MITIGATING EMERGING HACKER THREATS

By Peter Mell and John Wack
Computer Security Division
Information Technology Laboratory
National Institute of Standards and Technology

PLEASE NOTE (Updated Oct. 2010): Any mention of the ICAT website or URLs to the ICAT Metabase, the ICAT Metabase has been renamed to the National Vulnerability Database (NVD). The URL to the NVD website is: http://nvd.nist.gov/ . The NVD website is hosted by NIST's Computer Security Division.

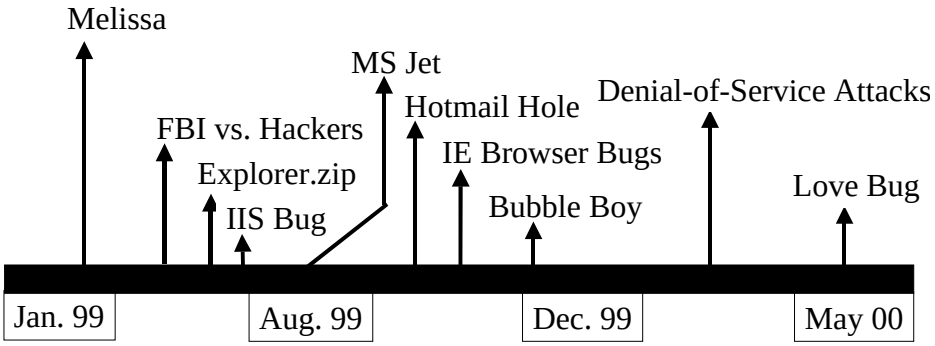
Introduction

It seems that every week, computer security organizations are issuing press releases concerning the latest hacker attack. Some sound dangerous, like the "Killer Resume," or mysterious like the "Mstream" distributed denial-of-service (DOS) program, or cryptic like the "cde-dtprintinfo" vulnerability. Each announcement represents a new threat that organizations must take seriously if they are to protect themselves, because even a single security hole can make an organization's networks vulnerable to a determined and persistent hacker.

The complexity and frequency of these announcements can be overwhelming to organizations, causing them to get lost in the details and to lose sight of the overall landscape of hacking events. This *ITL Bulletin* addresses the overall picture, trends, and solutions. First, we review the most significant computer vulnerabilities and attacks that have occurred in the past 16 months. Next, we discuss both novel and continuing hacking trends. Finally, we summarize the threats created by these new trends and techniques, and provide guidance on mitigating that threat.

Timeline of Major Hacking Events

The timeline shows the major alerts and "hacks" of the past 16 months. Each event shown had either a large impact or was indicative of a new trend or technique.



Timeline of major hacking events since January 1999

Event: Melissa Virus
Date: March 1999

Melissa is a Microsoft® Word macro virus that propagated itself through e-mail attachments. When the attachment was opened, it would mail itself to the first 50 users in a victim's Microsoft® Outlook® address book. The primary impact was that e-mail servers worldwide

were overwhelmed with copies of the virus. Melissa was a milestone virus because it marks the first time in ten years that the Internet had experienced a widespread worm.

Event: FBI vs. Hackers
Date: May 1999
The FBI investigated several United States-based hacking groups. After the seizure of a teenager's computer, several hacker groups retaliated by defacing many insecure government Web servers. At one point, a denial-of-service attack caused the FBI Web site to be taken offline for seven days.

Event: Explorer.zip
Date: June 1999
The Explorer.zip worm was very similar to the Melissa virus since it spread primarily through e-mail attachments that, once opened by a human, would automatically mail themselves out. Like Melissa, its impact was widespread. More importantly, perhaps, was the capability of Explorer.zip to spread between computers using network-sharing vulnerabilities in addition to e-mail. The network-sharing aspect of the worm enabled it to spread without requiring any human interaction (like opening an attachment).

Event: Internet Information Server (IIS) Buffer Overflow
Date: June 1999
A script was posted on the Internet that, if run, would give a hacker control over the most popular version of Microsoft®'s IIS Web server product. Overnight, roughly 1.4 million Web servers worldwide became insecure.

Event: Microsoft® Jet
Date: July 1999
Microsoft® Jet is a widely used backend database for many Microsoft® products (like Office) and many programs written using Microsoft® Visual C++®. A flaw in the software made it possible for a pure data file (containing no scripts) to infect a computer. As a result, a Microsoft® Excel 97 spreadsheet with scripting disabled could still run arbitrary code on a computer. The only solution to the problem was to disable Microsoft® Jet (an action that was unacceptable for many e-commerce sites). Due to the complexity of the bug, a patch was not released for weeks.

Event: Hotmail Hole
Date: August 1999
A hacker group posted a script on the Internet that allowed a person to access any Hotmail e-mail account. Instantly, the e-mail of 50 million Hotmail users was available to anyone who wanted access.

Event: Microsoft® Internet Explorer Bugs
Date: August 1999
Flaws that were discovered in Internet Explorer ActiveX® and Java™ technologies allowed Web sites complete access to a visitor's computer. Although these bugs are part of a continuous stream of such vulnerabilities, these were particularly dangerous. The Java™ bug shows that

despite the soundness of the Java™ security model, implementation difficulties continue to produce vulnerabilities. The ActiveX® bug is more surprising since its model is very simple. With ActiveX®, a user decides whether a signed piece of code should run on a host or not. However, due to the vulnerability, a Web server could run malicious code on a computer without the permission or knowledge of the user. For more information on security issues with active content, see the March 2000 *ITL Bulletin* entitled "Security Implications of Active Content" at http://www.nist.gov/itl/lab/bulletns/mar00.htm.

Event: Bubble Boy
Date: November 1999
A vulnerability was discovered that enabled an e-mail message to infect a computer system when the user merely previewed the e-mail. It used to be a popular myth that reading an e-mail could infect a computer. With new e-mail programs that can read all types of scripts (and that by default do read such scripts), that myth is now a reality.

Event: Traffic Storms on Web Servers
Date: February 2000
A series of very powerful distributed DOS attacks temporarily shut down some of the most prominent e-commerce sites on the Web. The attacks showed that a hacker, with some preparation and patience, could take down even the highest-capacity Web sites on the Internet using publicly available attack tools.

Event: ILOVEYOU Worm
Date: May 2000
E-mail messages were sent with a subject line "ILOVEYOU" and the text of the message encouraged users to execute an attachment to the message. The attachment infected the user's system, sent copies of itself to all addresses in the user's Outlook® e-mail address book, and performed other malicious actions. A year after Melissa, it appears that users worldwide can still be tricked into running malicious attachments. Because of the continued severity of such attacks, Microsoft® has created a patch to prevent such scripts from reading a user's address book.

Trends in Vulnerabilities and Hacking

The above timeline includes a number of new hacker strategies that have caused significant problems. In some cases, these problems may not be entirely preventable. The following sections present an analysis of these strategies, as well as past strategies that continue to cause significant problems.

The Emergence of Worms

In 1988, Robert Morris released a worm on the Internet that spread quickly throughout the Internet and shut down many networks for several days. At that time, though, the Internet was not nearly as important as today, and few non-technical people noticed. For ten years following that event, no widespread worms were seen on the Internet and many thought the "Morris worm" would be an isolated event. However, from the first half of 1999 to the middle of 2000, at least three new worms caused havoc and trouble across the world: Melissa, Explorer.zip, and ILOVEYOU. The method by which these programs propagated to other networks and systems in most cases involved a user clicking on and executing an e-mail attachment that contained a

worm. With deceptive wording, the initial coordinated deployment of e-mail messages enabled the worm to spread worldwide within hours. This gave scant time for anti-viral software vendors to write and disseminate software updates. On some networks, the servers dedicated to anti-viral updates were the subjects of the worm attacks, and users had to wait days before the software could be updated. Once anti-viral software was updated, hackers altered each of these viruses and re-released them in forms that were undetectable using the currently updated virus software. Melissa had at least 40 variants released during its lifetime. To combat such worms and their variants, Microsoft® recently released a patch for Microsoft® Outlook® to prevent worms from sending copies of themselves to e-mail addresses contained in Outlook®'s address book.

The Mixed Blessings of Software Homogeneity

As the world moves towards people and organizations using the same set of software, the vulnerabilities in that software can have a wide and enormous impact on society. For example, the Melissa and the ILOVEYOU worms were able to spread rapidly because of the widespread usage and exploitation of Microsoft® Outlook®. At the same time, protecting against this problem by installing heterogeneous operating systems, e-mail software, and office software eliminates the cost and efficiency benefits of using common applications.

Windows of Insecurity

Even the best security technology, policies, and procedures do not guarantee the security of a system. As seen with the Hotmail bug, a seemingly perfectly secure system can become completely insecure overnight if a clever hacker discovers a bug and posts an exploit on the Internet. People should certainly configure their systems correctly, install all patches, use firewalls, deploy an intrusion detection system, and regularly update their virus checkers. However, these techniques typically only prevent and detect known attack methods. If an attacker uses a new attack while engaging in a legitimate conversation with a system, the attack will go undetected and unstopped by the aforementioned methods.

About 30-40 unique computer attacks are published monthly on the Internet. For a certain amount of time after each attack is published, attackers have free rein to break into networks because administrators have not yet been able to apply a workaround or patch. It often takes incident response organizations hours to release workarounds and days to release patches. When that time is added to the time it takes an administrator to become aware of the problem and apply the patch, attackers have a large window of opportunity with every attack that is published. Since new vulnerabilities are discovered on a daily basis, patient hackers can wait for an applicable vulnerability to be published before launching an attack against their desired target.

Weaknesses in Virus Checkers

A related problem exists with virus checkers. Here, the attacker does not need to wait for a new attack to be released, but can simply create a new virus that won't be detected. The problem is that virus detection software detects only the viruses that have been previously analyzed and added to the software's database. Such software has great difficulty in detecting never-before-seen viruses. A hacker who wishes to penetrate a particular company can write a virus specifically for that organization. By testing the virus beforehand on the handful of popular virus-checking programs, the hacker can guarantee that the virus will enter the organization

undetected. The hacker then sends an innocuous e-mail and the malicious code will likely be executed within the target organization.

Denial-of-Service (DOS) Attacks

During the first months of 2000, hackers launched DOS attacks against a number of organizations' Web sites. The attacks were coordinated floods of legitimate-looking requests for connection to the sites. Often, the attacks were launched from a large set of attacking hosts spread throughout the world. In some cases, the Web sites were shut down for hours or days while administrators determined the originating sites of the attacks and installed filters on routers and firewalls to block connections originating from those sites. These sorts of attacks are difficult or impossible to block completely and force organizations that rely heavily on the availability of their Web sites to monitor traffic continuously and react quickly to any suspicious activity.

Lack of Automated Tracing

The primary method hackers employ to avoid being traced successfully is to log into a series of hosts in different countries or organizations before making an attack. To trace the hacker, the owners of each host in the attacker's chain must be contacted and asked to review their log files (if any exist). If the attacker's chain passed through several foreign countries, tracing is made more difficult. A secondary way hackers avoid being traced is by "lying" about their location. The attacker sends out malicious packets with a random Internet Protocol (IP) source address. To trace the source of the packets, one must manually contact router owners on the physical path taken by the packets and trace backwards along the path taken by the malicious packets. As before, if the malicious packets traverse several foreign countries, tracing becomes difficult.

Blurring between Data and Code

It used to be that some files contained only data while other files contained executable instructions. Today, almost all data files can contain small programs that aid in the presentation or use of the data. These programs or scripts embedded in data serve as an easy way for hackers to penetrate a network; the instructions can perform powerful functions and cause havoc. In many cases, the power provided by the scripts embedded into data is unneeded and unused by the user.

Inside-Out Network Subversion

Many organizations now use firewalls, and hackers have responded by developing new techniques for bypassing these security barriers. In many instances, this was accomplished by tricking inside users and systems to execute code containing worms, which could then spread to other systems behind the firewall. In other cases involving attacks that used JavaScript™ and ActiveX®, users were tricked into executing malicious code hidden in external Web sites. In the case of the Microsoft® Jet Engine incident, simply reading the e-mail that contained the embedded worksheet caused the malicious code to execute; no attachments were involved. There is some consensus among worldwide corporations that these "inside-out" attack scenarios are likely to be the most dangerous because they are difficult to detect and prevent.

Threat Summary

Network perimeter security mechanisms, while necessary and effective in stopping the majority of attacks, cannot provide sufficient protection against all outside threats. Attackers, faced with sophisticated firewalls, have developed mechanisms to bypass those firewalls by directly attacking user computers within the network. A common bypass mechanism is to attack a user through e-mail and Web browsing using a variety of security flaws in commonly used scripting languages. Users are often unaware when a script is being run since scripts can piggyback on most types of data files. Often, but certainly not always, such inside-out attacks rely upon a user performing an action such as opening an attachment. Attackers may create the malicious code themselves to ensure that it will not be detected by an anti-virus tool.

Another way of entering a network is to attack the software and servers that are visible from the Internet. As previously discussed, the most recent attack might be used so that it will not be detected by intrusion detection systems. Attackers frequently target e-mail servers, domain name servers, Web servers, routers, and even computer security devices (like firewalls). If such attacks are detected, it is unlikely that the attacker's identity can be found, as tracing expert hackers on the Internet is very difficult.

Security is often very lax inside a network since systems administrators generally do not have time to completely secure all internal hosts. Thus, worms or human attackers that enter a network via e-mail may spread their influence throughout a network using a variety of possible vulnerabilities. Typically, these other methods of spreading attacks are automated and do not require a legitimate user to be deceived into performing an insecure action.

Despite the severity and sophistication of a computer attack, the attacker may not be a large, well-funded organization. A lone hacker with patience and publicly available tools can cause an enormous amount of damage. A single teenager can marshal the resources to launch DOS attacks against the most robust of Web sites. A more-prepared and better-funded adversary could do much more damage. Organizations, large and small, must prepare against these emerging threats.

Recommendations

To mitigate the threat of hackers on the Internet breaking into or shutting down a network, organizations must divide their attention among several areas:

- securing a small number of externally visible systems,
- hardening a large number of vulnerable internal systems,
- responding to security incidents, and
- mitigating denial-of-service attacks.

A security architecture, policies, procedures, firewalls, virus checkers, intrusion detection systems, strong authentication schemes, virtual private networks, host encryption, personal firewalls for telecommuters, war dialers, and other appropriate security devices must also be in place and appropriately configured to support these activities.

Securing a Small Number of Externally Visible Systems

Due to the widespread use of firewalls, most hackers on the Internet can directly access only a few hosts in an organization. These hosts are usually firewalls, Web servers, routers, e-mail servers, and domain name servers. If the applications on these hosts are vulnerable, a hacker not only has access to a valuable resource, but also the host may provide an avenue by which to

break into the hosts behind the firewall. Thus, it is necessary to secure these hosts and to frequently patch and upgrade to mitigate emerging threats. Fortunately, the number of such important hosts visible from the Internet should be small relative to the total number of hosts in an organization. Therefore, a focused effort on this set of hosts is generally cost-effective.

The most important applications to patch, secure, and monitor include:

1. Domain name system (e.g., BIND)
2. CGI scripts employed by Web servers (be certain to remove vulnerable example scripts)
3. Web server vulnerabilities (e.g., Apache and Microsoft® IIS)
4. E-mail server software (e.g., Sendmail)
5. Operating system software
6. E-mail access protocols/daemons (e.g., IMAP and POP)
7. SNMP access control to networking devices

The SANS (System Administration, Networking, and Security) Institute has published a list of the top ten vulnerabilities which covers many of these "problem" applications. The paper is available at: http://www.sans.org/topten.htm. Also, NIST maintains a searchable index of serious vulnerabilities that contains over 600 entries. Called the ICAT Metabase, this index is a tool that allows one to search for vulnerabilities at a fine granularity (e.g., using software names and version numbers). For each vulnerability of interest, ICAT points a user to patch information and vulnerability databases that thoroughly describe the security issue. The ICAT Metabase is available at: http://nvd.nist.gov/.

(Webmaster's NOTE: The ICAT Metabase has changed name to the National Vulnerability Database (NVD) after this Bulletin was written). So anytime when ICAT Metabase is mentioned in this Bulletin, please refer to the National Vulnerability Database (NVD) and URL is provided in last sentence in paragraph above.

Hardening a Large Number of Internal Systems

Securing internal hosts in an organization is typically much harder because of scaling issues. Most organizations have a large number of insecure hosts sitting behind their firewall. In the near future, more vendors will provide automated ways to patch a large set of hosts from a single console. This technology will enable organizations, which have a standard host setup, to easily keep all hosts updated. However, this technology is not widespread and most system administrators have to patch hosts one computer at a time. Since the time required to install a patch on all hosts usually is prohibitively large, internal systems are not usually patched.

Despite this frustrating situation, there are ways to inexpensively harden internal systems against hackers on the Internet. The key is to realize that internal systems are typically penetrated through e-mail and Web access since the firewall, when properly configured and maintained, prevents most other types of access. We recommend the following actions:

1. Users of Microsoft® Windows should be trained in how to install security patches using the Windows Update feature: http://windowsupdate.microsoft.com. Active desktop users should be notified about when to accept the automatic notifications of security updates.

2. Users should be trained not to open attachments if an e-mail looks atypical (even e-mail from their friends). A reasonable rule is that a user should not open an attachment, without confirming with the sender, unless the context of the e-mail demonstrates that this is not a mass e-mailed virus.
3. Virus checkers should be installed on every computer and those checkers should automatically update themselves daily with new virus signatures.
4. Organizations should check for viruses at their firewall and e-mail server in addition to checking on each internal host. We recommend using a different virus detection product on internal hosts and backbone hosts in order to diversify, and thus strengthen, a network's detection and prevention capability.
5. Organizations should create an internal Web site for distributing virus software updates and patches for situations where vendors' Web sites are overwhelmed with update requests.
6. Scripts should be disabled when people preview and read their e-mail. Otherwise, as soon as a new script vulnerability emerges, hackers can send malicious e-mail that will automatically infect the receiver. It may be best to enforce this policy at the e-mail gateway where the scripts can be automatically removed or e-mail containing scripts can be automatically rejected.
7. For organizations requiring greater internal host security, an easy way to boost security is by installing internal firewalls to isolate critical subnets and by using personal firewalls on critical internal hosts.

Responding to Security Incidents

Despite our best efforts to secure systems, hackers will occasionally penetrate an organization. The response to such break-ins must be planned, timely, and appropriate in order to mitigate the damage. System administration staff must be trained concerning what to do or who to call during a security crisis. Incident response organizations are useful resources for advising an organization about recovering from an attack and setting up their own incident response capability (FedCIRC, www.fedcirc.gov [for civilian government] or CERT, www.cert.org).

Mitigating Denial-of-Service Attacks

There are two types of DOS attacks: flaw-based and flooding. Both attacks attempt to consume the resources of a host or application to prevent it from functioning. Some articles talk about "distributed DOS" attacks. These attacks are DOS attacks that are generated from multiple attacking hosts. Attackers use these multiple hosts in order to amplify the effect of their attacks.

Flaw-based DOS attacks make use of errors in software in order to consume resources. Patching and upgrading software can prevent these types of DOS attacks. Flooding DOS attacks send more information to an application than it can handle. These types of attacks cannot be prevented by software fixes because the software is functioning properly.

Several ways exist, however, to combat a flooding DOS attack. A simple solution is to install faster hardware. With this solution, one attempts to handle normal traffic in addition to the load caused by the attack; this can be effective against hackers with limited resources. Another solution is to attempt to filter out the attack packets before they reach the target software. Attackers are not always clever and may attack from the same IP address, use packets with the same contents, or use a recognizable pattern in port number choices. These features may help a

target distinguish attack packets from legitimate traffic. Once a distinguishing feature has been identified, routers can be configured to drop the malicious packets. This approach often works; however, a clever attacker with many resources can circumvent any such countermeasures.

Many organizations are concerned not only about being the target of a denial-of-service attack, but also they do not wish to be the unwitting source (or intermediary) of such an attack. A SANS paper, located at http://www.sans.org/ddos_roadmap.htm, describes how to reduce the possibility that an organization will be used by a hacker as the source of such attacks.

Glossary

BIND: BIND (Berkeley Internet Name Domain) is a set of programs used to implement a DNS server. For more information, see: http://www.isc.org/products/BIND/.

Domain name system: The domain name system (DNS) translates human-readable computer names, like http://csrc.nist.gov, to the numeric IP addresses used by computers. For more information, see: http://www.whatis.com/dns.htm.

Firewall: A firewall is a security device that separates two or more networks. It contains fine-grained rules on what types of traffic may pass between the networks and it often can analyze that traffic for known vulnerabilities. Any traffic that does not pass the firewall rules is thrown away and will not pass from one network to another. For more information, see: http://www.whatis.com/firewall.htm.

IMAP: The Internet Message Access Protocol (IMAP) is a popular protocol used to retrieve e-mail from an e-mail server. For more information, see: http://www.whatis.com/imap.htm.

IP: The Internet Protocol (IP) is the primary communication protocol used on the Internet. For more information, see: http://www.whatis.com/ip.htm.

Patch: A patch is a small program, typically released by a software vendor for free, that fixes bugs in that vendor's already-released software. For more information, see: http://www.whatis.com/patch.htm.

Personal firewall: A personal firewall is a scaled-down firewall designed for use on personal computers.

POP: The post office protocol (POP) is a popular protocol used to retrieve e-mail from an e-mail server. For more information, see: http://www.whatis.com/pop3.htm.

SNMP: The simple network management protocol (SNMP) is a common communication protocol used to control network devices. For more information, see: http://www.whatis.com/snmp.htm.

Virtual private network: A virtual private network (VPN) is an encrypted tunnel between two organizations (or hosts) that enable secured communication to occur over public networks. This

tunnel allows a variety of different types of traffic, which distinguishes a VPN from an encrypted connection. For more information, see: http://www.whatis.com/vpn.htm.

Worm: A worm is a computer virus that attempts to move itself to new computers. Most viruses do not attempt to move themselves but instead rely on humans to unintentionally move them among computers.

For More Information

NIST has issued publications on various aspects of network and computer security, and maintains a Web site, http://csrc.nist.gov, where these publications and other information are available. Of particular interest for this subject matter are recent ITL bulletins on:

- Security Implications of Active Content, March 2000
- Acquiring and Deploying Intrusion Detection Systems, November 1999
- Securing Web Servers, September 1999 and
- Computer Attacks: What They Are and How to Defend Against Them, May 1999

These bulletins can be found at: http://www.nist.gov/itl/lab/bulletns/cslbull1.htm.

Disclaimer: Any mention of commercial products or reference to commercial organizations is for information only; it does not imply recommendation or endorsement by the National Institute of Standards and Technology nor does it imply that the products mentioned are necessarily the best available for the purpose.