Network Security and Resilience / Advanced Security

Threats – overview

Lecture four

Outline of lecture

- Taxonomy of attacks and methodology
- Notable exploits past ten years
- Basic attacks

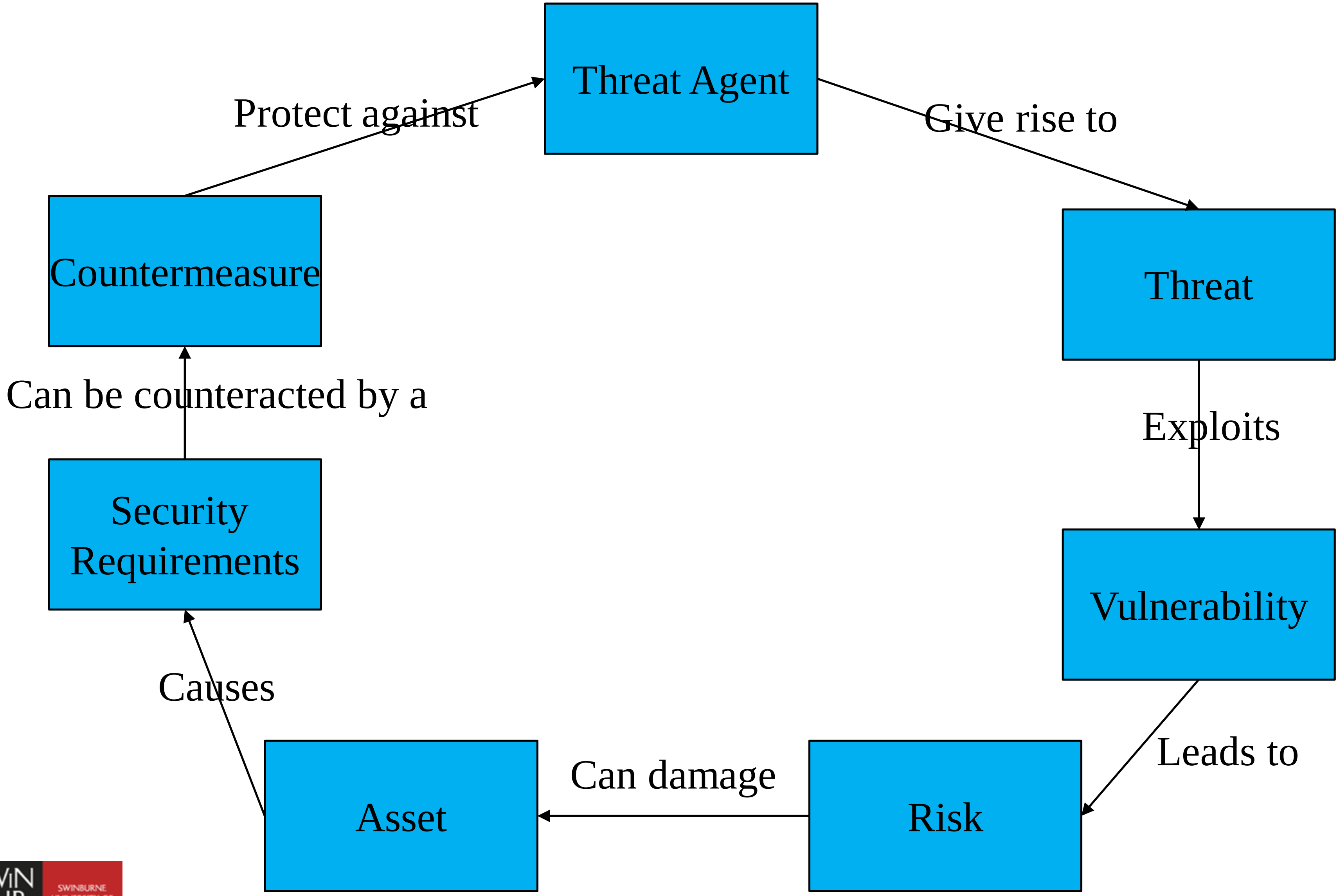
Taxonomy of attacks

- Vulnerability
  - A software, hardware or procedural weakness that may allow a threat agent to obtain unauthorised access to resources
- Threat
  - Any potential danger to resources
- Threat agent
  - An actor – human, programmatic or natural – that will act to increase the threat
- Risk
  - The probability of a threat agent, discovering and exploiting a vulnerability

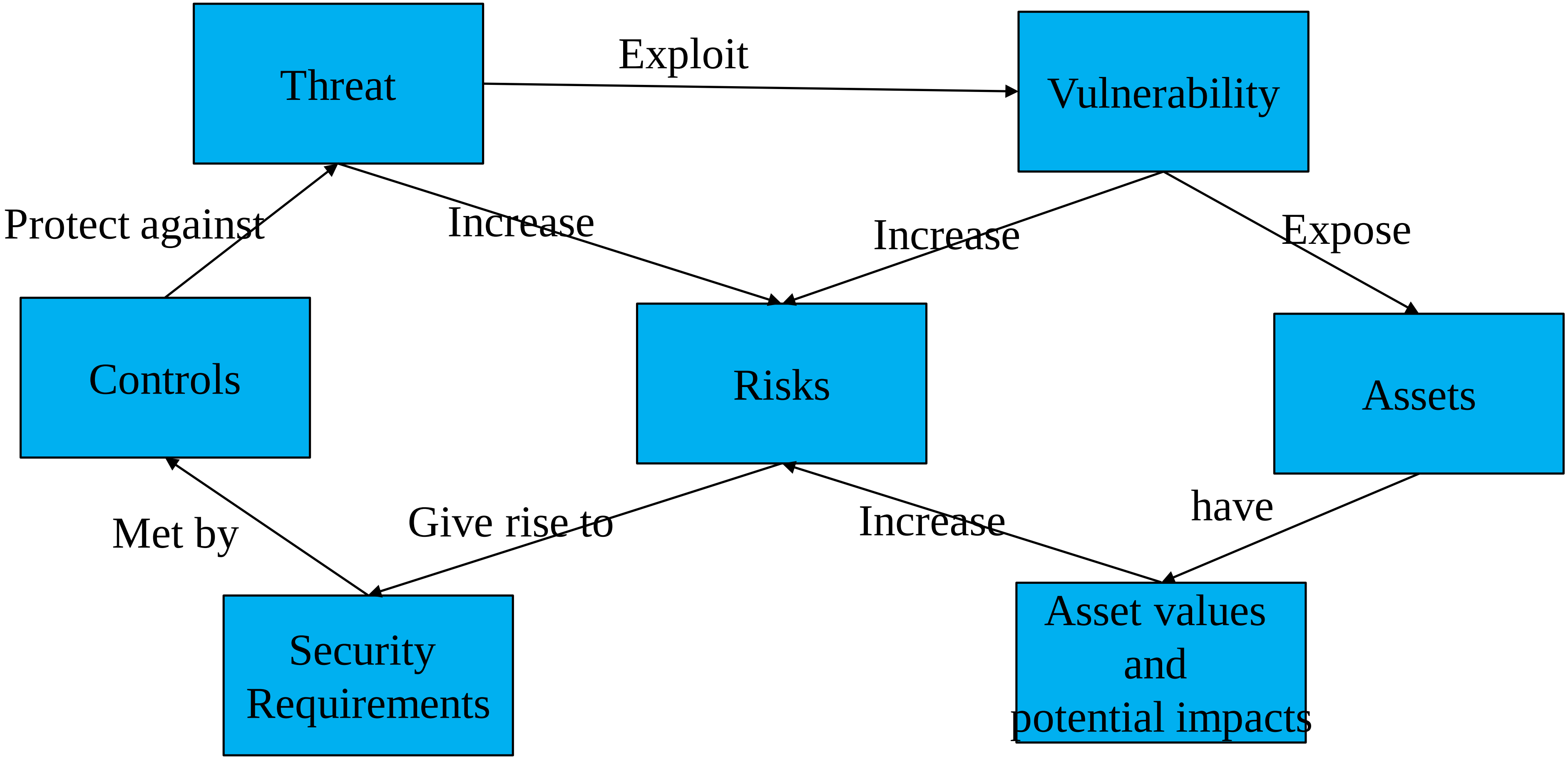
Taxonomy of attacks

- Exposure
  - An instance of risk of loss from a threat
- Countermeasures ore controls
  - A mechanism to eliminate or limit vulnerability

Taxonomy of Attacks



Taxonomy of Attacks



Vulnerability types

- Design errors
- Protocol weaknesses
- Software weaknesses
- Misconfiguration
- Hostile code
- Human factors

Adversaries

- Foreign states
- Terrorists
- Criminals
- Hackers
- Corporate competitors
- Government agencies

Classic attack structure (Cisco)

- Cisco suggest the following as a typical attack sequence
  - Reconnaissance
  - Identification of operating systems and applications
  - Access (via social engineering)
  - Escalate privileges
  - Gather additional passwords and secrets
  - Install backdoors
  - Exploit compromised host
- Of course lots of variations, but a useful framework that allows us to understand the role of different attack systems

Building blocks of attacks

- Buffer overflow
- Malware
- Backdoors
- SQL injection
- HTML
- Password attacks

Buffer overflow

- An important attack mechanism
  - Basis of some worm attacks
- A program with inadequate array bounds checking can be vulnerable to this attack
  - Commonly used where strings are passed as parameter
- Computer memory is organised in contiguous blocks executable code, data and stack space
  - Stack space contains return addresses from subroutines
- If data passed to the routine exceeds the expected size, and there is no array bound checking, it can overwrite the stack and substitute a new return address

Buffer overflow

- With some clever programming the return address can be to a routine that contains some executable code
  - For example, starts a shell script
- C programs particularly vulnerable
  - Poor array bounds checking
  - Can access pointer values in code
  - Windows and Unix mostly written in C
- Classic article on topic is “Smashing the stack for fun and profit”
  - http://inst.eecs.berkeley.edu/~cs161/fa08/papers/stack_smashing.pdf

Buffer overflow

```
void function(char *str) {
   char buffer[16];
   strcpy(buffer,str);
}
void main() {
  char large_string[256];

  for(int i = 0; i < 255; i++)
    large_string[i] = some clever string;

  function(large_string);
}
```

‘Malware’

- Malicious code
- Propagated through the Internet
  - Self-propagating or propagated through other applications
- Traditional classification is
  - Viruses, worms, trojan horses
  - But a lot of blurring
  - Malware may have characteristics of all three
- Useful source of information is the United States Computer Emergency Readiness Team (CERT) vulnerabilities database
  - http://www.cert.org/ and http://www.kb.cert.org/vuls

Viruses

- A virus is a small application or piece of code that infects other applications or codes
  - It cannot replicate on its own.
  - It uses an infected host to replicate and spread
- The virus vector is the mechanism by which the virus spreads
  - USB memory sticks
  - Word and Excel Macros
  - Floppy disk boot sectors (now quite rare)
  - Downloaded or emailed executables

Viruses

- Capabilities
  - Erase files on your machine
  - Delete directory structures
  - Encrypt files making them impossible for you to access (a Denial of Service mechanism)
  - Copy and send files on your machine
  - Send files to emails in your address book
  - Load logic bombs
  - Display inappropriate message
  - etc.

Virus vectors

- Early viruses were spread mainly on floppy disks.
  - Main way of information exchange on early personal computers
  - Viruses could be spread by infecting exchanged programs or were installed in the disk boot sector and executed at start up
- Online bulletin boards became main way of exchange in late 80s and early 90s
  - Viruses embedded in popularly traded software
- From mid-1990s main kinds of virus became macro viruses written in the scripting languages of Microsoft programs such as Word and Excel
- New viruses based on USB flash drives (memory sticks)

Virus signatures

- A unique string of bits or the binary pattern of a virus
- Consist of sequences of bytes in the machine code of the virus
  - Similar to a fingerprint
- Usually many candidates for virus signatures
  - Goal of those writing systems to identify and deal with viruses is to minimize false negatives and false positives
  - Good signature is one found in every object infected by the virus but is unlikely to be found if the virus is not present;
- Usually obtained by manual inspection
  - slow and error prone
  - Some work being done on automatic extraction

Virus coding

- Easy to code viruses
  - ‘script kiddies’
- Can download viruses from websites
- Many viruses script based
  - Use Visual Basic
  - Embedded in EXCEL or WORD macros
- Often new viruses are derived from the notification of security weaknesses
  - A company identifies a weakness in its software and posts a patch
  - Usually notifies US CERT
  - The virus writer exploits the weakness in the (reasonable) assumption that many users won’t install the patch

Worms

- A computer worm is a self-replicating computer program
- Self-contained and does not need to be part of another program to propagate itself
- Often designed to exploit the file transmission capabilities
- A worm uses a network to send copies of itself to other systems and it does so without any intervention
- Simple worms ‘only’ harm the network and consume bandwidth, whereas viruses infect or corrupt files on a targeted computer
- But occasionally malware takes on both worm and virus characteristics

Worms

- Email and Instant messaging worms
  - Do not infect files, but propagate by a file transfer system
    - eg email attachments
- File sharing worms
  - Exploit peer-to-peer systems
  - Innocuous named file located in a shared folder
- Network aware worms
  - Exploit security vulnerabilities such as unprotected shared drives, FTP weaknesses etc, usually by forcing a buffer overflow
  - Earliest examples of worms exploited buffer overflow

Examples of worms

- Morris Worm (more in next lecture)
  - First known worm (1988)
  - Unix based (BSD and derivatives)
  - Exploited buffer overflows in sendmail, finger and rsh
- WANK worm (1989)
  - The first known ‘political’ worm
  - Attempted to attack VAX machines at NASA
  - Refer to “In the Realm of the Hackers” (http://www.abc.net.au/tv/documentaries/stories/s853348.htm)
- Ramen worm (2001)
  - First known Linux worm
  - attacks Remote Procedure Call (RPC) service or ftp daemon
  - searches for vulnerable machines to propagate to

Examples of worms

- Code-Red worm (2002)
  - A particularly nasty IIS worm
  - Attacked 359,000 machines in 14 hours (peaked at 2000/minute)
- Blaster worm (August 2003)
  - A malicious worm
  - Exploited a buffer overflow weakness in Microsoft DCOM architecture
  - Intended to do a SYN flood attack on Microsoft site windowsupdate.com
  - A distributed denial of service attack
    - Author went to prison for 18 months

Examples of worms

- Welchia worm (August 2003)
  - A ‘good’ ish worm
  - The Welchia worm exploited a vulnerability in the Microsoft RPC service
  - it tried to help the user by downloading and installing security patches from Microsoft
  - Still causes lots of traffic, rebooted user’s machine and operated without user’s consent
- Consensus of security community is that all worms are bad

Examples of worms

- Mydoom (January 2004)
  - email worm
  - The mail contains an attachment that, if executed resends the worm to email addresses found in local files such as a user's address book
  - Two versions Mydoom.A and Mydoom.B
  - Mydoom.A allowed a backdoor into the victim’s computer with the aim of a Distributed Denial of Service Attack on SCO
  - Mydoom.B targets also blocks access to Microsoft website
- Unnamed myspace social networking worm (2007)
  - Propagated through myspace list of friends
- Conficker
  - A computer worm that spreads itself to other computers across a network or via USB without human interaction (from Microsoft.com)

Trojan Horses

- Simple Trojan Horse
  - Some inviting file name (use your imagination) with .exe suffix
  - User clicks on it
  - Program runs and (for example) deletes all files on c:\
  - Made easier by some Microsoft systems (eg. Microsoft Outlook Express) hiding file extension
    - eg. annakournikova.jpg.exe appears as annakournikova.jpg
- More sophisticated Trojan Horses
  - Allows remote user to control victim's machine probably without victim being aware of it
  - Victim's machine becomes a 'server' that responds to the attacker's 'client'
  - Often associated with 'rootkits', software that has root privileges and hides its presence

Ways that Trojan Horses can be used

- erasing or overwriting data on a computer
- corrupting files in a subtle way
- spreading viruses or worms
- setting up networks of zombie computers in order to launch DDOS attacks or send spam
- spying on the user of a computer and covertly reporting data like browsing habits to other people
- logging keystrokes to steal information such as passwords and credit card numbers
- phishing for bank or other account details, which can be used for criminal activities.
- installing a backdoor on a computer system.

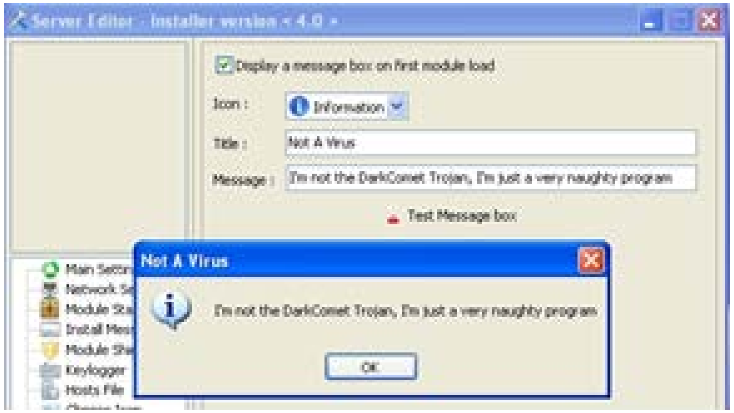
Examples of Trojan Horses

- Back Orifice 2000 (BO2K)
  - Windows based backdoor system
  - Client (attacker) can
    - run files on victim’s machine
    - log keystrokes on victim’s machine
    - restart or lock victim’s machine
    - read any files on victim’s machine
- PKZIP3
  - attempts to reformat harddrive
  - poses as newer version of popular software

Examples of Trojan Horses

- Dark Comet Remote Administration Tool
  - Can be used for legitimate purposes, and is promoted by its developers as such, but can also be used for taking control of a user's computer without their consent or knowledge
  - Can do key logging, send messages and run programs (such as open a webbrowser) on the victim's machine

(used to be at www.darkcomet-rat.com but now a dead link)



Server Editor - Installer version < 4.0 >

☑ Display a message box on first module load

Icon : Information

Title : Not A Virus

Message : I'm not the DarkComet Trojan, I'm just a very naughty program

Not A Virus

I'm not the DarkComet Trojan, I'm just a very naughty program

Backdoor

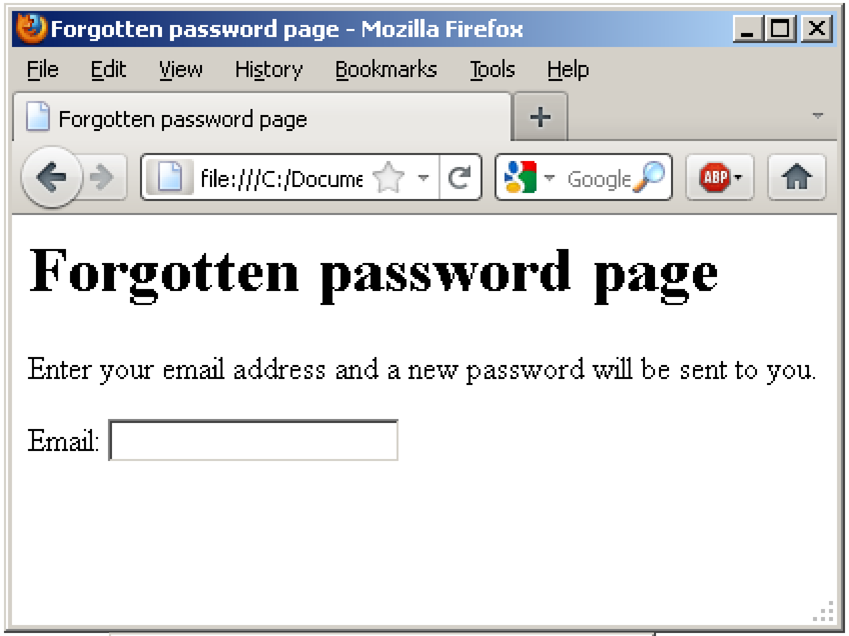
- A backdoor in a computer system is a method of bypassing normal authentication while remaining hidden from casual inspection. The backdoor may take the form of an installed program or could be a modification to a legitimate program.
- A backdoor in a login system could take the form of a hard coded user and password combination which gives access to the system
  - A famous example was an attempt to plant a backdoor in the Linux kernel in November 2003
  - a two-line change appeared to be a typographical error, but actually gave the caller to the sys_wait4 function root access to the system

HTML attacks

- Drive-by downloads
  - Website causes a trojan (typically) to be loaded onto the victim’s computer without his or her knowledge
- Cross-site scripting
  - Makes use of mixing of control and data information in HTML documents
  - Typically used to exploit credentials from the victim’s site (eg logged into gmail)

SQL injection

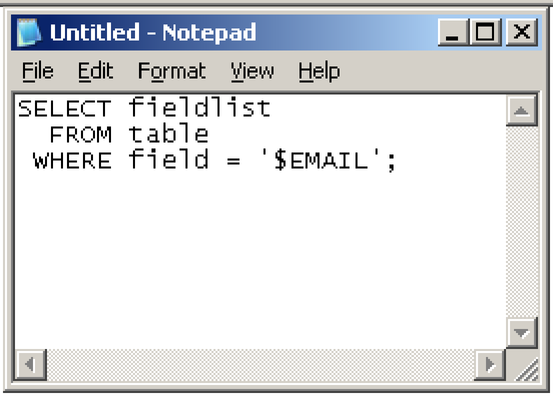
- A commonly used technique where a database is accessed via a webpage
  - – An example of poor input checking
- The input fields on a webpage are used to construct a query string of the full database
- SQL injection attacks can be prevented by verifying the input into the SQL database
- If an attacker were to enter
- x' OR full_name LIKE '%Bob% it would return details of everyone with ‘Bob’ in their name



Forgotten password page

Enter your email address and a new password will be sent to you.

Email:



```
SELECT fieldlist
  FROM table
 WHERE field = '$EMAIL';
```

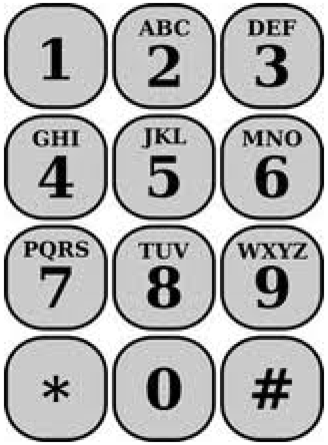
http://www.unixwiz.net/techtips/sql-injection.html

Password attacks

- Passwords are the most common authentication technique but are actually quite a weak way of proving identity
- There are a number of problems with passwords
  - People tend to use the same password across multiple systems
  - People tend to use passwords that are easy to remember
  - People tend not to change their passwords
- Some commentators believe passwords should no longer be used in authentication

Password attacks

- Some passwords are used very frequently
  - 1234, abcd, pass
- PINs are often keypad sequences that are easy to remember
  - 2580, 0852, 8879

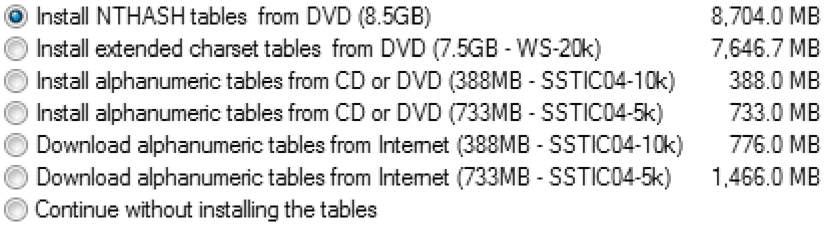


Password cracking

- It is very bad cryptographic practice to store a password in plaintext.
- Usually the hash (or multiple hash) of the password with some additional noise (salt) is stored rather than the password itself
- A hash is a one-way cryptographic function
  - Knowing the hash of a value tells you nothing about the original value
  - Well known hash functions are SHA-1, SHA-256 etc
  - A hash function takes any bit sequence and produces a fixed length ‘hash’
  - SHA-1 produces 160 bit hashes, SHA-256 produces 256 bit hash
- Login files contain only the hash of the password+salt

Password cracking

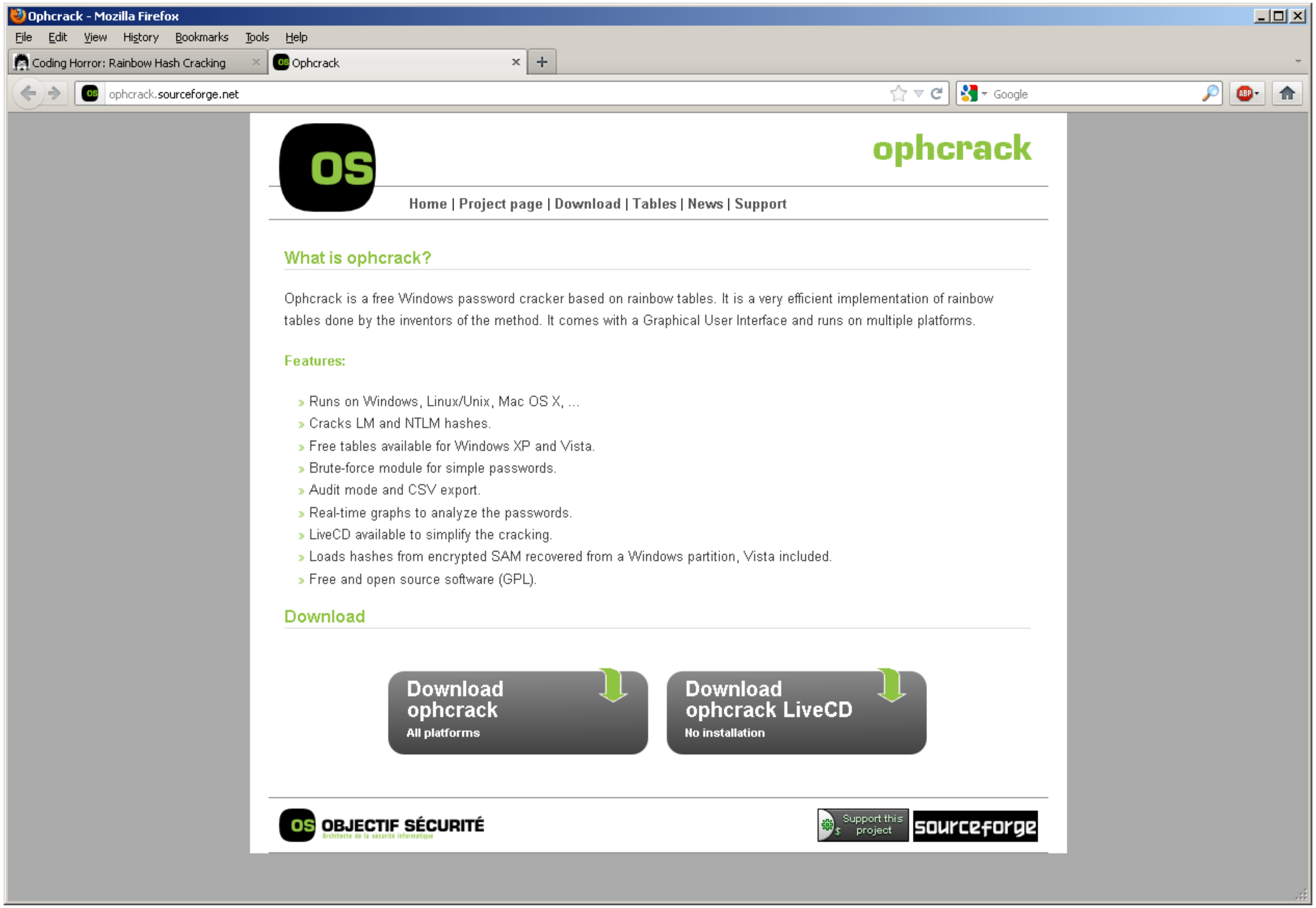
- Rainbow tables enable a hacker to lookup a hash value to find the corresponding plain text – the original password
  - Example “ophcrack”
- Rainbow tables can be very large
- Constructing them takes a great deal of computation
  - But once constructed, every attacker can use them
  - Schneier reports that there are multi-terabyte Rainbow tables



● Install NTHASH tables from DVD (8.5GB)	8,704.0 MB
○ Install extended charset tables from DVD (7.5GB - WS-20k)	7,646.7 MB
○ Install alphanumeric tables from CD or DVD (388MB - SSTIC04-10k)	388.0 MB
○ Install alphanumeric tables from CD or DVD (733MB - SSTIC04-5k)	733.0 MB
○ Download alphanumeric tables from Internet (388MB - SSTIC04-10k)	776.0 MB
○ Download alphanumeric tables from Internet (733MB - SSTIC04-5k)	1,466.0 MB
○ Continue without installing the tables	

From http://www.codinghorror.com/blog/2007/09/rainbow-hash-cracking.html

Password cracking


Ophcrack - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Coding Horror: Rainbow Hash Cracking
Ophcrack
ophcrack.sourceforge.net
Google
OS
ophcrack
Home | Project page | Download | Tables | News | Support

What is ophcrack?

Ophcrack is a free Windows password cracker based on rainbow tables. It is a very efficient implementation of rainbow tables done by the inventors of the method. It comes with a Graphical User Interface and runs on multiple platforms.

Features:

- Runs on Windows, Linux/Unix, Mac OS X, ...
- Cracks LM and NTLM hashes.
- Free tables available for Windows XP and Vista.
- Brute-force module for simple passwords.
- Audit mode and CSV export.
- Real-time graphs to analyze the passwords.
- LiveCD available to simplify the cracking.
- Loads hashes from encrypted SAM recovered from a Windows partition, Vista included.
- Free and open source software (GPL).

Download

Rainbow tables

- A rainbow table does not store every possible password and hash combination
  - What is stored are the start and endpoints of a sequence of hash and reduction functions
- The reduction function takes a hash value and maps it to a valid plaintext
  - It is not the inverse of the hash
- When the attacker wants to find the plaintext that matches the hash they apply the reduction function / hash function repeatedly until the hash is found. That means the plaintext is within that sequence.
- Start from the beginning of the sequence and apply hash / reduction until hash found

Rainbow tables

- Good overview and example at http://stichintime.wordpress.com/2009/04/09/rainbow-tables-part-5-chains-and-rainbow-tables/
- Most common defences are ‘salting’ passwords and passphrases and multiple hashing of hashes
  - Include an additional (not necessarily secret) string called the “salt” that makes string being hashed much longer and so much harder to find
  - Eg hash(password+salt) or hash(hash(password)+salt)
  - The salt might be stored in a database or it might simply be a function of the password itself such as a hash
  - Article on hack of system in which passwords were hashed but not salted. http://theconversation.edu.au/the-abcs-website-has-been-hacked-but-how-12522

Social Engineering

- Social engineering
  - obtaining confidential information by manipulation of legitimate users
- Exploits natural tendency of a person to trust others when talking to them
- Makes use of the weakest link in any security system
  - People
- Simplest attack is tricking user into thinking one is an administrator and requesting a password or credit card
  - eg phishing attacks.
  - Unsolicited surveys of ‘customer satisfaction’
  - Phone raffles

Social Engineering

- Dealing with Social Engineering attacks should be part of the organisation’s security policy
  - Prevention
  - Detection
  - Response
- Good summary in http://www.securityfocus.com/infocus/1533
- Well worth having a look at http://www.scamwatch.gov.au for examples of other social engineering attacks

Exercises

- Classify the following attacks:
  - An email purportedly from a well-known bank asking you to re-enter your account details (including PIN) to confirm the validity of your account
  - An Excel spreadsheet attachment to an email that when you open it, attempts to format your hard drive
  - A friendly phone call from your phone company doing a short interview on your satisfaction with the service who, a few questions into the interview, asks for your password so they can enter the details into your account

Conclusion

- This lecture gives an overview of the building blocks of attacks. We will look at specific attacks, particularly network based ones, in the next few lectures
- Worth noting that although there are always new attacks they are usually some variation of the above