An Ontology for Network Security Attacks

Andrew Simmonds[1], Peter Sandilands[1], Louis van Ekert[1]

[1]Faculty of IT, University of Technology Sydney, PO Box 123, Broadway, NSW 2007, Australia
{simmonds, psandy, ekertl}@it.uts.edu.au

Abstract. We first consider network security services and then review threats, vulnerabilities and failure modes. This review is based on standard texts, using well-known concepts, categorizations, and methods, e.g. risk analysis using asset-based threat profiles and vulnerability profiles (attributes). The review is used to construct a framework which is then used to define an extensible ontology for network security attacks. We present a conceptualization of this ontology in figure 1. **Keywords**: network, cyber, security, ontology, attack, threat, vulnerability, failure.

1 Introduction

This article was written as a result of the authors teaching a network security subject in the Faculty of IT, at the University of Technology Sydney. There are many concepts which need to be well understood by network security students and practitioners. To assist in this there have been several attempts to classify different aspects of the subject area. This article lists some of the common taxonomies, shows the relationship between them, and modifies or extends them where appropriate to make them consistent, and then defines an extensible ontology for network security based on this material. The article provides a framework to locate these taxonomies in the network security subject area. The aim of this article is thus to provide a new and improved understanding of the linkages between different components of a network security system.

In part 2 we consider security services; in part 3 we look at threats and system weaknesses; in part 4 we review failure modes - recognizing that perfect security is not achievable in practice; and finally in part 5 we define an ontology for network security attacks

2 Security Services

There are two mnemonics commonly used to summarize services which a network security system should provide: 'CIA' and 'Triple A' (see tables 1 and 2). CIA provides a key to remember three important security services (Confidentiality, Integrity and Availability), but really another three services should be added

(Authentication, Access Control and Non-repudiation), see Stallings (2000), to make 'CIA+' (table 1). Integrity is sometimes used to refer to the ability to prevent all the outcomes outlined in table 3 (part 5: Outcome) below, but we will use it in a narrower sense to mean the ability to guard against message modification.

The 'Triple A' mnemonic is useful in that it makes clear the relationship between these three services: you cannot use the accounting service until you have been authorized, and you cannot be authorized until you have been authenticated.

Table 1. Security Services CIA+

CIA+
1.Confidentiality
2.Integrity
3.Availability
plus:
4.Authentication
4.1.of people (*something you know, have, are*)
4.2.of organizations
4.3.of applications
5.Access Control
6.Non-repudiation

Table 2. 'Triple A' Services

Triple A
1. Authentication
2. Authorization
3. Accounting

3 Know the enemy and know yourself

Sun-Tzu states (400 – 320 BCE, translated Giles, 1910) "If you know the enemy and know yourself, you need not fear the result of a hundred battles". There is a clear need to understand different attacks and the people who would stage them.

Threat Profiles (table 3) considers individual threats. This table is from work on OCTAVE, by Wilson (2002), and Alberts and Dorofee. Each threat profile should be classified by its possible impact: low/medium/high. There are three phases to OCTAVE:

(i) build asset-based Threat Profiles (from table 3), marked low/medium/high impact;

(ii) identify vulnerabilities from Vulnerability Profiles (table 8);

(iii) develop a Security Strategy and Plan (based on a risk assessment from all the threat and vulnerability profiles).

The summation of the threat and vulnerability profiles will enable a risk assessment to be made, which together with other factors such as usability and cost determines the appropriate level of security for an organization. As there is no such thing as perfect security, there is always a trade-off, especially between (a) security and cost, and (b) security and usability.

In table 3 part 3, the term hacker is somewhat fluid: it is often used by the press to refer to someone who seeks to penetrate a computer system to steal or corrupt data, whereas people who call themselves hackers would reject that definition and use the term to describe someone who is enthusiastic and knowledgeable about computer systems. To avoid this confusion we use the term 'white hat' and 'black hat' (from the days of black and white cowboy films). Thus a 'white hat' hacker might be employed to test a system for flaws, whilst a 'black hat' hacker is synonymous with a cracker. A script kiddie is someone who uses already established and part automated techniques in attacking a system. Their expertise is less than a hacker, but still considerably more than a normal computer use. It would be unusual to have a 'white hat' script kiddie, so without a hat colour descriptor they are taken to be on the side of the black hats.

Table 4, which is an extension of a common classification scheme [e.g. Stallings (2000)], categorizes attacks in different ways and we then show examples of how to apply these categories to different types of threat in table 5. In table 4, some active attacks target the message - these are direct attacks on CIA. Other active attacks attempt to gain some level of control of the system. Once the system is compromised in this way then messages may be attacked, but this would be an indirect attack on CIA. The stages of an active attack to gain control of a system (table 6) are adapted from Cates (2003). Steps 1 – 3 are concerned with gaining access.

Table 3. Threat Profiles

1. Asset	**2. Access**	**3. Actor**
1.Intangible	(*attack on Access Control*)	1. Script kiddie
1.1.Trust	1.Physical	2. 'Black hat' hacker
1.2.Reputation	1.1.internal	3. Cracker
2.Information	- Trojan, bomb	4. Malevolent user
2.1.Sensitivity	1.2.physical	5. Malevolent sys admin
- unrestricted	2.Network	
- restricted	2.1.server	
- controlled	2.2.client	
2.2.Classification	2.3.man-in-middle	
- customer	3.Logical	
- business		

- employee	**4. Motive**	**5. Outcome**	***attack on***
2.3.Access	1.Accidental	Interruption	Availability
- internal employee	2.Deliberate:	Interception	Confidentiality
- external employee	2.1.Fun	Modification	Integrity
- business partners	2.2.Revenge	Fabrication	Authentication
- customers	2.3.Gain		
- 3rd parties	- Direct		
	- Indirect		

Table 4. Attack Classification

1. Active attack
1.1 Direct attack on CIA
Spoofing (Masquerade)
Replay
Modification of message contents
DoS
1.2 Attack on control of system
Root access - see table 6
Blind attack
1.3 Active attack identifiers
1.3.1. Program (complete or fragment)
1.3.2. Replicates (Yes/No)
2. Passive attack
Release of message contents
Traffic Analysis

Table 5. Some active attack threat examples

Threat	Active attack	Program	Replicates
Bacteria	DoS	yes	yes
Worm	blind attack	yes	yes
Virus	blind attack	fragment	yes
Trojan horse	root access	yes	no
Logic bomb	root access	fragment	no

Table 6. Active attack steps to gain root access

1. Reconnaissance
2. Get a shell
3. Elevate access rights
4. Make a back door
5. Execute attack
6. Erase the trail

Table 7. Severity (influence on system)

1. admin access
2. read restricted files
3. regular user access
4. spoofing
5. non-detectability
6. DoS

Sun Tzu also emphasizes the need to understand your vulnerabilities and weaknesses. Table 8 showing Vulnerability Profiles (or attributes) is drawn from Knight (2000), the notes show which other tables expand the entry. The severity (table 7 - with 1 highest severity) is from the point of view of the computer being attacked, not from the point of view of the resulting outcome or damage to the organization. In table 10, based on the "Map of Vulnerability Types" of Knight

(2000), the left side shows attacks and weaknesses of the security policy, whilst the right hand side shows technology vulnerabilities.

Table 8. Vulnerability Profiles

Fault Taxonomy – see table 9 from Aslam, Krsul and Spafford (1996)
Severity – see table 7
Authentication – see table 1
Tactics – this is subsumed into table 3.2 (Access)
Vulnerability Map – see table 10
Consequence – this can be taken to be the same as table 3.5 (Outcome)

Table 9. Fault Taxonomy

1.Coding faults
1.1.Synchronization errors – race conditions
1.2.Condition validation errors – buffer overflows, etc.
2.Emergent faults
2.1.Configuration errors – incorrect permissions
2.2.Environment faults – different modules interact unexpectedly

Table 10. Vulnerability Map

Security Policy	Technology	Time scale
1.Social Engineering - attack on Security Policy, e.g.	**2.Logic erro**r - attack on technology (see also Table 9)	Short-term
- Information fishing	2.1.bugs	
- Trojan	2.2.OS/application vulnerabilities	
	2.3.Network Protocol Design	
3.Policy oversight - weakness of Security Policy	**4.Weakness -** of technology, e.g.	Long-term
3.1.poor planning	- Weak password system	
3.2.poor control, e.g. allowing weak passwords	- Old encryption standards	

4. Failure

Since there is no such thing as perfect security, we need to consider how a system will react to a successful attack. Indeed for Schneier (2002) the most critical part of a security system is not how well it works but how well it fails. He categorizes systems as either brittle or ductile. The point being that a strong but brittle security system that

catastrophically fails is worse than a weaker but ductile system that degrades gradually (i.e. fails 'gracefully').

The number of faults that cause a system to fail can be (a) single, (b) dual, or (c) > 2 simultaneous failures ('baroque' faults). If a single event causes a system to fail then this (in table 9 Fault Taxonomy) is a coding fault. In a well designed system, more common causes of failure are dual faults or baroque faults (emergent faults in table 9).

To mitigate against failure, security systems should be small-scale, redundant and compartmentalized, and avoid a Single Point Of Failure (SPOF).

5. Network Security Attacks Ontology

This is a proposal to initiate the design of an ontology for network security attacks, it is meant to be extended. An ontology in this sense is an extensible specification of a vocabulary (McGuinness 2002), i.e. an attempt to define some realm of interest for network security. Together with the terms we have introduced in the previous tables (which become the classes in our ontology), we need properties to determine the relationship between the classes. In figure 1, the circles are the classes, with the number inside referring to the appropriate table (or sub-table), the arcs are the properties.

Figure 1 is meant to be used in conjunction with the tables presented in this paper. Thus the class 'Actor' with the annotation 3.3, means refer to table 3 part 3 for a breakdown of possible actors. The review and summarization of network security classifications in sections 2 and 3 thus forms the basis for the ontology presented here.

The classes (and sub-classes) for this Network Security Attacks Ontology are: **Access**, **Actor** (Black hat hacker, Cracker, Malevolent user, Malevolent Systems Administrator, Script kiddie), **Attack** (Attack on control of system, DoS, Modification of message contents, Release of message contents, Replay, Spoofing, Traffic analysis), **Impact**, **Information**, **Intangible** (Reputation, Trust), **Motive** (Fun, Gain, Revenge), **Outcome** (Fabrication, Interception, Interruption, Modification), **Systems Administrator**, **Threat** (Bacteria, Logic bomb, Trojan horse, Virus, Worm).

The properties are: **assesses**, **causes loss of**, **gains**, **has**, **loses**, **makes**, **reports**, **uses**.

Some other security ontologies are an ontology for describing trust relationships for web services, see also Kagal et al (2003, 2004), Denker (2003); and an ontology describing the National Security Organization of the US. Both these ontologies can be found in the on-line list at DAML (DARPA Agent Markup Language).

Conclusion

We have presented a framework for network security based on proven concepts. From this review we present an ontology for network security attacks which shows the relationship between many of the standard classifications used, with the concep-

tualization drawn in figure 1. The conceptualization is linked to the tables reviewed and presented in this paper.

In addition we have consolidated the work done for analyzing system vulnerabilities, see table 8 which gives a starting point for drawing up vulnerability profiles, and for analyzing threat profiles, see table 3.

The next step, after getting feedback and refining this proposal, is to create a machine readable form of this ontology.

References

Alberts, Christopher and Dorofee, Audrey *OCTAVE Threat Profiles*. Carnegie Mellon Software Engineering Institute, Pittsburgh, PA 15213, USA. Available from http://www.cert.org/archive/pdf/OCTAVEthreatProfiles.pdf [accessed 12th April 2004].

Aslam; Krsul and Spafford (1996) *A Taxonomy of Security Faults*. Purdue University COAST Lab. Available from: http://www.cerias.purdue.edu/about/history/coast/coast-library.html [accessed 28th March 2004].

Cates, Sol (2003) *The Art of Hacking*. TRIPWIRE Security Industry Seminar, July 28th 2003. Available from: http://www.tripwire.com/events/archived_webcasts/ [accessed 28th March 2004].

DAML, list of ontologies from: http://www.daml.org/ontologies/keyword.html [accessed 19th August 2004].

Denker, Grit et al (2003) *Security for DAML Web Services: Annotation and Matchmaking*. Proceedings, Second International Semantic Web Conference , September 2003.

Kagal, Lalana; Finin, Tim; Joshi, Anupam (2003) *A Policy Based Approach to Security for the Semantic Web*. Proceedings, 2nd International Semantic Web Conference (ISWC2003), September 2003.

Kagal, Lalana et al (2004) *Authorization and Privacy for Semantic Web Services*. Proceedings, First International Semantic Web Services Symposium, AAAI 2004 Spring Symposium, March 2004.

Knight, Eric (2000) *Computer Vulnerabilities*. Available e.g. from: http://www.fi.upm.es/~flimon/compvuln_draft.pdf [accessed 28th March 2004].

McGuiness, Deborah (2002), Knowledge Systems Laboratory, Stanford University, *Ontologies come of age* from Fensel et al (ed.) *Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential*, MIT Press. Available from http://www.ksl.stanford.edu/people/dlm/papers/ontologies-come-of-age-mit-press-(with-citation).htm [accessed 6th June 2004].

Schneier, Bruce (2002) interviewed for the Atlantic Monthly by Mann, Charles (September 2002) *Homeland Insecurity*. Available from http://www.theatlantic.com/issues/2002/09/mann.htm [accessed 12th April 2004].

Stallings, William (2000) *Network Security Essentials: Applications and Standards*. New Jersey, Prentice-Hall Inc.

Sun Tzu (400 – 320 BC) *On the Art of War*. Translated by Lionel Giles (1910). Available from: http://www.kimsoft.com/polwar.htm [accessed 28th March 2004].

Wilson, Bill (2002) *The OCTAVE Methodology for Self-Directed Risk Assessment*. Carnegie Mellon Software Engineering Institute, Pittsburgh, PA 15213, USA. Available from http://www.fedcirc.gov/library/presentations/octave.pdf [accessed 12th April 2004].

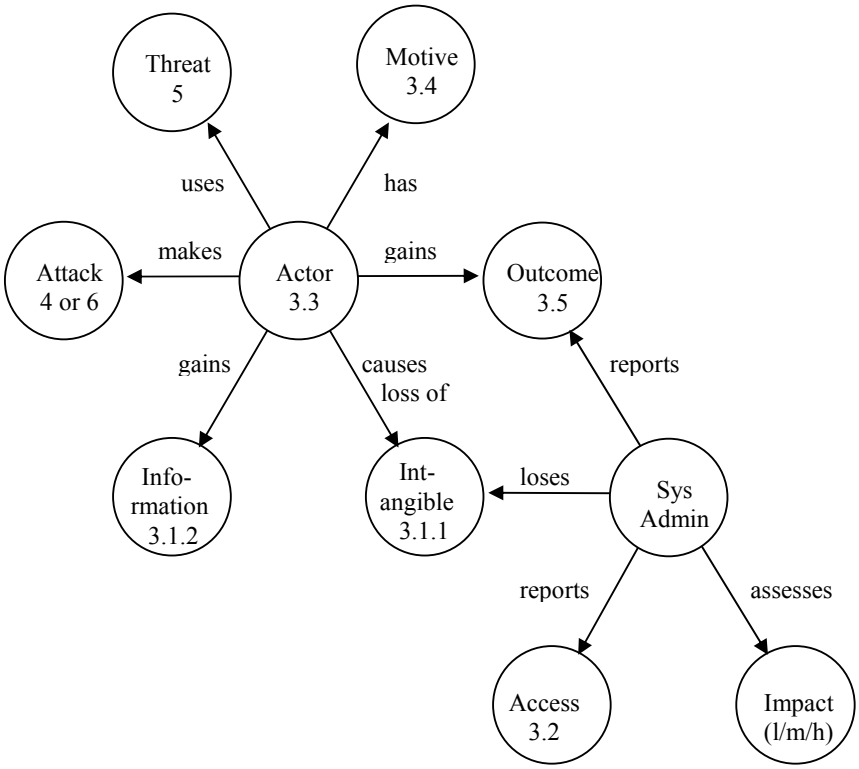

Fig. 1. Network Security conceptualization