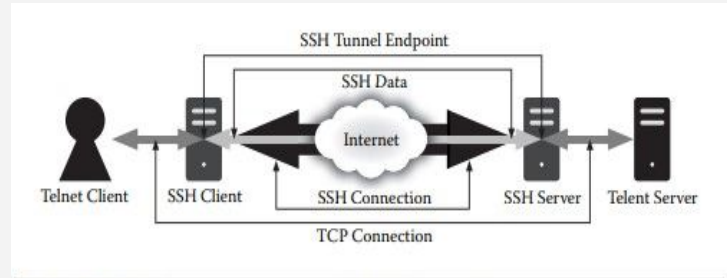


Exhibit 2-4 Telnet tunneled over a secure shell (SSH) connection.

Tunneling Technique

A. HTTP

HTTP has become the de facto high-level protocol on the Internet. As the protocol used for accessing content on the World Wide Web, developers adapted it to carry much more than just the static text and images of Web pages. It now carries audio and video streams, can transfer large files, and can even carry application to-application remote procedure calls (RPCs).

Exhibit 2-5 shows the syntaxes of an HTTP request and reply that illustrate areas of the protocol that can contain discretionary information for data transfer. As one can see, the protocol allows, in essence, unlimited space for content (or payload) in the request or reply message in addition to other open areas, such as the headers, whether this content includes arbitrary custom headers or inappropriate data invalid headers. This makes it convenient to transfer arbitrary data to and from an HTTP server. All one needs to tunnel the traffic is software that can pretend to talk to the protocol but in reality can transfer data for some other (perhaps nefarious) purpose.

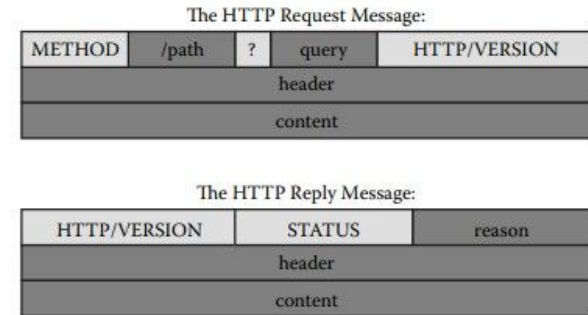


Exhibit 2-5 HTTP messages.

Tunneling Technique

B. DNS

The DNS is the core directory service of the Internet. Without it, translations between names, such as `www.verisign.com` and IP addresses, could not happen, and it would be difficult, if not impossible, to manage the daily operations of the Internet. Since DNS is a service that an administrator cannot block and must always make available, it is also a good choice for data exfiltration and tunneling. It is hierarchically decentralized so clients may not send transmissions directly to a specific end server, but other servers may relay it, and the size of the information contained in each burst of communication is relatively small. These features make the deployment of functional tunnels more difficult but not impossible.

Exhibit 2-6 shows the layout of DNS message packets; the darker areas indicate where software can hide payloads. Since there are few small areas where data can be stored, DNS tunnels need many packets to transfer large amounts of data and tend to be chatty. In most cases, the tunnel client is simple end user software that makes many requests for nonexistent hosts (where the host names are the encoded payload, such as `0123456789abcdef.badguy.goodguy`).

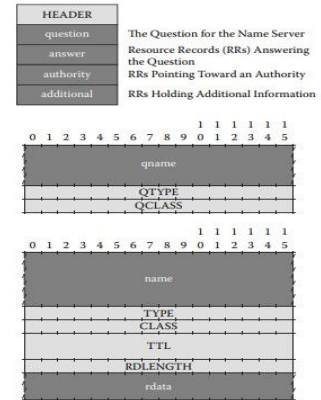


Exhibit 2-6 DNS messages. All communications inside of the domain protocol are carried in a single format called a message (top); the question section is used to carry the "question" in most queries (middle), and the answer, authority and additional sections all share the same format (bottom).

Tunneling Technique

C. ICMP

ICMP is a signaling protocol for IP. It is used mostly to deliver status and error messages when IP-based communication errors occur or to troubleshoot and test connectivity status. Although most enterprise policies already block outbound ICMP packets, some Internet service provider (ISP) solutions may not, and its use as a tunnel is mostly to bypass ISP authentication requirements or as a simple covert channel. Most ICMP messages offer little in the way of embedding payload, and implementation details may make it difficult to get the messages delivered; however, ICMP echo messages, which users and administrators alike use to test the accessibility of a host, are well suited for tunneling. Ping, as the most common software that implements this ICMP mechanism, sends data to a host and expects a reply.

Exhibit 2-7 is the layout of an ICMP echo message, again showing payload areas. As shown in the protocol illustration in Exhibit 2-4, there are plenty of data in which to place payload for a tunnel; therefore, ICMP offers good throughput in both directions. ICMP tunneling was one of the earliest 15 methods publicly available to transmit traffic over a protocol in a covert way that essentially abused the protocol. The open source community actively maintains and makes available several software packages to provide this functionality, including Ping Tunnel, ICMPTX, Simple ICMP Tunnel, and Skeeve.

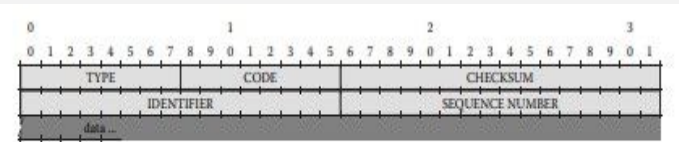


Exhibit 2-7 Internet Control Message Protocol (ICMP) echo message.

Tunneling Technique

D. Intermediaries, Steganography, and Other Concepts

Aside from the three common tunnels discussed in the previous paragraphs, hackers can modify any protocol that filters through a firewall to behave as a tunnel. Assuming deep-packet inspection requires that a given protocol at least match the appropriate syntax (headers must match, no arbitrary data, etc.), tool writers can coerce even FTP, SMTP, and the like into becoming covert channels. All such tunnels have one thing in common, however: it is apparent where their destinations lie. The task of tracking down specific tunnels and at least shutting down those that are readily apparent is a quick step: identify the destination and block it. This task can become much more difficult with advanced implementations such as intermediary hosts, however. For example, it is possible to create an HTTP tunnel that does not just connect directly to its destination, but rather drops a payload onto some public or common service (forums, comments on blogs, image hosts, etc.)—services that may have legitimate uses. Once the payload arrives at the intermediate service, the destination side of the tunnel picks it up and delivers a reply—all without ever revealing the malicious origin to the exfiltrated network. Recently, researchers discovered such a scheme on Twitter.²⁰ The intermediary problem can be even more complex. Steganography is the practice of hiding messages and data in content that is not readily apparent and is a form of security through obscurity. For example, steganographic software and tools can encode messages and data into images²¹ so that only users who know where the data exists can retrieve it. Tunnels that use intermediaries for data exchange can deposit payloads that are steganographically encoded to make it harder to detect the covert communication.

Tunneling Technique

E. Detection and Prevention

The potential of covertly extending a network to the outside world is a clearly unacceptable risk. While the firewalls and IDS that are in place today have their roles to play, they may not be able to identify or prevent tunneling. Tunnels abuse protocols in a way that matches the syntax or the rules of the specifications but not the intent, so despite the efforts of vendors using static signatures for detection—for example, iodine signatures available for Snort—it is trivial to “hack” tunnels to foil the current crop of defenses. Attackers can easily modify open-source tools to appear slightly different from the original, thus defeating a static rule. Any protocol can quickly become a tunnel harbor. Packet inspection firewall rules and IDScan go only so far in identifying and blocking the threats. Tunnels do have a weakness. They almost never adhere to historical or trended traffic patterns. While HTTP normally has small transfers outbound with larger transfers inbound, tunneling may cause this to reverse or become nearly equal. The duration of connections may also buck the trend, as tunnels need things like keep-alive messages and timeouts. In the case of DNS tunnels, the amount of requests per client, or a set of clients, or even across the enterprise may jump significantly. In the case of ICMP, packet sizes may not match the expected norms, and with any other tunnels, the ratios of different protocols, frequency, and volume can all be indicators of anomalies. These markers point to the need for traffic analysis. Using net flows and other packet capture and aggregation tools, it becomes a statistical problem to map enterprise network trends and identify anomalies. While the crop of commercial tools is limited, several open source solutions are available to begin the process of at least watching and understanding what is going on in the network. Tools like Silk and Sguil can become the gateway to better understanding. They can provide a foundation for trending the network and baselining behavior. Although it may be a labor-intensive process, using it is better than not knowing.



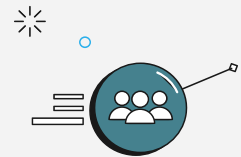
02

Fraud Techniques

Phishing, Smishing, Vishing, and Mobile Malicious Code

Many phishing attacks against mobile devices use short message service (SMS, or smishing) and voice-over Internet protocol (VoIP, or vishing) to distribute lures and collect personal information. Attackers often send fraudulent SMS messages containing a URL or phone number using traditional phishing themes. Responders either enter their personal information into a fraudulent website, as with traditional e-mail phishing, or, if calling phone numbers, may even provide their information directly to other people. To limit exposure to these growing threats, organizations should not send contact information to users via SMS but instead should be sure phone numbers are readily available on their websites. Phishing by way of mobile phones introduces new challenges for attackers and administrators alike.

Many phishing attacks against mobile devices use SMS (smishing) and VoIP (vishing). Attackers often send fraudulent SMS messages to many users attempting to gain private information or distribute malicious files. The messages include a URL or a phone number with themes similar to those of traditional phishing messages. Upon calling a phone number, the user may interact with an actual person or a voicemail system—both of which are risks to the user's personal information. Many legitimate services suffer from doubt and uncertainty related to sending legitimate SMS messages. Organizations should avoid repeating mistakes made with e-mail, which for many organizations is no longer a viable means of communicating with customers due to the pervasiveness of phishing and other fraud.



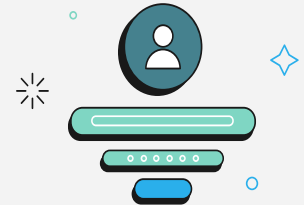
Phishing, Smishing, Vishing, and Mobile Malicious Code

A. Mobile Malicious Code

Although rare and only a more recent occurrence, SMS messages sent to mobile devices may also attempt to convince users to install a mobile malicious code. On or before February 4, 2009, Chinese mobile phone users began reporting a new virus that affects Symbian S60.25 A signature is required on all code that runs on the S60 third edition, and this virus is no exception; it uses a certificate from Symbian licensed to “ShenZhen ChenGuangWuXian.” After the user installs the program, it spreads to other users by sending SMS messages that contain URLs, such as the following, for users to download and install the code:

- hxxp://www.wwqx-mot.com/game
- hxxp://www.wwqx-cyw.com/game
- hxxp://www.wwqx-sun.com/game

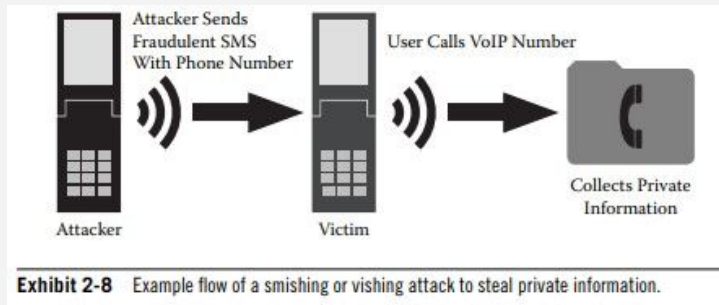
The “Sexy View” virus attempts to convince recipients to download and install a Symbian Installation file (SISX) at the URL, but it does not use any exploits to install automatically. Details on this virus are publicly available.²



Phishing, Smishing, Vishing, and Mobile Malicious Code

B. Phishing against Mobile Devices

Most instances of SMS phishing (smishing) target banks or financial institutions by sending a phone number that the victim calls after receiving the message, resulting in a vishing attack (see Exhibit 2-8).



In the past, attackers used vishing against random targets and were successful at avoiding defensive filters. For instance, actors have used SMS gateways that allow users to send emails instead of spending money per SMS message. In this way, actors send messages to all possible SMS recipients for a gateway. As an example, the SMS gateway receives e-mail messages sent to the phone number 111-222-3333 at the e-mail address 1112223333@mobile.gateway.example.com. SMS gateway providers have responded to abuse by rejecting excessive numbers of messages or fraudulent messages.

Phishing, Smishing, Vishing, and Mobile Malicious Code

C. Conclusion

To combat the uncertainty caused by smishing and vishing, organizations that plan to contact users via SMS should not encourage users to depend upon caller ID, phone numbers, or the contents of a message. To limit exposure to these problems, organizations should clearly advertise their legitimate SMS numbers via their website and avoid sending phone or SMS contact numbers within messages whenever they contact users.

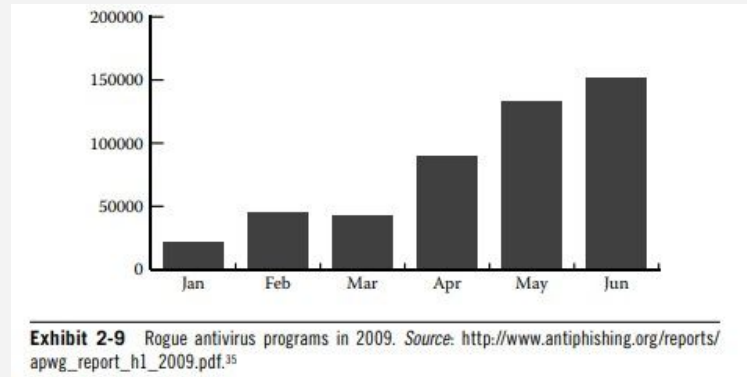
Concerning mobile phishing threats, financial institutions should take great care to educate their customers regarding how they plan to offer services and communicate via mobile devices. Additionally, customers should avoid accessing online banking through mobile devices until the platforms implement stronger anti phishing measures that are on par with desktop solutions. Some institutions choose to implement custom applications for mobile access to online banking, which may mitigate this threat when consumers use that as the sole mobile access method. Finally, financial institutions should carefully consider using mobile devices as two-factor authentication devices, given that customers may use the same mobile device to access the online banking system.



Rogue Antivirus

During the past year, fake antivirus programs have become dramatically more prevalent and are now a major threat to enterprises and home users. Moreover, attackers often bundle this software with stealthier malicious programs. Fortunately, in attackers' attempts to get users' attention, rogue antivirus software also alerts administrators to system compromises and inadvertently exposes other malicious software. Attackers aggressively target users with Trojan applications that claim to be antivirus programs.

(see Exhibit 2-9). According to Luis Corrons of PandaLabs, his company observed a significant growth in rogue antivirus applications from January to June 2009, the highest being in June 2009 with 152,197 samples. One possible reason for the increase is that pay-per-install and affiliate programs encourage more attackers to install such software. According to some pay-per-install rogue antivirus sites, affiliate programs offer an attacker approximately half of the purchase price for each victim who buys the software.³⁶ This encourages a diverse group of attackers to distribute the software. Though rogue antivirus software emerged in 2004, iDefense has observed a huge increase in this type of malicious activity between 2007 and 2010.



Rogue Antivirus



A. Following the Money: Payments

Most of the rogue antivirus incidents that iDefense investigated use third-party payment organizations. These organizations accept credit card payments and create a layer of protection and security for attackers who use them. These payment processors typically use legitimate SSL certificates and claim to handle fraud requests and operate on a permanent 24/7 basis. The payment processors' connection with rogue antivirus vendors is not exclusive; therefore, law enforcement cannot always shut them down immediately. In past instances, iDefense reported the abuse to the appropriate certificate authorities.

Afterward, authorities were able to take the payment processors offline. In many instances that iDefense investigated, several similar payment providers exist on the same IP address. The payment providers are highly suspicious because they use multiple registration names, domains, and contact addresses and countries, despite their singular purpose to accept money for rogue antivirus payments. Several of the payment provider sites do not list a phone number unless replying to an authorized customer. They also list in their terms of service that they avoid taking responsibility for customer content.

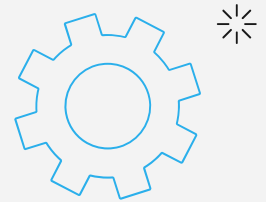
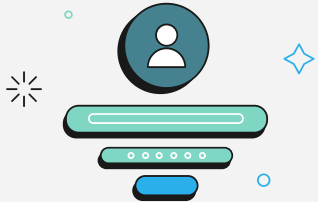
B. Conclusion

A large variety of attacks that install rogue antivirus applications exists. Many use social engineering because it seems somewhat more likely that attackers will be able to convince a victim of social engineering to pay for rogue antivirus software. However, attackers install it using a variety of other techniques and themes due to the pay-per-install model. Organizations that fall victim to rogue antivirus software should evaluate infected computers for bundled software that often accompanies the malicious programs and that may go unnoticed by victims who attempt to disinfect their computers.

Click Fraud

Having provided revenue for a substantial portion of online activity, advertising on the Web has largely been a success for advertisers and online companies. Not surprisingly, fraudsters abuse ad networks by generating invalid traffic for profit, competitive advantage, or even retribution. Advertisers complaining about charges for false traffic have made combating click fraud a major issue for online advertisers. As with most “free” content in other media, advertising funds much of the World Wide Web; however, unlike the world of television and print ads, it is very easy to become an ad publisher on the Internet.

The Web is interactive and allows advertisers to know exactly how many potential customers viewed an ad and how many clicked the ad. This knowledge leads to an advertising model known as pay-per-click (PPC), in which advertisers pay ad publishers each time a potential customer clicks an ad on the publisher’s website. This direct relationship between the number of clicks and the amount of money earned by the publisher has resulted in a form of fraud best known as click fraud. Anchor Intelligence reports that in the second quarter of 2009, 22.9 percent of ad clicks are attempts at click fraud.³⁷ In this section, we will look at how criminals make money through click fraud and how compromised computers make preventing this type of activity very difficult.



Click Fraud

A. Pay-per-Click

Any advertising transaction has three primary parties: the advertiser, the publisher, and the viewer. The advertiser is a company that produces content it would like to display to potential customers. This content is an advertisement for a specific product or service that is likely to generate revenue for the advertiser. The publisher is a creative outlet that produces content that will draw visitors to its medium. These visitors view the ad and, ideally, purchase the advertised product or service. The advertiser pays a fee for a specific number of “impressions,” which is the estimated number of times a viewer will see the ad. This model is essentially the same across all forms of media, including print, radio, and television.

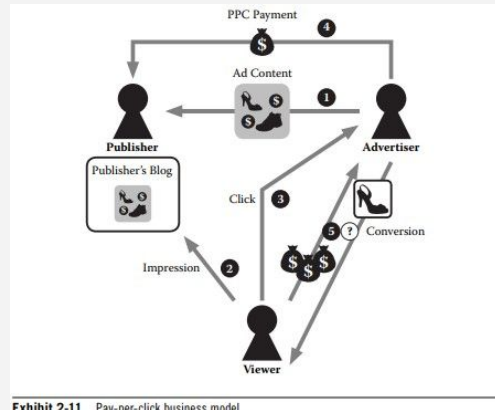
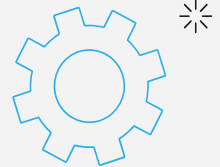


Exhibit 2-11 Pay-per-click business model.

Exhibit 2-11 shows how money flows in this business model. With the advent of PPC advertising networks like Google AdWords and Yahoo! Search Marketing, anybody with a website can become an ad publisher. Publishers who use these networks are affiliates. Affiliates add HTML code to their website, which draws ads from the advertising network and displays them inline with the affiliate's content. The affiliate and the advertising network then split the PPC fee each time a viewer clicks an ad.



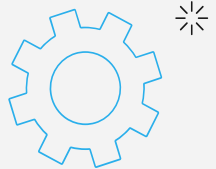
Click Fraud

B. Click Fraud Motivations

Click fraud occurs when an ad network charges an advertiser for a click when there was no opportunity for a legitimate conversion. There are many possible motivations for a person to click an advertisement without any intention to purchase a product or service. Publishers perform the most obvious and common form of click fraud. Clicking an ad on one's own website directly generates revenue for the publisher. Clicking the ad fifty times generates even more revenue.

While a publisher can click his or her own ad, he or she could just as easily ask friends to click the ads. For instance, a blogger who wants to increase revenue might make a post simply asking his or her readers to click every ad on his or her website each time they visit. While they are legitimate users, these clicks will not result in a conversion for the advertiser.

An advertiser's competitor might also be inclined to commit click fraud. If each click costs the Acme Corp. money, Acme's chief rival might click the ad a few hundred times a day to cost them as much money as possible. In this case, the publisher benefits from the fraudulent clicks, but the motivation is merely to harm the advertiser.



Click Fraud

C. Click Fraud Tactics and Detection

The simplest form of click fraud involves manually clicking advertisements through the browser. While this method is effective at generating a small number of additional clicks, fraudsters have developed sophisticated methods to produce the volume necessary to earn higher revenue.

First, the fraudster must create a website that displays advertisements. A popular way to do this is to create a search engine that only displays advertisements relevant to a queried word. One such search page uses a very unlikely typo of google.com, gooooooooooogle.com. The top portion of Exhibit 2-12 shows the results returned when searching this page for “puppies,” and the bottom portion shows advertisements displayed on Google’s search page when querying for the same word. All of the results returned by gooooooooooogle.com are actually advertisements, and many of them are the same ads returned by a Google search for the same term. A portion of the fee that advertisers pay for each click will go to the owners of gooooooooooogle.com. With the advertisements in place, the fraudster must now find a way to click as many of the ads as possible without the ad network noticing the abuse.

Botnets, the Swiss Army knife of the modern Internet miscreant, are the key to a successful click fraud campaign. As the click fraud problem grew, ad networks began developing fraud detection mechanisms that made simple click fraud impossible. For instance, when a single IP address registers multiple clicks in a 30-minute period, the ad network may simply discard all but the first click when

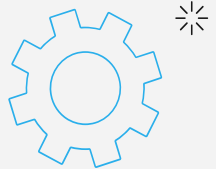


Click Fraud

D. Conclusions

While click fraud appears to be a problem with a scope limited to just advertisers and ad networks, fraudsters' use of infected computers to click ad links makes click fraud a problem for everyone with a computer. Being part of a click fraud botnet consumes a system's bandwidth and displays additional advertisements to the user, which is usually undesirable.

Companies should be cautious when spending advertising money on the Internet and should check which techniques their publishers use to detect and prevent click fraud. Organizations that already advertise on the Internet and are concerned that they may be victims of click fraud can use the techniques described in this section to detect some forms of click fraud. Companies such as Click Forensics³⁹ and Anchor Intelligence⁴⁰ provide third-party solutions to assist in discovering and weeding out invalid ad clicks.





03

Threat Infrastructure

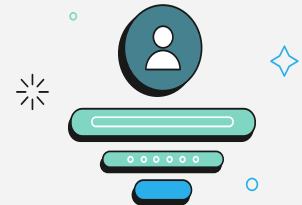
Threat Infrastructure

A. Botnets

Systems connected to the Internet are at risk of infection from exposure to social-engineering attacks or vulnerability exploitation. Regardless of the infection vector, compromised machines can wait for commands from the attacker, which turns the system into a bot. A bot is a single node added to a network of other infected systems called a botnet.

A botnet is a network of infected systems controlled by an administrator known as a botmaster. A botmaster controls many bots by issuing commands throughout the botnet infrastructure. The ability to run commands on many systems makes botnets practical for malware authors seeking a management solution and provides multiple capabilities.

Botnets would not be capable of performing any activities without communication between the botmaster and bots. The type of communication protocol depends on the network topology of the botnet. While botnets use many different topologies, all botnets fall into two main categories, centralized and decentralized; however, some botnets implement elements from both categories to create a hybrid structure.



Threat Infrastructure

A. Botnets

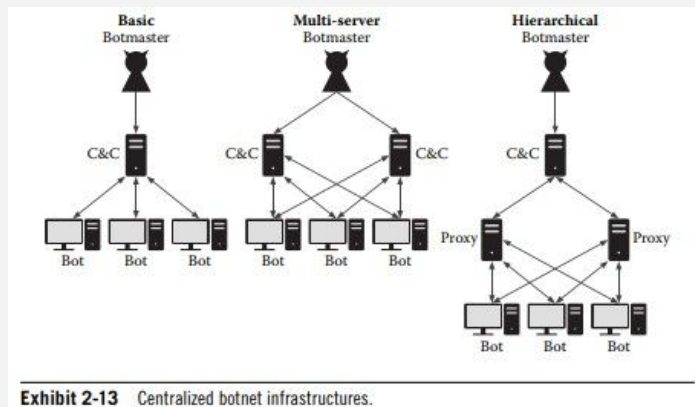
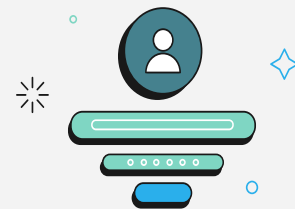


Exhibit 2-13 shows the infrastructures of the different centralized botnets. A multi server builds on the basic centralized botnet topology by using more than one server for C&C. Multiple C&C servers make botnets more reliable and less vulnerable to takedown attempts. This type of topology allows bots to receive commands even if one server is unreachable. If a server goes offline, the botmaster can still communicate with bots through other C&C servers. The Asprox botnet was an example of this type of botnet as it issued a configuration file to each bot that included a list of C&C servers.

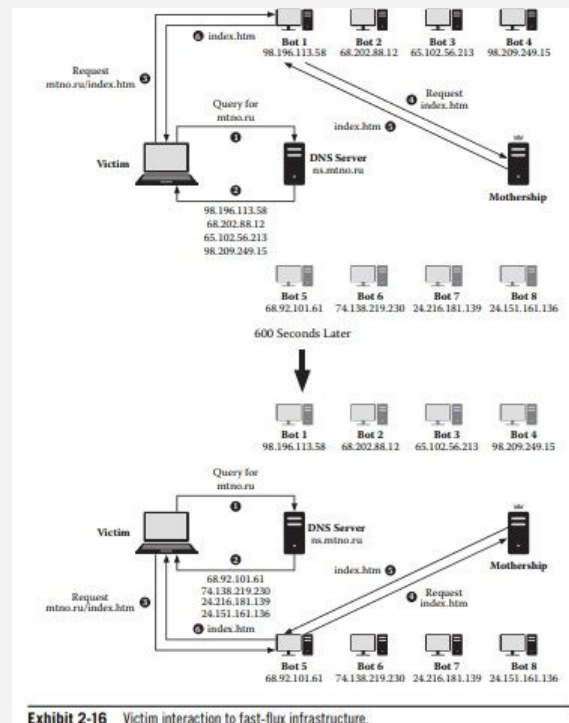


Threat Infrastructure

B. Fast-Flux

This section explains fast-flux attacks, which utilize temporary records in the domain name system (DNS) to achieve a distributed network of hosts. Resulting in constantly rotating IP addresses, single-flux attacks require disabling the DNS server to take down a domain. Even more robust, double-flux attacks rotate the IP addresses of both hosts and domain name servers, thereby making takedown even more difficult. In addition, fast-flux attacks create a layer of anonymity, further frustrating enforcement efforts. This technique is not new, but it grew dramatically in popularity within the past two years and is now a common occurrence. Exhibit 2-16 describes a concept known as single-flux, which utilizes static name servers (DNS NS records) to update DNS A records with IP addresses of infected computers and continuously cycle new IP addresses.

Client requests for a single-flux domain received by the domain's name server are resolved to multiple bots (bots 1–8 in Exhibit 2-16). All requests to the domain go through infected computers to the mothership. Acting similar to a reverse proxy, the infected computers receive the mothership's response to the client's initial request and forward it to the client. Multiple other variations of fast-flux currently exist, and all incorporate the same techniques with different levels of complexity. Double-flux adds a second layer of complexity by cycling IP addresses to name servers, and hydraflux introduces multiple motherships



Threat Infrastructure

C. Advanced Fast-Flux

Described the workings of a basic fast-flux infrastructure, which uses DNS records to obfuscate the location of malicious sites and frustrates takedown efforts. Variants of the fast-flux technique further complicate tracking and takedown by using multiple domain name servers or even multiple command-and-control (C&C) servers, also known as motherships. These advanced fast-flux methods are known as double-flux and hydra-flux.

Exhibit 2-17 shows the protective layers that each type of flux provide the source of the malicious content or actor. The diagram shows single-flux using groups of bots, a single DNS server, and a single mothership that provide three layers of protection between victims and the malicious actor or content. The use of a single DNS server is a shortcoming in single-flux by presenting a single point on which to focus efforts to stop malicious activity. Double-flux addresses this flaw by adding multiple DNS servers, which in turn adds another layer of protection for a total of four layers. The multiple DNS servers increase the complexity of the infrastructure to conceal the content source further, but double-flux's weakness lies in the use of one mothership.

Once discovered, investigators can take down the single mothership to stop malicious activity related to the double-flux domain. Hydraflux fixes this vulnerability with an extra layer on top of double-flux by implementing multiple motherships. With five layers of protection, hydra-flux is the most advanced type of fast-flux and causes hardship to those attempting to stop malicious activity. With these layers of multiple bots, DNS servers and motherships render server takedown procedures ineffective. The best method to cease malicious hydra-flux activity requires the domain registrar to suspend the domain name.

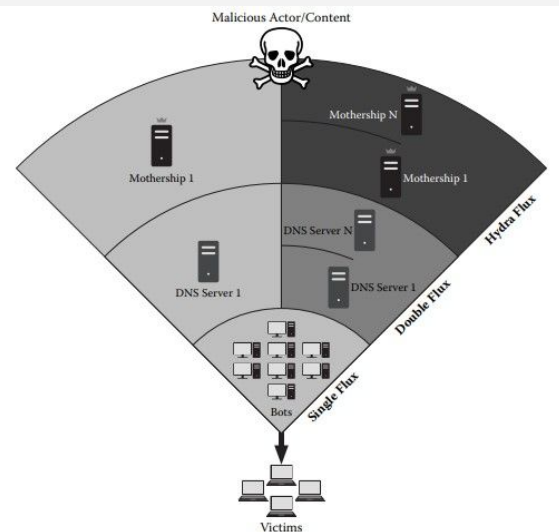


Exhibit 2-17 Diagram of the protective layers that flux domains provide.



Thank You