

# The Cognitive Dimensions and Security

Luke Church

PolyMorphiX Networks  
luke@church.name

## Abstract

This position statement considers some of the applications of Cognitive Dimensions to understanding security issues in modern connected software systems.

## Introduction

Security is an increasingly important field, especially with internet connected and ubiquitous computing. Much consideration has been given to technical defenses against a variety of attacks. This position paper seeks to view the attacks from a Cognitive Dimensions[1] (CDs) perspective to help to understand the human causes of security vulnerabilities.

## General points that apply to most attacks

All unknown security vulnerabilities can be considered to be hidden dependency issues. The behaviour of the program is dependent, in an unknown manner, on the vulnerable code. Most input dependant security vulnerabilities are also viscosity related. They do not sufficiently resist the change from normal operational input to hostile input.

## Buffer Overrun attacks

The classic buffer overrun vulnerability is one where an input is not checked for length and is copied without truncation into a fixed size buffer. There is a premature commitment issue in that the size of the buffer is set before the size of the input is known. Many buffer overrun vulnerabilities have additional hidden dependencies. These are caused by interactions between functions, in which the semantics for safe operation are not followed. This is further exacerbated by incorrect trust allocation; where the author of one function believes, incorrectly, that another will make the input safe.

## Cryptographic attacks

Cryptographic systems exhibit powerful hidden dependencies that have lead to vulnerabilities. The WEP key vulnerability results from a hidden dependency on previous packets [2, 3]. Cryptographic functions often have hidden dependencies on the fundamental properties of numbers and the randomness of their random number generator [4]. Systems also exhibit multiple hidden dependencies on time. They may be directly vulnerable to timing attacks [5]. Cryptanalysis may reveal weaknesses in the algorithms [6] and as computer power improves it may become computationally feasible to brute-force exhaustive search the key space [7].

For many reasons, including the above, custom cryptographic implementations by inexperienced developers tend to be weak [8].

An exhaustive consideration of all major classes of vulnerability is beyond the scope of this document. Table 1 roughly categorises some other attacks.

**Table 1.** Other attacks by CD. Note in many cases these are simplifications

	<b>Viscosity</b>	<b>Hidden Dependency</b>	<b>Visibility</b>	<b>Role Expressiveness</b>
<b>XSS</b>	Too low	Yes	Too high	
<b>Internationalisation</b>	Too low	Yes		Poor
<b>Spoofing</b>	Too low	Yes		
<b>Security API verbosity</b>		Frequently	Too high	
<b>Format string attacks</b>	Too low	Yes		Poor
<b>Secrets compromise</b>	Too low	Frequently	Too high	
<b>Hide extensions for known file types</b>			Too low	Very poor

## Further considerations using Cognitive Dimensions

CDs can be used to consider interaction between users and security features. For example, running as a low privilege user tends to increase the viscosity of the system as it requires permission checks or a temporary change of the user's account through the use of 'su' or 'runas' to carry out administrative tasks. If an application issues too many disruptive security warnings or permission checks, the user may choose to disable its security completely or grant it excessive privileges to prevent the disruption.

CDs can also be used to provide some insight into the methods used by attackers to find and exploit weaknesses. There is some circumstantial evidence that some buffer overrun attacks have been discovered due to an interface exhibiting inadequate viscosity. In one example an attacker first tested an application's attack surface using exponentially increasing blocks of random data. They then observed which area gave the most dubious response and proceeded to probe that point for vulnerabilities, often with slightly mutated genuine data to test the viscosity at the boundary between legitimate and hostile data. Input points that exhibited high viscosity in their response to hostile data were likely to get passed over.

There is further work to be done applying CDs to attacker methodology.

## Conclusion

The application of Cognitive Dimensions to security may help to understand the nature of security vulnerabilities from a human perspective, possibly assisting with the process of threat discovery and mitigation.

## References

1. Green, T. R. G. & Petre, M. (1996) Usability analysis of visual programming environments: a 'cognitive dimensions' framework. *J. Visual Languages and Computing*, 7
2. Fluhrer, S., Mantin, I. & Shamir, A. (2001) Weaknesses in the key scheduling algorithm of RC4. Eighth Annual Workshop on Selected Areas in Cryptography.
3. Stubblefield, A. Ioannidis, J. Rubin, A. D. (2001) Using the Fluhrer, Mantin, and Shamir Attack to Break WEP. AT&T Labs Technical Report TD-4ZCPZZ
4. Goldberg, I. & Wagner, D. (1996) Randomness and the Netscape Browser. *Dr. Dobb's Journal*
5. Kocher, P. C. (1996) Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems, CRYPTO 1996
6. Wang, X., Feng, D., Lai, X. & Yu, H. (2004) Collisions for Hash Functions MD4, MD5, HAVAL-128 and RIPEMD
7. Brute-force searching DES, [www.distributed.net/des/](http://www.distributed.net/des/) (1997, 1998, 1999)
8. Howard, M. LeBlanc, D. (2002) *Writing Secure Code Second Edition*. Microsoft Press

## Glossary

### **XSS Attack** – Cross Site Scripting (XSS) Attack

An XSS attack uses a weakness in input processing on a website to allow the attacker to run scripts on the victim's machine in the website's vulnerable domain. This can be used to corrupt or steal user information. If the vulnerable domain has weaknesses in its authentication procedures it may also be used to impersonate the user. The underlying issue is that too much information is returned to the user, in a manner in which it can be exploited.

CERT issued advisory CA-2000-02 about XSS attacks.

### **Internationalisation Attack** – Attack based on the international features of a system.

For example, conversion between different character tables can result in buffer overrun attacks due to character length issues. Another example is that distinct characters can have the same visual appearance, leading a user to believe that they are the same. For example 'localhost' refers to the local loopback interface, however 'localhost' does not. (The o is Unicode: U+03BF, Greek small Omicron not U+006F as in the first example)

A sample vulnerability is noted in US-CERT vulnerability note VU#273262

### **Security API verbosity weakness** – Attacks based on information disclosure from a security sensitive API.

Some Security APIs contain vulnerabilities that allow sensitive information to be extracted from them. An example is the ATM vulnerability discussed in Cambridge technical report UCAM-CL-TR-560.

### **Spoofing Attack** – In a spoofing attack a user masquerades as a different user.

This may result in inappropriate trust allocations. An example is an attacker modifying their IP stack so as to make their messages appear to come from a different IP address.

CERT issued advisory CA-1996-21 about IP Spoofing Attacks.

### **Format String Attacks** – A class of attack that involves the use of format strings.

The weakness arises from C functions where a variable number of arguments may be passed. C is unable to determine the actual number of arguments passed. This allows memory state information disclosure through passing strings that would normally be used with additional arguments. In certain cases it can allow arbitrary memory overwriting.

An example of this vulnerability is noted in US-CERT Vulnerability Note VU#29823

### **Secrets Compromise** – A secrets compromise attack is the general classification for attacks that result in the extraction of information that should not have been revealed.

For example, a system that stores plaintext passwords in a database, rather than one-way salted hashes of passwords. These secrets could then be trivially compromised if the database was comprised.

### **'Hide extensions for known file types' weakness**

This is a feature in the default installation of several versions of Microsoft Windows, including Microsoft Windows XP. If a file ending is recognised by Windows, it is not displayed. This can result in vulnerabilities where an end user incorrectly believes that a file is of a 'safe type'. For example 'readme.txt.exe' will be displayed as 'readme.txt'.

CERT issued Incident Note IN-2000-07 commenting on exploits of this weakness, including the infamous VBS/LoveLetter worm.