ELSEVIER

Computers & Security

www.elsevier.com/locate/cose

A taxonomy of network and computer attacks

Simon Hansman, Ray Hunt*

Department of Computer Science and Software Engineering, University of Canterbury, New Zealand

Received 3 February 2004; revised 8 June 2004; accepted 18 June 2004

KEYWORDS
Taxonomy;
Computer attack;
Network attack;
Classification scheme;
Attack vector;
Attack target;
CERT

Abstract Attacks over the years have become both increasingly numerous and sophisticated. This paper focuses on the provisioning of a method for the analysis and categorisation of both computer and network attacks, thus providing assistance in combating new attacks, improving computer and network security as well as providing consistency in language when describing attacks. Such a taxonomy is designed to be useful to information bodies such as CERTs (Computer Emergency Response Teams) who have to handle and categorise an every increasing number of attacks on a daily basis. Information bodies could use the taxonomy to communicate more effectively as the taxonomy would provide a common classification scheme. The proposed taxonomy consists of four dimensions which provide a holistic taxonomy in order to deal with inherent problems in the computer and network attack field. The first dimension covers the attack vector and the main behaviour of the attack. The second dimension allows for classification of the attack targets. Vulnerabilities are classified in the third dimension and payloads in the fourth. Finally, to demonstrate the usefulness of this taxonomy, a case study applies the taxonomy to a number of well known attacks.

Introduction

Network and computer attacks have become pervasive in today's world. Any computer connected to the Internet is under threat from viruses, worms and attacks from hackers. Home users, as well as business users, are attacked on a regular basis. Thus the need to combat computer and network attacks is becoming increasingly important.

Since 1999 there has been a marked increase in the number of incidents[1] reported as statistics from the Computer Emergency Response Team

* Corresponding author. Tel.: +64 336 423 47; fax: +64 336 425 69.
E-mail address: ray@cosc.canterbury.ac.nz (R. Hunt).

[1] An incident is an attempt at violating security policy, such as attacking a computer or attempting to gain unauthorised access to some data.

0167-4048/$ - see front matter © 2004 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cose.2004.06.011

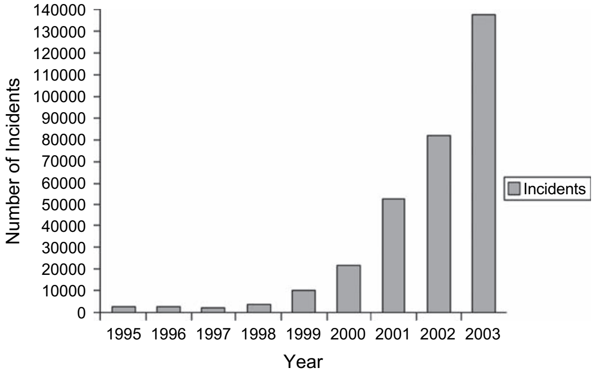


Figure 1 Incidents over the past nine years.

Coordination Center (CERT/CC) (CERT, 2003) show. Fig. 1 shows graphically the number of incidents as reported by CERT/CC over the past nine years with an alarming rise to 137,500 in 2003.

Not only has there been a marked increase in the number of attacks, but the sophistication and complexity has also increased. Thus many attacks are now relatively ''user-friendly'' and in-depth technical knowledge is no longer required to launch an attack. This has led to the rise of various groups of attackers, such as ''script-kiddies'', who while ignorant of how their attack works, can cause great damage. In Lipson (2002), this trend is represented graphically as shown in Fig. 2.

The purpose of a classification or taxonomy is to provide a useful and consistent means of classifying attacks. Currently attacks are often described differently by different organisations, resulting in confusion as to what a particular attack actually is. For example, one organisation may classify an attack as a virus while another classifies it as a worm. The proposed taxonomy (section ''Proposal for a new prototype taxonomy'') is an attempt to provide a common classification scheme that can be shared between organisations.

A taxonomy also allows for previous knowledge to be applied to new attacks as well as providing a structured way to view such attacks. The proposed taxonomy aims to create categories that enable this to occur easily so that similarities between attacks can be highlighted and used to combat new attacks.

Another of the proposed taxonomy's goals is to provide a holistic approach to classifying attacks, so that all parts of the attacks are taken into account, but at the same time the taxonomy is specific. Such a taxonomy has not been suggested before, as previous taxonomies either focus on one part of the attack, and/or classify in general terms. That is, the proposed taxonomy aims to take into account all parts of the attack (from the vulnerability, to the target, to the attack itself) and talk in terms of the target being, for example, MS Windows XP Home with Service Pack 1. Previous taxonomies and requirements for the proposed taxonomy are discussed in detail in the next section.

Requirements and existing classification methods

Requirements of a taxonomy

Before examining existing taxonomies and developing new ideas and methods, it is important to

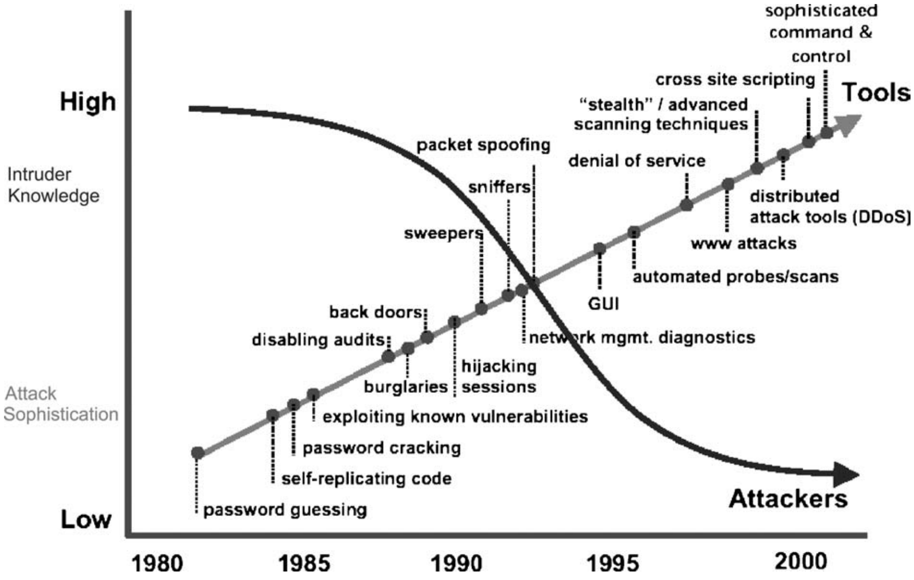


Figure 2 Attack sophistication vs. intruder technical knowledge.

define what a good taxonomy consists of. A number of requirements have been compiled from various sources in Lough (2001) and are listed below:

Accepted (Amoroso, 1994; Howard, 1997): The taxonomy should be structured so that it can become generally approved.

Comprehensible (Lindqvist and Jonsson, 1997): A comprehensible taxonomy will be able to be understood by those who are in the security field, as well as those who only have an interest in it.

Completeness (Amoroso, 1994)/*Exhaustive* (Howard, 1997; Lindqvist and Jonsson, 1997): For a taxonomy to be complete/exhaustive, it should account for all possible attacks and provide categories accordingly. While it is hard to prove a taxonomy that is complete or exhaustive, it can be justified through the successful categorisation of actual attacks.

Determinism (Krsul, 1998): The procedure of classifying must be clearly defined.

Mutually exclusive (Howard, 1997; Lindqvist and Jonsson, 1997): A mutually exclusive taxonomy will categorise each attack into, at most, one category.

Repeatable (Howard, 1997; Krsul, 1998): Classifications should be repeatable.

Terminology complying with established security terminology (Lindqvist and Jonsson, 1997): Existing terminology should be used in the taxonomy so as to avoid confusion and to build on previous knowledge.

Terms well defined (Bishop, 1999): There should be no confusion as to what a term means.

Unambiguous (Howard, 1997; Lindqvist and Jonsson, 1997): Each category of the taxonomy must be clearly defined so that there is no ambiguity with respect to an attack's classification.

Useful (Howard, 1997; Lindqvist and Jonsson, 1997): A useful taxonomy will be able to be used in the security industry and particularly by incident response teams.

Depending on the goals, a taxonomy may not necessarily meet all the requirements identified above. All are useful properties for a taxonomy, but not all are necessary. For example, not all taxonomies strive to be mutually exclusive. The goal for the proposed taxonomy is to adhere to all of the above requirements.

Existing taxonomies and previous work

The field of network and computer security has seen a number of taxonomies aimed at classifying security threats, such as computer and network attacks and vulnerabilities. In the following section some of the more prominent taxonomies will be examined. Some taxonomies are too trivial to include. For example Symantec (http://securityresponse.symantec.com/avcenter/vinfodb.html) categorises virus attacks by name in 26 groups (A through Z!)

Early security taxonomies

The two most important early taxonomies in the security field were the Protection Analysis (PA) (Bisbey and Hollingworth, 1978) taxonomy and the Research in Secured Operating Systems (RISOS) (Abbott et al., 1976). While these focus on vulnerabilities rather than attacks, they provide a good background to proposing new taxonomies. Both focused on categorising security flaws and both resulted in similar classification schemes. Each consisted of a number of classes that are roughly equivalent. As Bishop and Bailey (1996) points out, both taxonomies suffer from ambiguity between the classes. Some vulnerabilities may fall across multiple classes and therefore the taxonomies will not be mutually exclusive. However, the concepts from these early taxonomies are valuable, and have been used in newer taxonomies (Lough, 2001; Bishop, 1995; Aslam, 1995). Comparisons of the two taxonomies can be found in Bishop (1995), Bishop and Bailey (1996) and Lough (2001).

Bishop's vulnerability taxonomy

Bishop has made several important contributions to the field of security taxonomies. In Bishop (1995), he presents a taxonomy of Unix vulnerabilities in which the underlying flaws or vulnerabilities are used to create a classification scheme. Six ''axes'' are used to classify the vulnerabilities, viz.:

- *Nature*: the nature of the flaw is described using the Protection Analysis categories
- *Time of introduction*: when the vulnerability was introduced
- *Exploitation domain*: what is gained through the exploitation
- *Effect domain*: what can be affected by the vulnerability

- *Minimum number*: the minimum number of components necessary to exploit the vulnerability
- *Source*: the source of identification of the vulnerability

Bishop's approach is interesting, as instead of a flat or tree-like taxonomy, he uses axes and in our proposed taxonomy (section "Proposal for a new prototype taxonomy") a similar structure is used although with different axes variables. Bishop and Bailey (1996) also performed a critical analysis of other vulnerability taxonomies. Previous taxonomies such as PA, RISOS and Aslam's taxonomy (Aslam, 1995) are assessed and compared. He also examines the issues surrounding taxonomies and especially what makes a good taxonomy. Bishop suggests that one of the main benefits of a taxonomy is that it should assist in the decision on resource investment.

Howard's taxonomy

Howard (1997) presents a taxonomy of computer and network attacks. The approach taken is broad and process-based, taking into account factors such as attacker motivation and objectives.

The taxonomy (Fig. 3) consists of five stages: attackers, tools, access, results and objectives. The attackers consist of a range of types of people who may launch an attack. These range from hackers to terrorists. Tools are the means that the attackers use to gain access. Access is gained through either an implementation, design or configuration vulnerability. Once access is gained, the results may be achieved such as corruption or disclosure of information. From this process the attacker achieves their objectives which may vary from inflicting damage, to gaining status.

In our proposed taxonomy (section "Proposal for a new prototype taxonomy"), the tools used by Howard's taxonomy are roughly analogous. However, ours is focused solely on the attacks, rather than the attack process.

Howard attempts to focus attention on a process-driven taxonomy, rather than a classification scheme. This means the whole attack process is considered, which is certainly valuable. However, as Lough (2001) points out, Howard fails to meet one of his taxonomy requirements: mutual exclusion. Some of the categories shown in Fig. 3 may overlap. For example the attacker's category contains classes that may not be mutually exclusive. As Lough points out: "Depending on one's point of view, a *terrorist's* actions could be indistinguishable from those of a *vandal*. A *spy* could be a *professional criminal*."

Howard's approach is still useful in gaining insight into the process of attacks. However, for information bodies such as CERT, such a taxonomy may not be of much practical value. Information bodies are more concerned with the attack itself, than with the motivations and objectives behind it.

Some of Howard's ideas have been applied in our proposed taxonomy, notably in the third and fourth dimensions (sections "The third dimension" and "The fourth dimension"). Howard and Longstaff (1998) extends his work further by refining some of the stages. However, the problems mentioned above still exist even with the refined taxonomy.

Lough's taxonomy

In 2001, Lough proposed another taxonomy called VERDICT (Validation Exposure Randomness Deallocation Improper Conditions Taxonomy) and is based upon the characteristics of attacks. Instead of a tree-like taxonomy, Lough proposed using four characteristics of attacks:

- *Improper validation*: insufficient or incorrect validation results in unauthorised access to information or a system
- *Improper exposure*: a system or information is improperly exposed to attack
- *Improper randomness*: insufficient randomness results in exposure to attack
- *Improper deallocation*: information is not properly deleted after use and thus can be vulnerable to attack

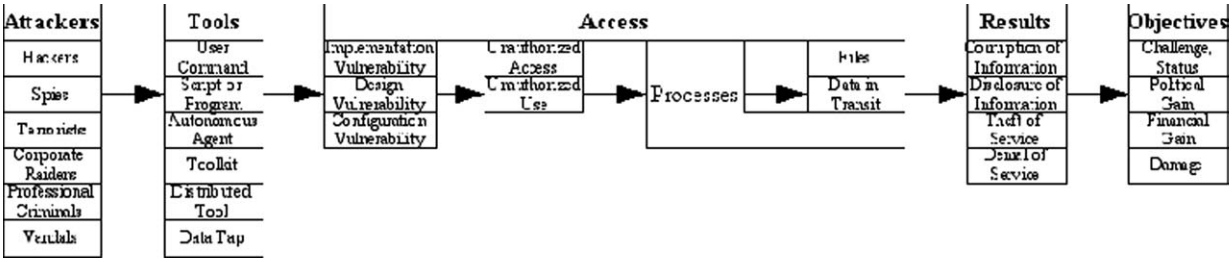


Figure 3 Howard's process-based taxonomy.

Lough proposes that any attack can be classified using these four characteristics. By basing the taxonomy on these characteristics, the taxonomy can easily and tidily classify blended attacks. Lough's approach is similar to both Bishop's axes and to our proposed taxonomy's dimensions. There are, however, a few shortcomings to Lough's taxonomy. While it is useful for applying to a new technology (Lough applies it to IEEE 802.11 and finds numerous vulnerabilities) to discover new vulnerabilities and to classify existing ones, it may be helpful to have a more specific taxonomy.

In terms of an information body such as CERT, Lough's taxonomy may not be useful for the day-to-day task of identifying and classifying new attacks, and issuing advisories. Lough's taxonomy is general, and does not talk about attacks in terms of worms, viruses, and trojans, which is how attacks are usually described in practice.

In the end, the goals of the taxonomy determine its usefulness. Our proposed taxonomy aims to be a practical, specific taxonomy that can be used by information bodies to classify new attacks. Lough's taxonomy on the other hand, succeeds in providing a taxonomy that is useful for analysis and for the prediction of new attacks.

OASIS web application security technical committee

The OASIS Web Application Security Technical Committee (OASIS WAS TC) (OASIS, 2003a) is a current attempt to provide a classification scheme for web application vulnerabilities. Currently it is being developed and is in the early stages of being drafted. OASIS WAS TC is leaning toward using attack vectors as the first step of classification, in a similar way to what is suggested in our proposed taxonomy. XML is being used to describe vulnerabilities so that interoperability is enhanced.

It will be interesting to see how the OASIS WAS TC progresses over the next few years. While still in its early stages, it has produced some good ideas and there is active discussion on the committee's mailing lists (OASIS, 2003b).

Proposal for a new prototype taxonomy

Alternative strategies for a taxonomy design

While the taxonomies discussed in the previous section are useful, they tend to be general in their approach to classifying attacks. Taxonomies such as Howard's (section "Howard's taxonomy") provide a good overview of the attack process, but avoid examining the categories of attacks that face computers and networks each day. For example, classifying attacks such as the Code Red worm would be hard to do using Howard's taxonomy. Therefore, there is a need for a taxonomy that allows for specific kinds of computer and network attacks, such as worms, viruses and buffer overflows. The goal is to provide a pragmatic taxonomy that is useful to those dealing with attacks on a regular basis.

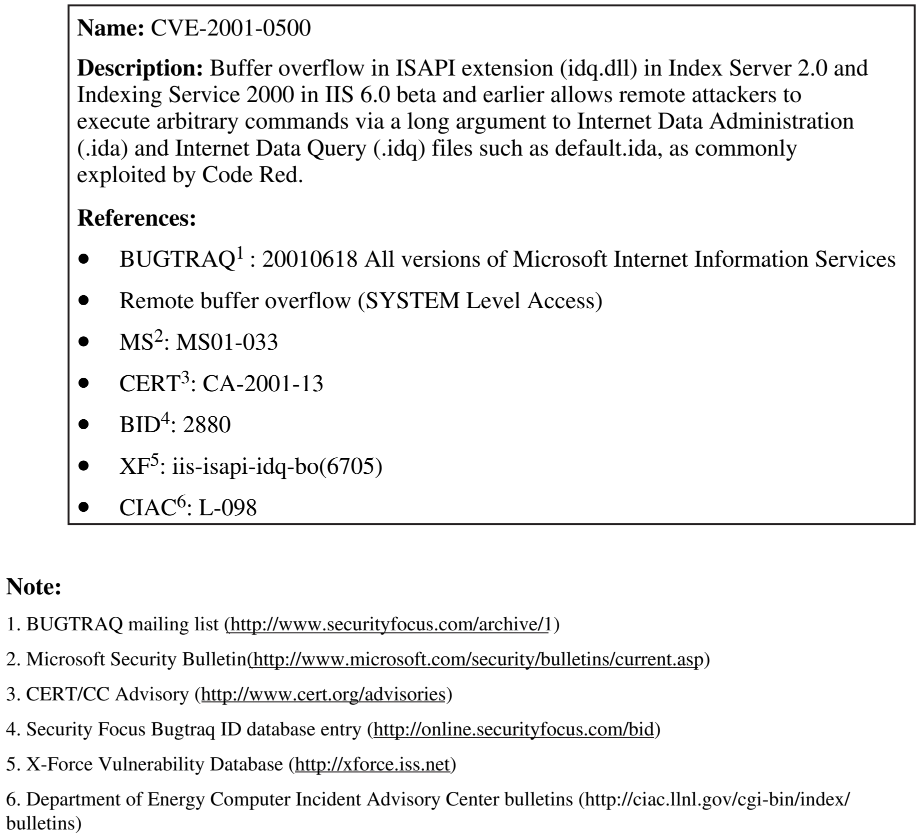
During the taxonomy's development, several model taxonomies were attempted without success. The initial approach was to create a taxonomy analogous to the animal kingdom's taxonomy. The resulting taxonomy would be a tree-like structure with the more general categories at the top, and specific categories at the leaves. This is a logical method of representation and is consistent with representing more general categories at the root of the tree and more specific categories further down the tree. However, while such a taxonomy is certainly desirable, in practice it is not possible to implement in an acceptable manner.

The first problem with such a taxonomy is how to deal with blended attacks. To allow for attacks to contain other attacks there are two possible solutions. One is to allow for cross-tree references, that is when one leaf node points to another leaf node somewhere else in the taxonomy. This approach leads to a messy tree and would be hard to use in classification. The second is to have recursive trees, so that each leaf on the base tree may have another tree (or more) under it. This again leads to a messy structure and would be of limited use.

The second problem is that attacks, unlike animals, often do not have many common traits. This makes the creation of broad categories hard. While worms and viruses have much in common with each other[2] they do not directly have a lot in common with other attacks such as Denial of Service and trojans, although in some cases such attacks can be components of worms and viruses. This means that the taxonomy tree would have to branch out immediately into a number of unrelated categories. The benefits of the tree-like structure are therefore lost. With these two problems, the tree-like taxonomy was discarded.

Another way taxonomies are sometimes created is through lists. A list-based taxonomy contains a flat-list of categories. There are two approaches

[2] As both are self-replicating.



Name: CVE-2001-0500

Description: Buffer overflow in ISAPI extension (idq.dll) in Index Server 2.0 and Indexing Service 2000 in IIS 6.0 beta and earlier allows remote attackers to execute arbitrary commands via a long argument to Internet Data Administration (.ida) and Internet Data Query (.idq) files such as default.ida, as commonly exploited by Code Red.

References:

- BUGTRAQ[1] : 20010618 All versions of Microsoft Internet Information Services
- Remote buffer overflow (SYSTEM Level Access)
- MS[2]: MS01-033
- CERT[3]: CA-2001-13
- BID[4]: 2880
- XF[5]: iis-isapi-idq-bo(6705)
- CIAC[6]: L-098

Note:

1. BUGTRAQ mailing list (http://www.securityfocus.com/archive/1)
2. Microsoft Security Bulletin(http://www.microsoft.com/security/bulletins/current.asp)
3. CERT/CC Advisory (http://www.cert.org/advisories)
4. Security Focus Bugtraq ID database entry (http://online.securityfocus.com/bid)
5. X-Force Vulnerability Database (http://xforce.iss.net)
6. Department of Energy Computer Incident Advisory Center bulletins (http://ciac.llnl.gov/cgi-bin/index/bulletins)

Figure 4 Sample CVE entry (CVE-2001-0500).

that could have been taken in the proposed taxonomy. Firstly, a flat-list with general categories could be suggested, or secondly, a flat-list with very specific categories could be proposed. The problem with the first case is that general categories are of limited use. In the domain of network and computer attacks, the categories would have to be very general to accommodate the problem of blended attacks. Such a general taxonomy will not be very useful. The second case also suffers from the problem of blended attacks. If very specific categories were chosen, such that any type of blended attack had a category, the list would become almost infinite, with few instances within each category.

The proposed taxonomy takes a different approach from either of the tree-like or flat-list taxonomies. However, both of these approaches are used by the proposed taxonomy as components and are explained in the following sections.

Overview

The proposed taxonomy works by using the concept of dimensions. Dimensions are a way of allowing for a classification of an attack to take a more holistic view of such an attack. The taxonomy proposes four dimensions for attack classification. Before examining how the taxonomy works, the dimensions to be used are briefly explained.

The first, or base, dimension is used to categorise the attack into an attack class that is based on the attack vector,[3] or if there is no attack vector, the attack is classified into the closest category.

The attack target is covered in the second dimension. The target can be classified down to very specific targets, such as Sendmail 8.12.10 or can cover a class of targets, such as Unix-based systems.

The third dimension covers the vulnerabilities and exploits, if they exist, that the attack uses. The vulnerabilities and exploits do not have a structured classification due to the possible infinite number of vulnerabilities and exploits. Instead the list defined by the CVE (Common Vulnerabilities Exposures) project (CVE, 2003) is used as a starting point (Fig. 4).

The fourth dimension takes into account the possibility for an attack to have a payload or effect beyond itself. In many cases an attack will be

[3] The attack vector is the method by which an attack reaches its target.

clearly defined, but yet it will have a payload or cause an effect that is different. For example, a virus that installs a trojan horse, is still clearly a virus, but has a trojan as a payload. In each dimension, the classifier must classify attacks as specifically as possible. This means attacks should be classified down to the smallest sub-class in each dimension that makes sense.

The taxonomy allows for the possibility of further dimensions which, although not necessary, may enhance the knowledge of the attack. Some further dimensions are discussed in section "Other dimensions". An attack must have at least the first dimension, but depending on the attack, or how specific the classifier wishes to be, all, some or none of the other dimensions may be used. The next section explains the details of each dimension and how they work to provide such a classification.

Classification using dimensions

The following sections describe how each dimension works and how the dimensions work together to provide a classification. For examples of the taxonomy applied to various attacks, including a detailed examination of the Morris Worm, see section "Classification case study".

The first dimension

Classification in the first dimension consists of two options:

- If the attack uses a single attack vector, categorise by the vector.
- Otherwise find the most appropriate category, using the descriptions for each category below.

The attack vector of an attack is the main means by which the attack reaches its target. For example, the Melissa "Virus" uses email as its main form of propagation, and therefore is, in the first dimension, a mass-mailing worm. The virus-like capabilities of Melissa are handled in the other dimensions.

It is very important that attack vectors are identified if possible, as they provide the most accurate description of an attack. For example, an attack that infects computers through a TCP network service and then installs a trojan on the infected computer, should be classified by its attack vector — which is a worm (i.e., it spreads via network services). If it is classified as a trojan instead, then there is no opportunity to describe the worm-like behaviour of the attack, which is essentially the most important feature of the attack.

If an attack vector is not present or is too trivial[4] then the attack can be categorised by finding the category closest to how the attack works. For example, an attack run locally that gains control of another process by overflowing a buffer, is a buffer overflow attack.

The following definitions assist in categorising attacks which lack obvious attack vectors. The category that best matches with the definitions below is chosen. Once the general class has been chosen, the attack may be further classified by using the sub-classes, if they exist.

- *Virus*: self-replicating program that propagates through some form of infected files
- *Worms*: self-replicating program that propagates without using infected files; usually worms propagate through network services on computers or through email.
- *Trojans*: a program made to appear benign that serves some malicious purpose
- *Buffer overflows*: a process that gains control or crashes another process by overflowing the other process's buffer
- *Denial of service attacks*: an attack which prevents legitimate users from accessing or using a host or network
- *Network attacks*: attacks focused on attacking a network or the users on the network by manipulating network protocols, ranging from the data-link layer to the application layer
- *Physical attacks*: attacks based on damaging physical components of a network or computer
- *Password attacks*: attacks aimed at gaining a password
- *Information gathering attacks*: attacks in which no physical or digital damage is carried out and no subversion occurs, but in which important information is gained by the attacker, possibly to be used in a further attack

The first dimension is summarised in Table 1. The categories are reasonably broad. To categorise more specifically, other dimensions need to be used. The categories that can be used as attack vectors are: viruses, worms and trojans. These categories have the necessary characteristics[5] to be vectors. While it may not be impossible to use another category as an attack vector, it should be a rare occurrence and would suggest that either

[4] That is, the vector is outside the categories defined in the first dimension.

[5] Such as having the ability to carry other attacks.

Table 1 The first dimension's categories

Level 1	Level 2	Level 3
Viruses:	File infectors	
	System/boot record infectors	
	Macro	
Worms:	Mass mailing	
	Network aware	
Buffer overflows:	Stack	
	Heap	
Denial of service attacks:	Host-based:	Resource hogs
		Crashers
	Network-based:	TCP flooding
		UDP flooding
		ICMP flooding
	Distributed	
Network attacks:	Spoofing	
	Session hijacking	
	Wireless attacks:	WEP cracking
	Web application attacks	Cross site scripting
		Parameter tampering
		Cookie poisoning
		Database attacks
		Hidden field manipulation
Physical attacks:	Basic	
	Energy weapon:	HERF
		LERF
		EMP
	Van Eck	
Password attacks:	Guessing:	Brute force
		Dictionary attack
	Exploiting implementation	
Information gathering attacks:	Sniffing:	Packet sniffing
	Mapping	
	Security scanning	

a new category has been identified or an incorrect classification has been made.

The second dimension

The second dimension covers the target(s) of the attack. As an attack may have multiple targets, there may be multiple entries in this dimension. It is important to note that targets should be made specific. That is, for an attack on Server A, we are not concerned that Server A was attacked. Rather the operating system of Server A and service that was attacked are important. So for example, if Code Red attacked Server A, the target would not be Server A, but the IIS service running on this machine.

A further consideration occurs when an attack targets a specific configuration of a target. For example, vulnerabilities may be introduced by incorrectly configuring a web server. In such a case, the second dimension does not, by itself, categorise this. However, the second and third dimensions can be used together to cover this type of vulnerability. The second dimension categorises what the target is, while the third dimension categorises what is being used to attack the target. Therefore in the above example, the second dimension covers the web server, while the third dimension covers the vulnerabilities introduced by the configuration.

Table 2 shows samples of the categories of the second dimension. There are a wide range of potential targets and each year the list increases. Instead of providing an exhaustive list, a generalised way of classifying the targets is shown, with a few specific examples. The entries in Table 2 that contain "..." show where extra categories can be added to the classification. Extra entries should be added in a way that conforms to how the sibling categories have been defined. For example, if adding a category for the DOS operating system, firstly a "DOS Family" entry should be created under Software → Operating System, then the flavours of DOS should be created within the "DOS Family" entry. Finally, within each flavour of DOS entry, specific versions should be created.

Table 2 The second dimension's categories

Level 1	Level 2	Level 3	Level 4	Level 5	Level 6
Hardware:	Computer:	Hard-disks	…		
		Network equipment:	Routers		
			Switches		
			Hubs		
			Cabling		
			…		
		Peripheral devices:	Monitor		
			Keyboard		
			…		
Software:	Operating system:	Windows family:	Windows XP		
			Windows 2003 Server		
			…		
		Unix family	Linux:	RedHat Linux 6.0	
				RedHat Linux 7.0	
				…	
			FreeBSD:	4.8	
				5.1	
				…	
			…		
		MacOS family	MacOS X:	10.1	
				10.2	
				…	
			…		
		…			
	Application:	Server:	Database	…	
			Email	…	
			Web:	IIS:	4.0
					5.0
					…
			…		
		User:	Word processor	MS Word:	2000
					2003
					…
			…		
			Email client:	…	
			…		
		…			
	Network:	Protocols:	Transport-layer:	IP	…
				Network-layer:	TCP
					…
				…	
			…		
		…			

The leaf nodes of the structure should be specific versions of a product that is being targeted. If a category for the product does not exist, a new category should be created using the above method, thus allowing for specific versions to reside in that category.

Hardware targets can be broken down into three main sub-classes: computer, network equipment and peripheral devices. Computer targets are computer components, such as CPUs and hard-disks. Network equipment might be devices such as routers, switches or hubs. Finally, peripheral devices are devices that are not essential[6] to a computer's operation — for example monitors.

Software targets have two main classes: operating systems and applications. Operating system targets are targets within the operating system itself, while application targets are targets that are running on top of the operating system.

[6] Essential devices are ones that the computer could not operate without. For example, the CPU and memory are essential.

Finally, a network target is one in which the network itself or its protocols are targeted. For example, a ping-flood attacks a network rather than hardware or software.

The third dimension

The third dimension covers the vulnerabilities and exploits that the attack uses. An attack may exploit multiple vulnerabilities, so there may be more than one entry in the third dimension. Entries in the third dimension are usually a Common Vulnerabilities and Exposures (CVE) entry, but in the case that a CVE entry does not exist, the vulnerability is classified generally as described later in this section.

The Common Vulnerabilities and Exposures project (CVE, 2003) is designed to produce common definitions of vulnerabilities. The idea for CVE was proposed by Mann and Christey (1999). The CVE project has become the de facto standard for vulnerabilities and so it is desirable that the proposed taxonomy utilises this. It should be noted that vulnerabilities are wide and varied and usually apply to specific versions of a piece of software or operating systems. This means that a classification scheme would have to include every piece of software in use today.

Below is an example of a CVE entry showing a vulnerability in Microsoft's Internet Information Services which is exploited by the Code Red worm.

Once the vulnerability or vulnerabilities that an attack exploits are known, the relevant CVE entries can be found. Howard (1997) suggests three general types of vulnerabilities:

- *Vulnerability in implementation*: The design of the system is secure, but the implementation fails to meet the design and thus vulnerabilities are introduced. Buffer overflows often exploit such vulnerabilities, for example a program may be designed securely, but its implementation contains bugs that can be exploited.
- *Vulnerability in design*: The fundamental design of the system is flawed, so that even a perfect implementation will have vulnerabilities. For example, a system which allows users to choose weak passwords will have a vulnerability in its design.
- *Vulnerability in configuration*: The configuration of the system introduces vulnerabilities. The system itself may be secure but if configured incorrectly, renders itself vulnerable. An example would be installing a secured operating system and then opening a number of vulnerable ports.

If no CVE entry exists, then one of Howard's types of vulnerabilities should be selected, and a description of the vulnerability should be created. As time progresses, CVE entries may be added, in which case classifications may have to be updated to reflect this.

The fourth dimension

The fourth dimension deals with attacks having payloads or effects beyond themselves. For example, a worm may have a trojan payload, or it may simply destroy some files. The payload may be another attack itself and so the first dimension can be used to classify the payload if this is the case. Thus, the taxonomy allows for attacks (first dimension attack) to launch other attacks (fourth dimension payloads). The fourth dimension consists of five categories:

1. First dimension attack payload (section "The first dimension")
2. Corruption of information
3. Disclosure of information
4. Theft of service
5. Subversion

Categories 2—4 were previously identified by Howard (1997). Corruption of information occurs when a payload corrupts or destroys some information. When a payload discloses information that is not intended by the victim to be disclosed, the payload is a disclosure of information payload. Theft of service payloads use a system's services without authorisation, but without impacting the service of legitimate users. Howard has a fourth category, denial of service. However, this possibility is covered in Category 1. Finally, a subversion payload will gain control over part of the target and use it for its own use.

It should be noted that apart from the First Dimension Attack Payload, the categories are general. This is because while general types of payloads can be identified, there is a wide range of implementations of the various payloads. For example, two attacks may corrupt information in that they delete files, but may only differ in which files they delete. In most cases it should be possible to use a first dimension category as the payload.

An attack may have multiple entries in this dimension, and the categorisation need not be mutually exclusive. If the attack cannot be categorised using the first category, any number of the remaining categories can be used. For example, some payloads may both disclose information and steal service at the same time.

Other dimensions
Besides the four dimensions described above, a number of further dimensions could be added to enhance the taxonomy. Several are discussed below and although they are more abstract and are not as essential as the dimensions previously described, they are still useful in classifying attacks, especially in regards to how to react to a new attack that falls into a certain category. For example, the following are dimensions that would be useful for an organisation dealing with attacks:

- *Damage*: A damage dimension would attempt to measure the amount of damage that the attack does. An attack such as the recent SoBig virus cause more damage than a simple virus such as the Infector virus.
- *Cost*: Cleaning up after an attack costs money. In some cases millions of dollars are spent on attack recovery.
- *Propagation*: This category applies more to replicating attacks. The propagation of an attack is the speed at which it reproduces or spreads. For attacks such as worms and viruses, a dimension covering this aspect would be useful.
- *Defence*: The methods by which an attack has been defended against could be made into a further defence dimension.

It should be noted that the new dimensions suggested above are "post-attack" dimensions. That is, the attack will have to have had time to show its attack potential, so that an accurate assessment of the damage or cost can be made. The four base dimensions, however, can be applied relatively soon after the attack has been launched. There is also the possibility for classification refinement, so that as more information is known about an attack, the classification is made more specific.

Classification case study

Table 3 shows the results of classifying a number of attacks using the proposed taxonomy. The table shows the first, second and fourth dimensions in full, but the second dimension has been truncated to show only the final entry. So for example, Code Red's second dimension is Software → Application → Server → Web → IIS → Versions 4, 5, and 6.0 beta, but only IIS 4, 5 and 6.0 beta are shown. Also some entries are not complete, for example the Land attack has more than 40 different operating systems that it targets. Only a few of these are shown, but in a complete entry, all targets would be included. To elaborate on the classification process further, the Morris Worm's classification is discussed below.

The Morris Worm consisted of a number of components which made it a dangerous blended attack. The worm consisted of three main components which were used to spread and infect:

- The Sendmail attack
- The Fingerd attack
- The Rsh/Rexec attack

More details on the worm can be found in Eichin and Rochlis (1988) and Spafford (1988). The worm used each of these methods to spread, and thus it had three attack vectors. The first dimension categorisation therefore is a worm, as the attack propagated without using infected files and had multiple attack vectors. As it also used network services to spread, it is therefore a network-aware worm. The worm attacked Sun Microsystems Sun 3 and VAX computers running BSD 4 variants. Therefore the second dimension consists of the entry: Software → Operating systems → Unix family → BSD family → 4 → VAX variants & Sun 3 Variants. Note the three attacks above use the vulnerabilities discussed below to attack VAX and Sun 3 BSD variants (that is, Sendmail on the VAX and Sun 3 systems, for example, is not so much a target as it is a vulnerability).

The worm used a number of vulnerabilities to spread. As the CVE project does not go as far back as the Morris Worm, the broader categories in the third dimension are used. Namely, a vulnerability in design for both the Sendmail and Fingerd attacks (as both exploited bugs in the implementation of Sendmail and Fingerd) and a vulnerability in implementation for the Rsh/Rexec attack (as weak passwords were targeted). Finally, the fourth dimension categorisation consists of two entries: theft of service (as the worm stole both network and computer resources) and subversion (as infected systems were used to propagate the worm).

Conclusions

The proposed taxonomy is a good start towards a taxonomy for computer and network attacks. In general it works well, and attacks are easily categorised. However, as always, there is room for improvement. As described in the above sections, some requirements have not been fully met and some areas could do with refinement.

Blended attacks were sometimes difficult to categorise as they contained numerous sub-attacks.

Table 3 Classification results

Attack	1st Dimension	2nd Dimension	3rd Dimension	4th Dimension
Blaster	Network-aware worm	MS Windows NT 4.0, 2000, XP, Server 2003	CAN-2003-0352	TCP packet flooding DoS
Chernobyl	File infector virus	MS Windows 95 & 98		Corruption of information
Code Red	Network-aware worm	IIS 4, 5 & 6.0 beta	CVE-2001-0500	Stack buffer overflow & TCP packet flooding DoS
Use of John the Ripper	Guessing password attack	Unix family, Windows NT, 2000 & XP	Configuration	Disclosure of information
Infector	File infector virus	DOS family		Host-based crasher DoS
Land	Crasher DoS	Windows 95 and NT 4.0, Windows for Workgroups, 3.11, ...	CVE-1999-016	
Melissa	Mass-mailing worm	MS Word 97 & 2000	Configuration	Macro virus & TCP packet flooding DoS
Michelangelo	System boot record infector virus	DOS family		Corruption of information
Nimda	Mass-mailing worm	MS IE 5.5 SP1 & earlier except 5.01 SP2	CVE-2001-0333 & CVE-2001-0154	File infector virus, Trojan and DoS
PKZIP 3 Trojan	Trojan	DOS family		Corruption of information
Ramen	Network-aware worm	RedHat Linux 6.2 & 7.0	CVE-2000-0573, CVE-2000-0666 & CVE-2000-0917	Host-based DOS, UDP and TCP packet flooding DoS & subversion
Slammer	Network-aware worm	MS SQL Server 2000	CAN-2002-0649	Stack buffer overflow & UDP packet flooding DoS
Sobig.F	Mass-mailing worm	Email client	Configuration	Trojan
Trojaned Wuarchive FTPD	Trojan	Unix family		Subversion
Morris worm	Network-aware worm	BSD 4 Sun 3 & VAX variants	Implementation & design	Theft of service & subversion

The issue here is not so much the taxonomy, but how the blended attacks have been analysed and described. Sometimes blended attacks are analysed in a way that mixes sub-attacks together. Therefore, the classifier must be able to sift through blended attack descriptions to find the information required. Future work on how to sift through attack descriptions would be helpful.

Attacks that have targets (or vulnerabilities) that require other targets are not fully modelled in the taxonomy. It would be useful in future versions of the taxonomy to be able to relate items within a dimension better. Relating items so that an attack can have a combination of targets that are required, rather than a list of targets that have no relationship, would be useful.

To help understand classifications better, and to correlate attacks, some form of visualisation would be useful. Due to taxonomy having four dimensions, this is a non-trivial task. However, even if not all the information contained within the dimensions is presented, some form of visualisation allowing correlation between attacks would be helpful.

Research on correlation between attacks within the taxonomy would be interesting. The dimensions allow for attacks to be correlated through properties such as the vulnerabilities used by attacks. This means attacks that previously may have appeared to have nothing in common can be related through one of the dimensions. More research could be carried out on how this works and how beneficial it could be.

Further work could be carried out in moving the taxonomy towards a knowledge base approach. That is, as new classifications are created, they are added to a knowledge base. The knowledge base could detect correlations and allow for greater analysis of existing attacks. Another aspect would be the classification process. A step-by-step questionnaire could be used to ease classification. For example, the first few steps for classifying a worm in the first dimension might consist of:

- Is the attack self-replicating? (Yes = worm or virus, No = other 1st dimension attack)
- Does the self-replicating attack propagate through infected files? (Yes = virus, No = worm)
- Does the worm spread through email? (Yes = mass-mailing worm, No = network-aware worm)

This would continue until the worm has been classified in the all dimensions and would make the process of classifying easier and reduce the chance of error.

If a knowledge base was implemented, artificial intelligence (AI) could be used to test the taxonomy. The knowledge base could then be learnt by the AI system, and new attacks could be given to the AI system to classify.

Acknowledgement

The authors acknowledge the support offered by the CCIP (Centre for Critical Information Protection) of the New Zealand Government.

References

Abbott RP, Chin JS, Donnelley JE, Konigsford WL, Tokubo S, Webb DA. Security analysis and enhancements of computer operating systems. Technical Report NBSIR 76 1041, Institute for Computer Sciences and Technology, National Bureau of Standards; April 1976.

Amoroso E. Fundamentals of computer security technology. Englewood Cliffs, New Jersey: P T R Prentice Hall; 1994.

Aslam T. A taxonomy of security faults in the Unix operating system. Master's thesis, Purdue University; 1995.

Bisbey II R, Hollingworth D. Protection analysis: final report. Technical report, University of Southern California; May 1978.

Bishop M. A taxonomy of (Unix) system and network vulnerabilities. Technical Report CSE-9510, Department of Computer Science, University of California at Davis; May 1995.

Bishop M, Bailey D. A critical analysis of vulnerability taxonomies; September 1996.

Bishop M. Vulnerabilities analysis. International symposium on recent advances in intrusion detection; 1999.

CERT Coordination Center. CERT/CC statistics, <http://www.cert.org/stats/cert_stats.html>; 2003.

CVE. Common vulnerabilities and exposures. <http://www.cve.mitre.org/>; 2003.

Eichin M, Rochlis J. With microscope and tweezers: an analysis of the internet virus of November 1988. Technical report, Massachusetts Institute of Technology; 1988.

Howard JD. An analysis of security incidents on the internet 1989–1995. PhD thesis, Carnegie Mellon University; 1997.

Howard JD, Thomas A Longstaff. A common language for computer security incidents. Technical report, Sandia National Laboratories; 1998.

Krsul IV. Software vulnerability analysis. PhD thesis, Purdue University; 1998.

Lindqvist U, Jonsson E. How to systematically classify computer security intrusions. IEEE Security and Privacy 1997:154–63.

Lipson HF. Tracking and tracing cyber-attacks: technical challenges and global policy issues. Technical report, CERT Coordination Center; November 2002.

Lough DL. A taxonomy of computer attacks with applications to wireless networks. PhD thesis, Virginia Polytechnic Institute and State University; 2001.

Mann DE, Christey SM. Common vulnerabilities and exposures. Technical report, The MITRE Corporation, <http://www.cve.mitre.org/docs/cerias.html/>; 1999.

OASIS WAS TC. OASIS Web Application Security Technical Committee. <http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=was>; 2003a.

OASIS WAS TC. OASIS Web Application Security Technical Committee list archives, <http://lists.oasis-open.org/archives/was/>; 2003b.

Spafford E. The internet worm program: an analysis. Technical report, Department of Computer Sciences, Purdue University; 1988.

Ray Hunt is an Associate Professor specialising in Networks and Security in the Department of Computer Science and Software Engineering at the University of Canterbury, New Zealand. He has been involved with industry-based studies in the area of Wireless LAN performance and security and runs a laboratory with support from Telecom New Zealand in which a variety of performance and security experiments are carried out.

Simon Hansman is the Lead Systems Developer at Lakros Technologies. He completed an honours degree in Computer Science and Software Engineering at the University of Canterbury in 2003.