GRESHAM COLLEGE
Founded 1597

Cybersecurity Transcript

Date: Tuesday, 3 May 2016 - 6:00PM

Location: Museum of London

03 May 2016

Cybersecurity

Professor Martyn Thomas

Introduction and Summary

Most of us have become used to receiving a daily stream of *spam* — unwanted email messages. Some of this will be legal, because marketing emails can be sent without prior consent by organisations who obtained your email address when you bought something from them if they are advertising similar products or services. However, these marketing emails must abide by strict rules regarding their content and provide you with the opportunity to opt out . Other spam may be legal because it is sent to an employee of a Limited company or Limited Liability Partnership who has not requested that the sender stops sending these messages. Where the spam is unlawful, it may still be harmless (except to the extent that it raises your blood pressure!) but an increasing amount is a *cyber-enabled crime* or a *cyberattack*. These two categories overlap, so I shall describe how I will use the terms within this lecture.

Cyber-enabled crimes are conventional crimes that have moved online, for example boiler room fraud , 419 letters , romance scams and Ponzi schemes . Cyberattacks are also crimes but they focus on gaining control of a computer system or something held on a computer, such as private data or passwords.

A cyberattack may target bank accounts or other financial assets.

Malware

Cyber-attack warning after millions stolen from UK bank accounts

Top crime agency delivers advice after virus used to access online banking details, with UK losses estimated to hit £20m

Vikram Dodd

Tuesday 13 October 2015 22.04 BST

Computers become infected with the virus when users receive and open documents in seemingly legitimate emails, the NCA said. Photograph: Brian Jackson/Alamy

It may target health records so that the data can be used for marketing, blackmail, or to target future frauds.

WEDNESDAY, AUGUST 5TH ABOUT SECURITY LEDGER OUR SPONSORS BECOME A SPONSOR CONTACT STAFF SUBSCRI

the security ledger

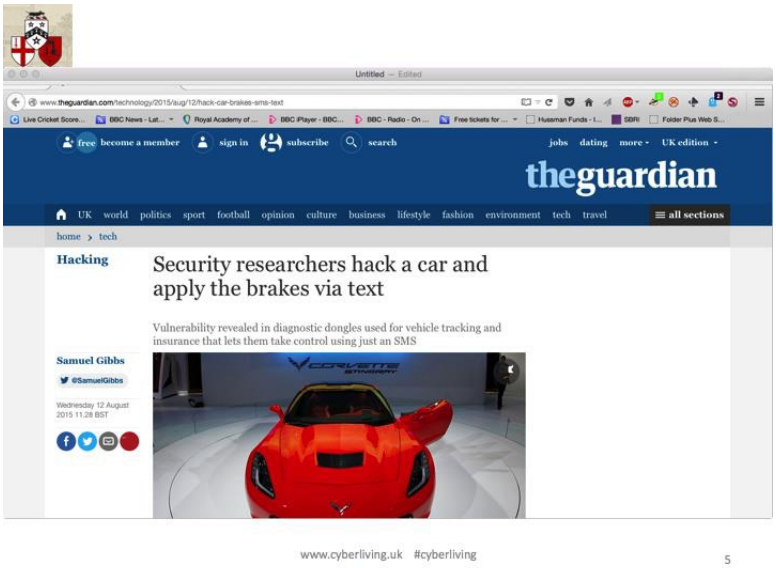
search...

INTERNET OF THINGS | CONNECTED CARS | THREATS | THOUGHT LEADERSHIP | PODCASTS | VIDEO

You are here: Home » Companies » Anthem Healthcare » Doctors Still In the Dark After Electronics Records Hack Exposes Data on 4 Million

Doctors Still In the Dark After Electronics Records Hack Exposes Data on 4 Million

It may seek to control physical machinery …



or to extort money by encrypting all the files on the computer.

In this lecture, I shall describe how some of these attacks work and how professional software developers protect their systems from some of the more common exploits, such as SQL injection and buffer overflows.

All knowingly unauthorised use of computer equipment is an offence under the *UK Computer Misuse Act 1990* and in the UK it carries a maximum penalty on summary conviction of a 12 month prison sentence or a fine or both. Any misuse of computers, software, networks or websites may also break end-user license agreements or acceptable use policies, with other consequences for offenders.

Basic information to help computer users to protect themselves from the most common cybercrimes and cyberattacks can be found on the CyberStreetWise and Get Safe Online websites.

Common ‘social’ attacks

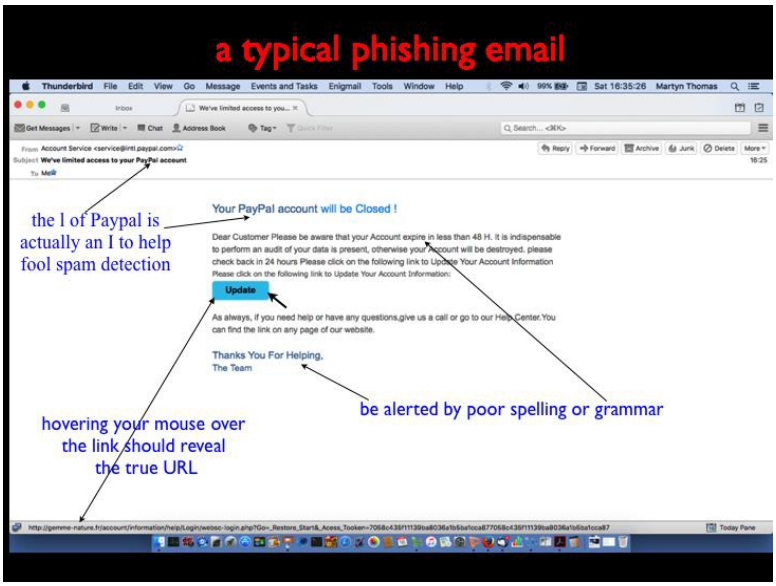
One class of cyber-attacks involves exploiting the gullibility or inattention of computer users or their desire to give higher priority to convenience than to security.

1. *Phishing emails*

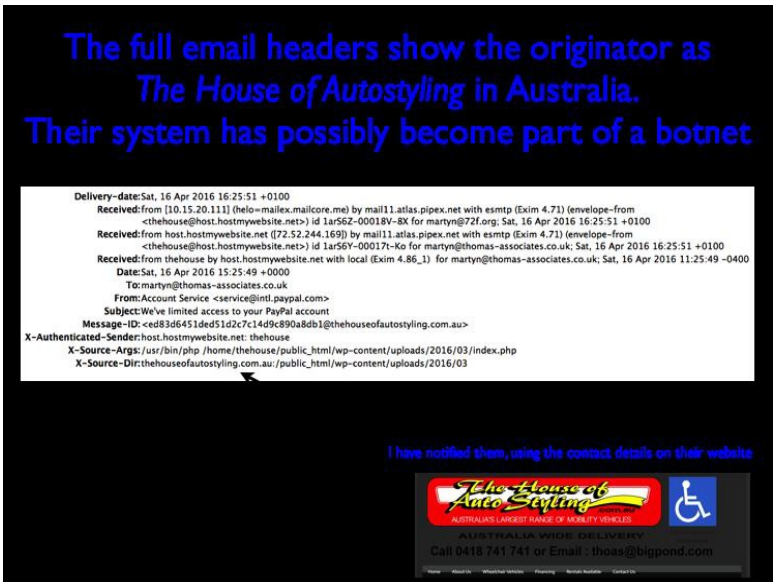
“Phishing” refers to emails that are designed to trick you into opening an attachment that contains some

malicious software (often in the form of an exe file, java or a Word or Excel file with embedded macros) or to click on a web link that leads to a web page that either downloads malicious software or that tricks you into revealing your account and password. It can be easy to copy a legitimate website and to modify it, so once the user has been led to the false site they are likely not to realise that they have been tricked.

Phishing emails can often be spotted if you are really careful. This is one that I received recently.



By viewing the email headers in my email reader, I could see where the email had probably come from.



It is easy to forge the “From” header in an email, so one should never trust this! But other headers are added automatically and may reveal the forgery.

This phishing attempt was easy to spot but criminals are becoming better at their fraud attempts. Increasingly, instead of “Dear Customer” emails, you will get emails that address you by your name and that contain personal details that make you think that the email is legitimate. This has been called “spear-phishing”, meaning targeted and personalised emails. The criminals often use data from hacked websites or from social media sites such as Twitter, Facebook and LinkedIn. Many users of these sites reveal their full names, address, date of birth, the company they work for, their friends’ and partners’ names, their pets, where they have been on holiday and much more. All of this is enormously helpful to a cyber criminal. Every time you reveal details about yourself or someone else, you increase the risks of cybercrime. This is one reason why the claim about privacy that “if you have nothing to hide, you have nothing to fear” is naïve at best. (I shall return to this topic next month in my lecture on *The Broken Promise of Anonymisation*).

A fraud that is growing recently involves sending phishing emails to staff in company account departments, purporting to come from their Finance Director or CEO and instructing them to make an urgent bank transfer to the fraudster’s account. The emails look legitimate, with company logos and other details — all of which are freely available online, of course. Finance staff are encouraged to check and recheck before taking action on any unusual or unexpected instruction.

2. *Hijacked email accounts and website accounts*

Criminals steal accounts in various ways and for various purposes.

Probably the easiest way to access an account is by guessing the password, because so many people use obvious ones. Among the most common are number sequences such as 1234, 12345678 or 11111111, keyboard wipes across such as qwertyuiop and shorter sequences, keyboard wipes downsuch as 1qaz2wsx and variants, and obvious words. A recent password "top 20" includes *pass, password, passw0rd, letmein, master, football, pussy, starwars, dragon, monkey,* and names, foods, colours, and makes of cars – probably the same make of car that the user owns and that they have revealed on Twitter and Facebook. Any real word may be vulnerable to a *dictionary attack*, where a website is hacked and the password file is stolen. If the passwords are encrypted using a common hashing method, the criminal can then try all the words in their dictionary, offline. If any of them generate hashes that match an entry in the password file, it reveals the password.

Many people use the same password for several accounts, so if one account is compromised and their password is revealed, their other accounts may be very easy to access. Using the same password for multiple accounts is very risky because it makes the security of all the accounts as weak as the weakest.

Phishing emails, leading to spoof websites as I described earlier, are the second most common way that criminals take over accounts.

The third most common way to take over someone else's account is probably through the *lost password* route if the website allows a new password to be set by answering security questions (online or over the telephone) because the answers are often readily visible in the victim's social media, and call centre staff have been shown to be very 'helpful' when wrong answers are provided.

A further way that criminals capture passwords is by setting up their own wireless access points in public areas and either capturing the data that flows across them or accessing the computers that connect. The criminal creates a wifi point, perhaps just by using a laptop computer, in a public area such as a station café. They name the network *free customer wifi* – perhaps incorporating the café name as well – and customers start to use it. The network traffic can be captured using widely available tools that have a legitimate use in analysing corporate networks . It is never a good idea to do something confidential over a public wifi.

There are many other social attacks. If you would like to discuss these, please ask questions or post comments on my cyberliving lecture series website.

Common technical attacks

Another set of attacks exploit weaknesses in the way that websites and other software has been implemented.

One of the commonest is the **buffer overflow attack** where an attacker exploits the fact that some software has been written to accept a string of characters (for example a username or password, or data packets over a communications link) but the programmer has omitted to check the length of the input before copying it into a fixed-length storage location; this allows the attacker to overwrite other locations in store and either cause a crash or, if the input string is very carefully constructed, to take control of the computer system.

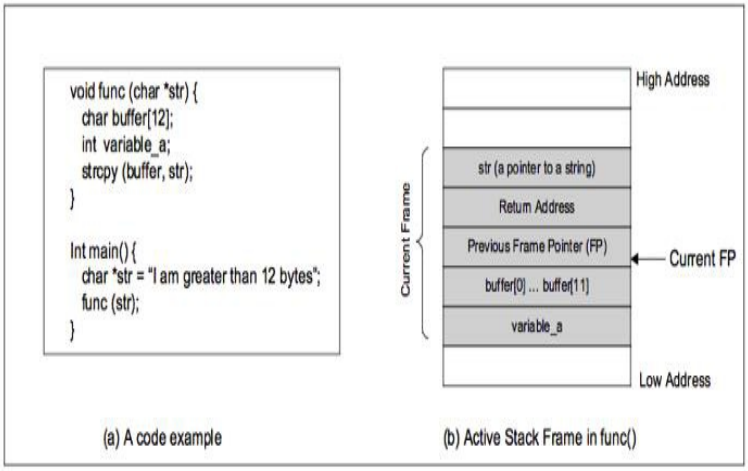
Buffer overflows often occur in software written in C or C++. Character strings in C are just sequences of characters terminates by a *null* character; there are safe ways to copy strings but some programmers simply uses the standard function *strcopy*, which is defined as follows:

strcpy char * strcpy (char *destination, const char *source);

Copies the C string pointed by source into the array pointed by destination, including the terminating null character (and stopping at that point).

To avoid overflows, the size of the array pointed by destination shall be long enough to contain the same C string as source (including the terminating null character), and should not overlap in memory with source.

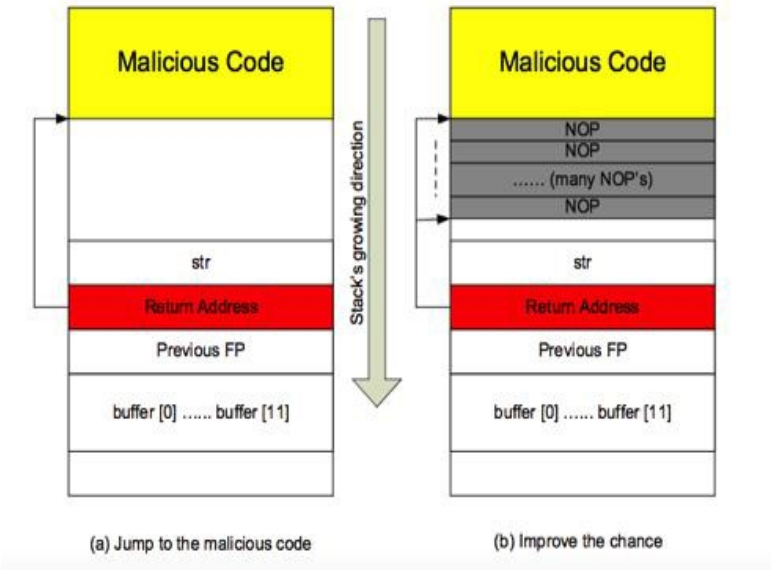
A "buffer" is a fixed length area of computer memory, typically defined as an array of characters. An example would be the memory for a username or password. Often, these will be held on the program stack, as shown below



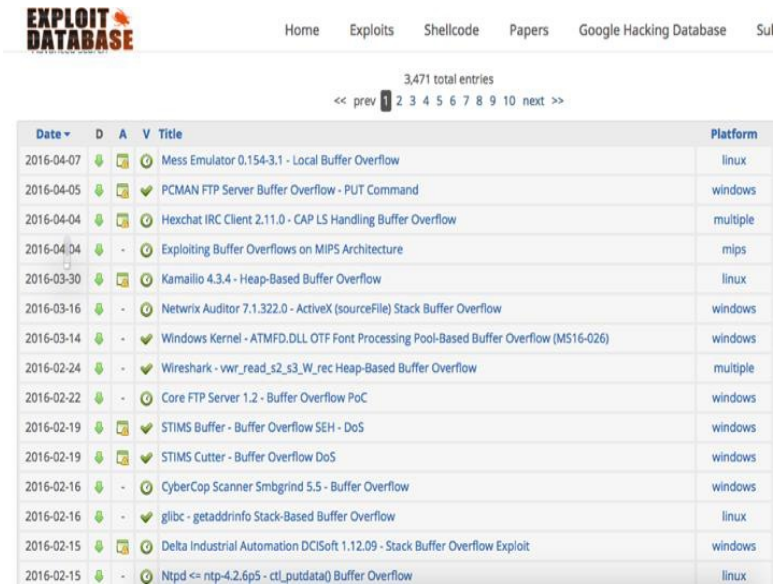
- Stack Direction: Stack grows from high address to low address (while buffer grows from low address to high address)

The buffer overflows up the stack, overwriting the other variable in the stack frame (if any) and potentially overwriting the *return address* that program control will be returned to when this function exits. This means that the attacker can construct a string that causes program control to jump to any location in memory that the attacker chooses; commonly this will be some malicious code that the attacker has also introduced (possibly in the same input string that caused the buffer overflow).

This takes careful design and it can be difficult to get the jump destination exactly right; the attacker will usually improve their chances by preceding the malicious code with a lot of *no operation* instructions so that if the jump targets any of these, control will quickly run through to the malicious code.



Buffer overflows really shouldn't exist in professional software as they are easily avoided, but they remain one of the common reasons why security patches have to be issued, as can be seen from this recent page from an exploits database.



EXPLOIT DATABASE

Home Exploits Shellcode Papers Google Hacking Database Sul

3,471 total entries

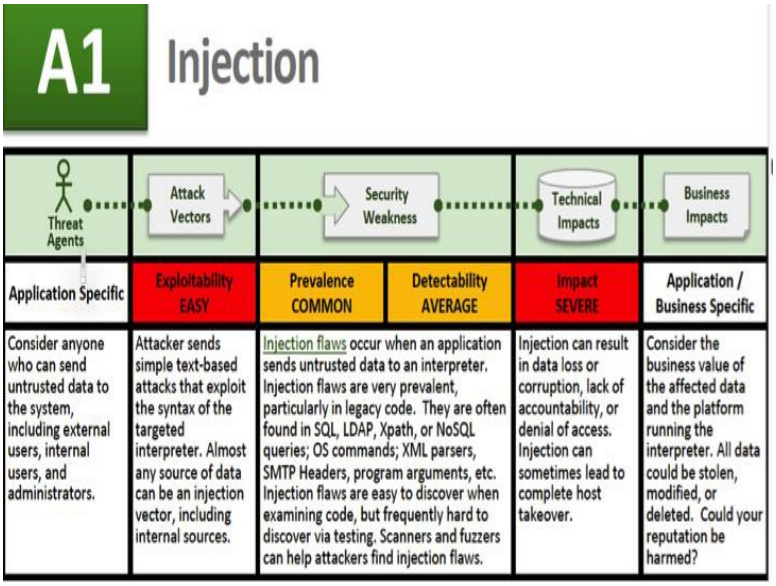
<< prev 1 2 3 4 5 6 7 8 9 10 next >>

Date	D	A	V	Title	Platform
2016-04-07				Mess Emulator 0.154-3.1 - Local Buffer Overflow	linux
2016-04-05				PCMAN FTP Server Buffer Overflow - PUT Command	windows
2016-04-04				Hexchat IRC Client 2.11.0 - CAP LS Handling Buffer Overflow	multiple
2016-04-04		-		Exploiting Buffer Overflows on MIPS Architecture	mips
2016-03-30				Kamailio 4.3.4 - Heap-Based Buffer Overflow	linux
2016-03-16		-		Netwrix Auditor 7.1.322.0 - ActiveX (sourceFile) Stack Buffer Overflow	windows
2016-03-14		-		Windows Kernel - ATMFD.DLL OTF Font Processing Pool-Based Buffer Overflow (MS16-026)	windows
2016-02-24		-		Wireshark - vwr_read_s2_s3_W_rec Heap-Based Buffer Overflow	multiple
2016-02-22		-		Core FTP Server 1.2 - Buffer Overflow PoC	windows
2016-02-19				STIMS Buffer - Buffer Overflow SEH - DoS	windows
2016-02-19				STIMS Cutter - Buffer Overflow DoS	windows
2016-02-16		-		CyberCop Scanner Smbgrind 5.5 - Buffer Overflow	windows
2016-02-16		-		glibc - getaddrinfo Stack-Based Buffer Overflow	linux
2016-02-15				Delta Industrial Automation DCISoft 1.12.09 - Stack Buffer Overflow Exploit	windows
2016-02-15		-		Ntpd <= ntp-4.2.6p5 - ctl_putdata() Buffer Overflow	linux

The next class of attacks that I want to discuss are **insertion attacks**, and in particular **SQL Insertion**. This is reported to have been the attack underlying the data breach at the communications company TalkTalk that was in the news earlier in 2016.

Insertion attacks work by crafting an input string (a username for a website, for example) in such a way that it causes the attacker's instructions to be passed to some software that interprets and executes the instructions.

The Open Web Application Security Project lists injection attacks as the most serious vulnerability in 2016. Their overview explains why:



A1 Injection

Threat Agents	Attack Vectors	Security Weakness		Technical Impacts	Business Impacts
Application Specific	Exploitability EASY	Prevalence COMMON	Detectability AVERAGE	Impact SEVERE	Application / Business Specific
Consider anyone who can send untrusted data to the system, including external users, internal users, and administrators.	Attacker sends simple text-based attacks that exploit the syntax of the targeted interpreter. Almost any source of data can be an injection vector, including internal sources.	Injection flaws occur when an application sends untrusted data to an interpreter. Injection flaws are very prevalent, particularly in legacy code. They are often found in SQL, LDAP, Xpath, or NoSQL queries; OS commands; XML parsers, SMTP Headers, program arguments, etc. Injection flaws are easy to discover when examining code, but frequently hard to discover via testing. Scanners and fuzzers can help attackers find injection flaws.		Injection can result in data loss or corruption, lack of accountability, or denial of access. Injection can sometimes lead to complete host takeover.	Consider the business value of the affected data and the platform running the interpreter. All data could be stolen, modified, or deleted. Could your reputation be harmed?

The basic issue is this. A website asks for a username, looks up some data about that user and uses or displays it.

The query to the database may look like this – **SELECT passwd FROM USERS WHERE uname IS '$user'** – where the variable $user contains the username string typed in by the attacker.

In the normal case, this query might be

SELECT passwd FROM USERS WHERE uname IS 'Thomas'

but if I type my username as **';DROP TABLE USERS; --**

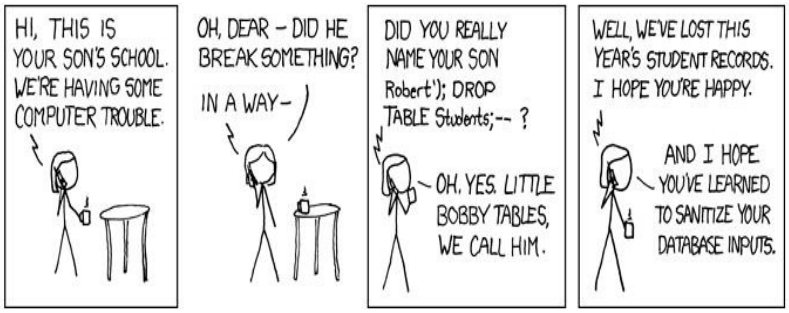
then if the programmer has taken no precautions, the query will become

SELECT passwd FROM USERS WHERE uname IS '';DROP TABLE USERS; --'

Which will select a null username, then delete the entire USER table from the database. (The double minus at the end causes everything that follows to be ignored).

The same trick can be used with other SQL commands to select all the users and change their passwords, for example.

The exploit is so well known that it has become an internet joke



Injection attacks are potentially possible wherever user input is user to build a string that is passed to an interpreter. In the SQL case, a single quote and semicolon was able to terminate one query and start another, in other circumstances, a different character – a null character perhaps ¬may be used to terminate an input string. It all depends on the syntax of the website or other software.

There are many other common attacks. The OWASP top ten for 2013 are

A1 Injection

A2 Broken Authentication and Session Management

A3 Cross-Site Scripting (XSS)

A4 Insecure Direct Object References

A5 Security Misconfiguration

A6 Sensitive Data Exposure

A7 Missing Function Level Access Control

A8 Cross-Site Request Forgery (CSRF)

A9 Using Components with Known Vulnerabilities

A10 Unvalidated Redirects and Forwards

The excellent OWASP website describes each of these and provides guidance on how to avoid these risks in your systems.

All the vulnerabilities can be avoided by professional software engineering. They are not inevitable and software customers should not be exposed to the cybersecurity risks that result. Of course, it is the criminals who are primarily responsible for committing cyber crimes but the people who manufacture and sell products that contain software have a responsibility to sell secure products. If a lock manufacturer sold locks which were easy to defeat they would soon go out of business, because crime is to be expected and a lock that can easily be defeated is not fir for purpose.

The same is true for software. Cybercrime is to be expected, and insecure software is not fit for purpose, whether it is website software or the control systems for an oil refinery or a driverless car.

Please share your opinions and knowledge, and join in the discussions on www.cyberliving.uk.