Criminalization only really works when it has a suitable deterrent effect or if the incarceration of those responsible keeps sufficient proportions of potential malefactors away from opportunity. Other forms of crime are also being linked to cybercrime in recent developments. For instance, DDoS attacks are now being launched and stopped, followed by a demand for payment to avoid further attacks. Businesses which operate solely or primarily online are the main targets of such attacks – online casinos for instance. Small e-commerce sites are also often targeted. Such small operations often do not have the necessary technical expertise to understand how to quickly shift their servers and are often reliant on low-cost ISPs (their margins may be small, so low-cost ISPs are necessary parts of their cost base) who may not themselves have the capability to protect the site against DDoS attack.

DISCUSSION TOPICS

Grey Hat Cracking Should Be Legalized

Analogies to homes and businesses abound when discussing the ethics of grey hat cracking. If someone leaves their front door open does that give you the right to enter and rifle through their private belongings? Many small businesses do not have significant security while people are at work, so does that give you the right to walk in and start looking through unlocked filing cabinets? A policeman sees someone walking down a street at night checking every door to see if it is locked. On confronting him, he finds that the man is wearing a striped shirt and carrying a bag marked 'SWAG'. Well, okay, maybe he is carrying a set of tools that would allow him to bypass an alarm system and sundry other breaking-and-entering equipment. In the United Kingdom, he can be arrested and charged for 'going equipped'. There have been proposals to define 'going equipped to crack' as a computer crime, combining the presence of tools used for cracking with the Internet equivalent of trying every door in the street, which is systematic port scanning. The UK CMA incorporates a small aspect of this, in that there is no requirement to demonstrate that a cracker deliberately targeted a particular system, just that they did gain unauthorized access to it, while intending to gain

unauthorized access to something. In reaction to the growing problem of computer crime, we might make the laws more draconian. However, this ignores the fact that burglars are a small percentage of society and that an even smaller number of police are sufficient to provide reasonable insurance against them or to capture them after their activities. On the Internet, however, there appear to be too many crackers and not enough policemen (the lack of qualified and employed law enforcement officers is a constant complaint of business but no one wants to pay for enough double-qualified specialists – computer experts who are also trained policemen).

In such circumstances perhaps it is time to allow a certain sort of vigilantism: not the type that goes after the offenders, but those who check up on the security of others as a pastime but intend to do no damage. Perhaps the litmus test for criminality should not be the accessing of information but what is done with that information and how the intruder acted: whether they installed a back door for further access, whether they read large amounts of data or 'sampled' what was there to estimate the value of where they'd penetrated, whether they quickly and efficiently informed the victim of their successful attack and provided information on plugging the hole. As we said in the introduction to this chapter, criminality is defined by society and if the actions of grey hat crackers have positive results, assuming sane and positive attitudes on behalf of the 'victims', surely they are to be applauded and encouraged for identifying security problems that might allow black hat attacks to succeed, rather than being criminalized.

Web Scrapers and Robot Denial Files

In 1993, the first forerunners of Google were being developed. In keeping with the concept of the Web of information, with threads of hyperlinks being the structure, spiders (or web crawlers) were developed. However, many of the sites linked to the Web at that time had limited bandwidth and processing power available to them. In fact a number of the machines running as web servers were simply individuals' desktop machines. This was all very well for the usual amount of web access that went on in those days when there were hardly more people online than were publishing information. There was no Google and, in particular, there was no

Slashdot to generate overwhelming traffic to sites: people found new information by following individual links. Denial of service problems caused by too many people requesting pages from a low-bandwidth site were rare. Into this sphere, the first web crawlers emerged. Instead of requesting a single page from a site, looking through it and maybe requesting one or two more pages within minutes, the simple early spiders would download a document, scan through it for other links and download all those documents as well, indexing the original document as a separate process. If you were running a site with many documents that had internal links between them a single visit by a spider could end up requesting the entire site within a few seconds. Given the low bandwidth and low power of the server this could amount to a denial of service attack by accident. However, the new spiders were the beginnings of a useful communal tool for the Web. Without Google and its predecessors, the Web would not be the immensely useful tool for finding information that it is. Instead of making accusations of criminal behaviour or trying to block requests from spiders completely, a protocol was developed where web servers have a configuration file called 'robots.txt' at the top of their domain (see `www.robotstxt.org` for the current protocol). To act responsibly, the authors of web crawlers programmed their systems to read this file and then follow the limitations it described. Originally this was simply a list of directories not to be crawled – pages under construction, placeholders, and so on. Later, more sophisticated protocols were added such as indicating dynamically created pages that should not be requested, speed of requests to ensure that a low-power server was not overwhelmed and so on.

A more recent variant of the web crawler is the web scraper, which looks for certain kinds of information – prices of particular goods from various online stores for instance – and then aggregates it into new web pages, often using a database. The originators of this information are usually unhappy with such things as they are likely to attract customers away from their sites. As a consumer one might think this is reasonable but the world of e-commerce is not that simple: aggregate deals, special offers, tax, postage and packing charges all differ between sites and can lead to the headline prices produced by web scrapers being unrepresentative of the actual deals on offer. In addition, some web scrapers download almost the entire contents of another site and automatically create a copy (or at least a close equivalent). This is a violation of copyright and can lead to problems such as out-of-date information. Since web scrapers are automated services it should be possible to have the robots.txt protocol indicate in a fair amount of detail what the limitations are for automated access. In fact, the protocol does include at least a coarse-grained version of this. However, the users of web scrapers are often making a profit from their actions and ignore the protocol. Should web scraping which ignores the robots.txt instructions be considered hacking? It is, after all, a violation of the implicit 'terms and conditions of use' of web sites and the argument that this is an automated system that cannot distinguish the terms of use is falsified by the existence of the robots.txt protocol.

An Immune System for the Internet

The number of worms and viruses that exploit known vulnerabilities for which patches have been released but which still cause problems for many users is large and growing. Even if a machine is up-to-date with patches, constant bombardment from infected machines can cause denial of service or at least an annoying slowdown. There have been a number of apparently benevolent worms released in the past few years which attempt to patch vulnerable systems and block the entry of other worms and viruses. Given that so many business and home users seem reluctant to take measures to keep their machines patched and part of the solution rather than part of the problem, it's time for the good guys to fight fire with fire and release well tested and robust worms which carefully fix the problem on any system they infect and then cautiously seek out other vulnerable hosts to fix. There will be some problems caused: patches are not always compatible with other programs; it is the propagation of some worms that causes a denial of service and not a payload action; installing patches often requires a reboot. However, since the machines which are subject to most of the problems are the unpatched yet still networked machines which cause so many problems for others (including the spread of malicious worms), it is the fault of the owner of the machine: if they kept their machine patched and secure they would not be vulnerable.