The threats to our products

April 1, 1999 — By Loren Kohnfelder and Praerit Garg

The growing use of computer systems to store data critical to businesses, as well as users' personal data, makes them very attractive targets for security attacks. Successful attacks can lead to loss of privacy, disclosure of sensitive data, and disruption or denial of service—losses that can cost millions of dollars. The Microsoft Security Task Force has defined a security threat model that it recommends all Microsoft product teams adopt to secure our products for our customers.

The **S.T.R.I.D.E.** security threat model should be used by all MS products to identify various types of threats the product is susceptible to during the design phase. Identifying the threats is the first step in a proactive security analysis process. Threats are identified based on the design of the product. The next steps in the process are identifying the vulnerabilities in the implementation and then taking measures to close security gaps.

S.T.R.I.D.E. stands for:

- **Spoofing of user identity**
- **Tampering with data**
- **Repudiability**
- **Information disclosure (privacy breach)**
- **Denial of Service (D.o.S.)**
- **Elevation of privilege**

Some attacks can be very sophisticated and have several steps. In such attacks, one minor break-in leads to another, and eventually substantial system damage is done. In most such cases, one of the links is the weakest, and the security of the entire system typically is no better than its weakest link. Finding and improving such weak links is how threat analysis helps improve the security of our products and services.

This article describes various threat categories in the S.T.R.I.D.E. model and provides examples of vulnerabilities that may be exploited by various kinds of attacks to make a threat a reality. This is intended to help you identify potential vulnerabilities in your product during a security analysis.

Each threat is discussed in the context of Microsoft products, which fall into the following five categories:

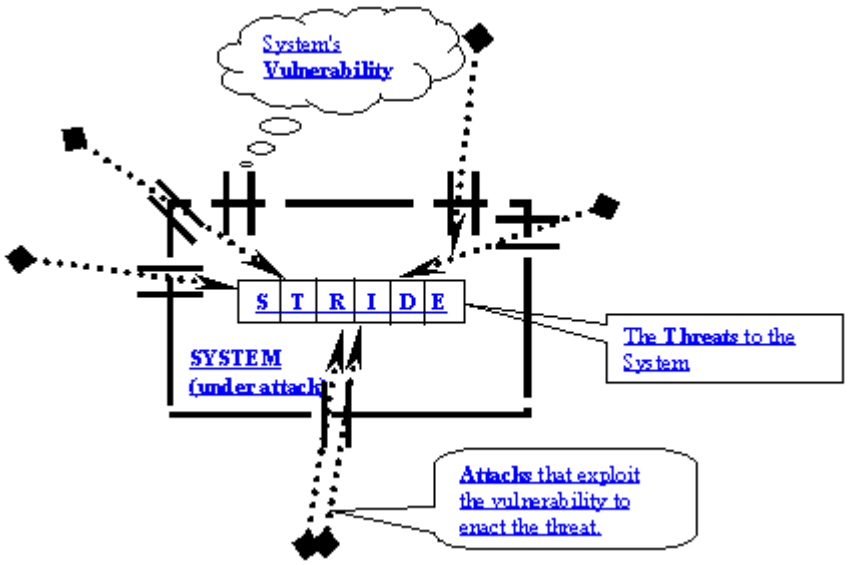
- **Server operating systems** Windows NT/2000 Server
- **Client operating systems** Windows NT/2000 Workstation, Win9x, WinCE, Internet Explorer
- **Client/server applications** Exchange, SQL, etc.
- **Desktop applications** Office, etc.
- **Web and media applications** WebEssentials, portal Web sites, etc.

But first, let's define some important terms that we'll use throughout this piece and that have very precise meaning in security discussions.

- **Threat** Any potential occurrence, malicious or otherwise, that can have an undesirable effect on the system resources (files, registry keys, data-on-wire, etc.). Undesirable effects can be a system crash, the ability to read a sensitive file or modify a registry key, and so forth.

- **Vulnerability** Some unfortunate characteristic that makes it possible for a threat to occur. Examples include bad security on a file, buffer overflows, and (in a server product running on Windows NT) missing client impersonation calls when servicing client requests.

- **Attack** An action taken by a malicious intruder to exploit certain vulnerabilities to enact the threat. Examples of attacks include steps taken by a non-administrator to acquire administrator privileges and a technique that allows private data to be leaked.



Three aspects of system security

The S.T.R.I.D.E. model

Security threats fall into the six major categories listed below. In addition to describing each general threat and the kind of software products or services it typically applies to, we offer a few examples to convey the varying character of the threat. Some of the examples may be specific to certain products or technologies; it is important to understand the threats themselves but not necessarily all of the examples given.

Spoofing of user identity

What's the threat? Breaching the user's authentication information. In this case, the hacker has obtained the user's personal information or something that enables him to

replay the authentication procedure. Spoofing threats are associated with a wily hacker being able to impersonate a valid system user or resource to get access to the system and thereby compromise system security.

What do these threats have in common?

- Ability to change the identity associated with an object.
- Subversion of secure logon mechanism.
- Successful use of false credentials.

Examples

- A malicious impersonator (man-in-the-middle) spoofs IP (Internet Provider) packets to hijack a connection to the server. The vulnerability here is that the communication protocol does not incorporate confidentiality and integrity.
- Authentication protocols that use passwords without encrypting them disclose credential information to an eavesdropper, who can then use this information to impersonate the user. The vulnerability here is the credential information not being properly encrypted.
- The "Trojan horse" attack is the classic spoof. For example, on a browser, a Web page might manage to construct an exact visual duplicate of the system log on and trick users into typing their name and password, not suspecting they were actually giving the information to a Web site.
- Replay where an eavesdropper can replay a client/server exchange to the server, such as a debit transaction on a bank account. The vulnerability here is missing sequence detection.
- Forging e-mail. The vulnerability in this case is lack of confidentiality and integrity in email messages.
- DNS poisoning. The vulnerability here is ability to do untrusted updates to the DNS database.

What products are susceptible? All types of software products may be subject to these threats.

Tampering with data

What's the threat? Modifying system or user data with or without detection. An unauthorized change to stored or in-transit information, formatting of a hard disk, a malicious intruder introducing an undetectable network packet in a communication, and making an undetectable change to a sensitive file are all tampering threats.

What do these threats have in common?

- Modification of data that should not be accessible.
- Causing a trusted entity to modify data improperly.
- Elevation of *privilege* can enable tampering

Examples

- Packet injection attacks where data on the wire is modified. The vulnerability that exposes this threat is a lack of integrity on data sent on the wire.

- Modification of file data without authorization checks. The vulnerability that exposes this threat is missing access checks, buffer overflows, no integrity checks, and so on.

- Data corruption due to execution of erroneous code. Vulnerabilities include unhandled memory allocation failures, uninitialized memory, use of freed memory resources, and miscalculations like divide by zero.

- Data corruption or modification by Trojans and viruses. The vulnerability is the software's susceptibility to Trojans.

What products are susceptible? All kinds of software products are susceptible to data tampering threats and therefore should address them.

Repudiability

What's the threat? An untrusted user performing an illegal operation without the ability to be traced. Repudiability threats are associated with users (malicious or otherwise) who can deny a wrongdoing without any way to prove otherwise.

What do these threats have in common?

- Way to avoid logging of important security event.

- *Spoofing* can be used to conceal the identity of the agent performing an action.

- *Tampering* with security log can result in repudiability.

Examples

- Undetected attempts to break into a user account by the attacker. Lack of failed logon audits is the vulnerability.

- Deletion of sensitive files inadvertently or maliciously by a user. Lack of successful auditing of object access is the vulnerability.

- Ability of a malicious user to deny sending a message. Lack of message signatures and signature verification before accepting the message is the vulnerability.

What products are susceptible? All software products, with the possible exception of desktop applications, are susceptible to such threats. Desktop applications typically depend on the underlying operating system to handle non-repudiability requirements of the environment. This mostly includes the ability to trace "who-done-it" for unauthorized data modifications.

Information disclosure (privacy breach)

What's the threat? Compromising the user's private or business-critical information. Information disclosure threats expose information to individuals who are not supposed to see it. A user's ability to read a file that she or he was not granted access to, as well as an intruder's ability to read the data while in transit between two computers, are both disclosure threats. Note that this threat differs from a spoofing threat in that here the perpetrator gets access to the information *directly* rather than by having to spoof a legitimate user.

What do these threats have in common?

- Access to data that is considered private and should be protected.
- Sniffing data in a network or that has been left inadvertently in storage.
- Protocols or interfaces that improperly reveal user identity, location, passwords, and so on.
- *Spoofing* or *elevation of privilege* can enable an attacker to access private data.

Examples

- A data leak due to buffer overflow attacks. Sophisticated attacks where a handcrafted call stack is placed on a vulnerable system call (a call to the operating system or a privileged server) can cause privileged code to return information, such as kernel memory dump, back to the unauthorized user. The vulnerability here is buffer overflow in the system service.
- Data snooping due to man-in-the-middle attacks, as well as simple attacks where packets that have not been encrypted are sniffed. Also, sophisticated attacks where a flawed authentication protocol enables an eavesdropper to compute or break the session key so that the eavesdropper can decrypt all encrypted and signed data. The vulnerability for all three of these examples is security flaws in the network protocol.
- Getting data without authorization. Servers that miss impersonating the client or that return data without performing access checks (even if they do impersonations) are examples. The vulnerabilities include missed impersonation (i.e., client gets access to anything server has access to) or missed access checks.
- Obtaining data by exposing common coding errors, such as memory leaks.
- Improper handling of reused object. Data leaks can result when a file system allocates the same blocks to a new file that were previously held by another file and returns data from those blocks without upper or lower watermark checks or without clearing the blocks before reallocation.
- Win9x PWL (password log) files can be used to reveal a user's credential information, leading to other sophisticated attacks.
- Physical access to a hard disk leading to unauthorized data access.
- When a client accesses data from multiple locations, the compartmentalizing of mishandled information can cause information from one location to become available to another.
- Office macros can be used to leak data. These fall into the general class of Trojan vulnerabilities.

What products are susceptible? All software products can be vulnerable to information disclosure threats. Therefore, a proactive security review process for every Microsoft product must outline various information disclosure threats and how they will be addressed.

Denial of Service (D.o.S.)

What's the threat? Making the system temporarily unavailable or unusable, such as those attacks that could force a reboot or restart of the user's machine. When an attacker can temporarily make the system resources (processing time, storage, etc.) unavailable or unusable, we have a denial of service threat. We must protect against certain types of D.o.S. threats for improved system availability and reliability. However, some types of D.o.S. threats are very hard to protect against, so at a minimum, we must identify and rationalize such threats.

What do these threats have in common?

- Processing: consumption of CPU cycles by infinite or very long programmatic looping.
- Storage: large allocation of memory or file quota that blocks legitimate use of the same.
- Excessive and unwanted use of screen space, printer paper, and so forth.
- Causing a crash or error mechanism that interferes with normal usage or that requires restarting.
- *Elevation of privilege* can exacerbate D.o.S. by gaining larger resource quotas.

Examples

- SYN attacks and packet bombs that use various network protocol vulnerabilities to cause servers to crash.
- Sophisticated buffer overflow problems, such as parameters with no length, can cause the server to chase a nonexistent memory location and crash. Similarly, GetAdmin-style handcrafted stacks can cause privileged instructions to shut down the system.
- Common coding errors, such as unhandled memory allocation failures (referencing an invalid pointer), uninitialized memory (bad data used), use of freed memory and resources (referencing invalid memory), and miscalculations (divide by zero), can cause exceptions that would crash the software.
- Weak policies (inherent in design or due to misconfiguration), such as a process taking up all CPU time.
- Trojans, such as viruses, can also cause the software to become unusable.

What products are susceptible? All software products are susceptible to denial of service threats. Microsoft product groups should address them in the proactive security process by identifying various vulnerabilities that can result in denial of service. While D.o.S. is one of the hardest security threats to address, and in many cases it is reasonable not to address them, your team should identify and rationalize all such cases.

Elevation of privilege

What's the threat? An unprivileged user gains privileged access and thereby has sufficient access to completely compromise or destroy the entire system. The more dangerous aspect of such threats is compromising the system in undetectable ways

whereby the user is able to take advantage of the privileges without the knowledge of system administrators. Elevation of privilege threats include those situations where an attacker is allowed more privilege than should properly be granted, completely compromising the security of the entire system and causing extreme system damage. Here the attacker has effectively penetrated all system defenses and become part of the trusted system itself and can do anything.

What do these threats have in common?

- Improperly gaining unrestricted rights (becoming a "administrator").
- Running untrusted data as native code in a trusted process, such as by buffer overrun.
- *Spoofing* identity to gain access to resources not otherwise available.

Examples

- Buffer overruns, such as handcrafted stacks in a GetAdmin attack, causing user code to be executed at an elevated privilege and thereby compromising the entire operating system's trusted computing base.
- The ability to run executables without the (privileged) user's consent can allow the perpetrator to perform privileged operations, such as making himself or herself a privileged user.
- Rogue OCX/ActiveX control with malicious code.
- Missing impersonation in the server, or client-side impersonation such as the one leveraged by SecHole.exe, causes the server to do privileged operations on behalf of an unauthorized user, thereby effectively raising the privilege level of the malicious user.
- Missing or improper access checks in the security subsystem itself can result in privilege elevation. For example, if group membership of administrators was updated without an access check, it would allow an unauthorized user to become a system administrator.

What products are susceptible? All server products (operating systems, server applications, content and media services) are susceptible to privilege elevation threats. Because client systems are assumed to run in the context of the unprivileged user, they should not be trusted to not misuse the user's capabilities. This becomes even more important when a privileged user such as an administrator uses the client software. For this reason, privileged users are expected to run only "trusted clients."

Client operating systems, such as Windows 2000 Professional, are subject to the same threats as the server operating system. Because they do not have the concept of privileged vs. unprivileged mode, operating systems such as Win9x (which runs only in privileged mode) cannot be associated with such threats.

Because desktop applications typically depend on underlying operating systems to handle privileged vs. unprivileged user distinctions, privilege elevation threats are not applicable to desktop applications. But applications can create vulnerabilities that cause such attacks to be launched against the underlying operating system. Examples include supporting Trojans (such as Office macros) and disobeying the "least privileged" rule, where applications open files for more access than is necessary, thereby causing the system to be configured with lax security.

Beyond the basic threats

Several other threats cannot be completely addressed in software, yet still require proper policies and procedures to be in place. Software can help *raise the bar*, however, and protect against some of these threats. They are valid customer security issues, so Microsoft products should consider the following when doing security analysis (during product development) and identify which are not addressed with clear rationales.

- **Privilege misuse** is one of the very common attacks we have seen associated with various Microsoft products. It happens when a user with administrative access does things that violate security procedures, such as browsing an untrusted Web site or editing a document from an untrusted source. This problem is very common with PC systems that are often used without formal logon, so in essence every user has full administrative rights.

- **Rogue administrator** is another variation of privilege misuse attack. With all our products, an administrator typically has godlike powers, and when the administrator turns rogue, there is very little we can do to protect the system from utter compromise and destruction. However, more often the attack is unintentional because the administrator is doing non-administrative operations like browsing the Web with administrative privileges.

- **Trust abuse** is another attack in this category. Trust abuse is the kind of attack where a software product from a trusted source, such as Microsoft, intentionally or unintentionally violates the user's privacy requirements. An example of trust abuse is if a piece of code sends a user's profile information back to the vendor without the user's confirmation.

Technologies like Restricted Tokens and RunAs in Windows 2000 attempt to address these issues at different levels, but a security-conscious customer must still play a part by setting proper policies and procedures to prevent non-administrative operations, such as browsing the Internet, when logged on using accounts with administrative privileges. Because of the potential impact of such issues, each of our products should clearly identify during proactive security reviews whether this is addressed or not.

- **Multiple users on Win9x** is also a system where one user's data is not protected from another user. This is a fundamental limitation of the platform.

- **Intellectual property violation** is another security threat that is becoming significant and will need to be addressed soon. Though it is not necessary for products to handle it, it is necessary for them not to violate it intentionally. The Security Task Force recommends that a proactive security review process for Microsoft products explicitly call this out as a customer consideration and provide a rationale.

- **Physical security** is typically assumed in most software security systems. But because physical security requirements are becoming hard to meet in today's Internet-based, connected world, meeting physical security threats is becoming important even though it is not typically necessary, so product review should explicitly make recommendations even if the product doesn't handle it.