

# Ethical Hacking and Cyber Security Fundamentals

---

EHAC | Lecture 1

Dr. Roberto **Metere**

✉ roberto.metere@york.ac.uk

# Objectives

At the end of this lecture you will know about...

---

- ▶ what is Ethical Hacking (what is this course mainly about)
- ▶ fundamentals of cyber security
- ▶ classification of hacker's attitudes and actions

# Let's break the ice

- ▶ Does any of you hold a certification related to ethical hacking?
- ▶ Does any of you have participated to a capture-the-flag cybersecurity competition?
- ▶ Have you completed the modules CTAP and NETS?
- ▶ How strong are you on the command line?

Terminology...

- ▶ What is security?
- ▶ What is a hacker?
- ▶ What *makes* an attacker?



<https://www.menti.com/ale5qkae4s1r>

# Terminology | Security

0 1/3

process of mitigating threats to protect valuable resources

12

The goal of making data and functionality so expensive to access or break that it's not worth the cost

12

Protecting assets from unauthorised access

9

Confidentiality, integrity, authenticity

9

Protection of assets and data

8

Protection against adversaries

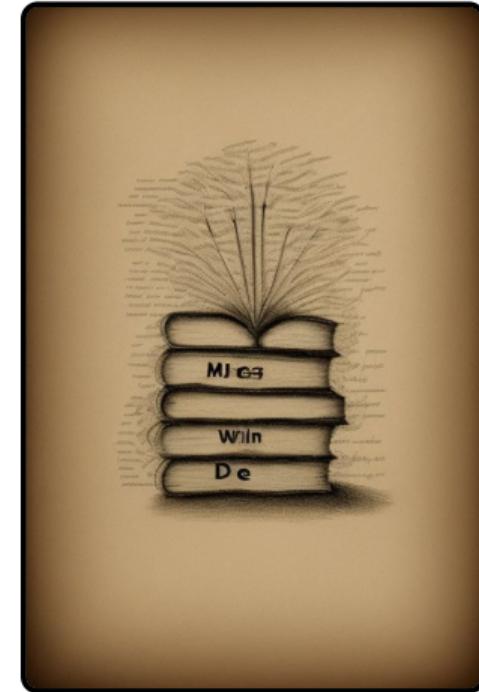
11

The Salvos bouncers

10

A series of items and procedures designed to make it harder for an attacker to achieve an aim that you don't want them to

7



# Terminology | Security

Q 2/3

A very broad term

7

Protection against unauthorised access

6

risk management in order to ensure business function

4

A system being difficult/ expensive to use maliciously

3

Security is a way for people to feel safe when they actually aren't :)

6

Maintaining integrity of a system or data

4

To protect something whether it's digital or physical.

3

The prevention of attacks and penetrations of physical, digital, political, economic or other systems.

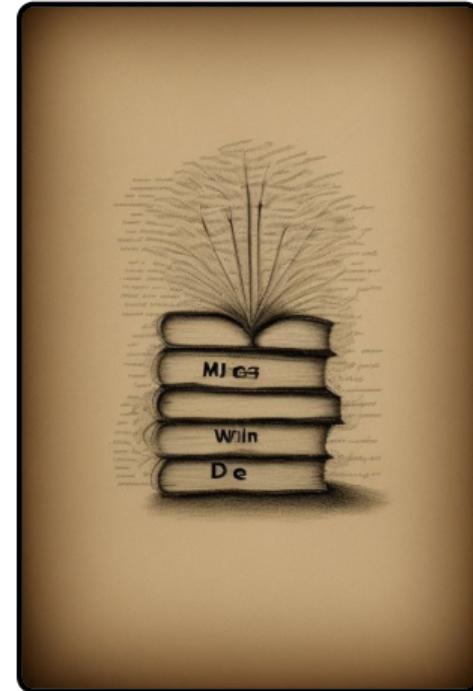
3

protecting data

5

Safety

2



# Terminology | Security

O 3/3



Department of Computer Science

not getting hacked..?

2

Protection against adversary

2

Legal practice of protection

1

Making something unfeasible to access

1

Boundaries

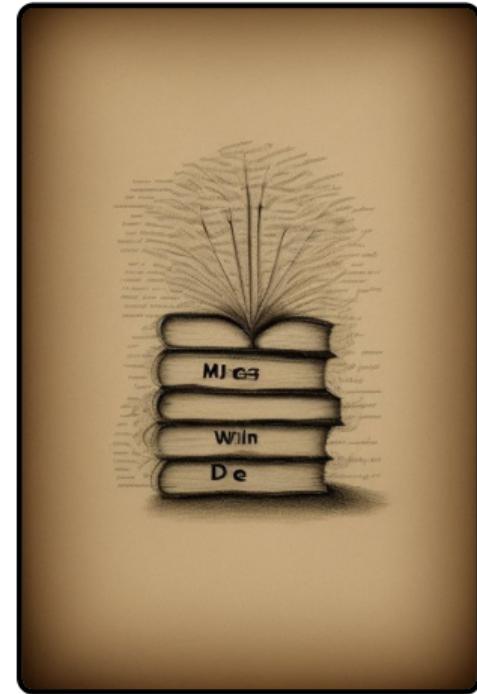
1

Banned unauthorized access

“ The protection of information against being stolen or used wrongly or illegally

”

from the Cambridge Dictionary



# Terminology | Hacker

0 1/5



Department of Computer Science

A malicious actor attempting to cause harm to a system or network.

10

someone who looks to exploit vulnerabilities in a system

10

A person / adversary exploiting vulnerability in a system to gain access to data or some sort of thing for some benefits.

10

Can you close the door please, it's cold

9

someone who gains unauthorised access to a system or resource

7

Individual who has skills to access systems without authorization and able to cause harm to organization or state for personal, command of state or opponent organization.

7



# Terminology | Hacker

0 2/5



Department of Computer Science

A hacker is a person skilled in information technology who achieves goals by non-standard means.

6

Mr  
Robot

4

A user who tries to use a system in a way that is undesirable to the owner by exploiting vulnerabilities

4

A person who tries to gain unauthorised access to a system

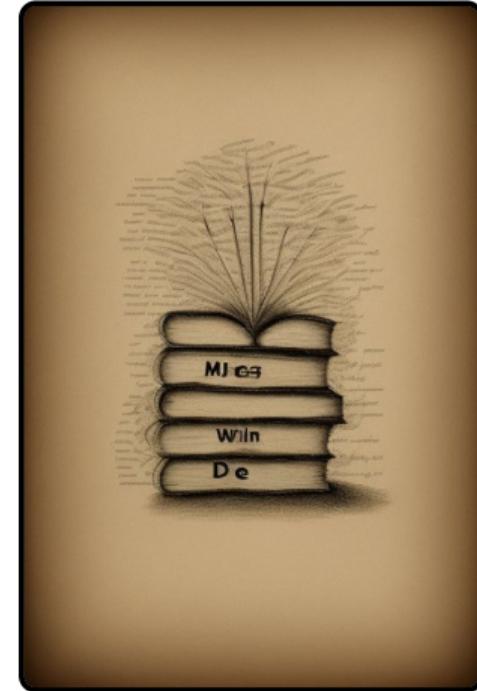
5

someone who has cool stickers on their laptop

5

A person that is capable of bypassing (cyber)security measures to obtain unauthorized access to data/resources

4



# Terminology | Hacker

Q 8/5

a person who has unauthorised access to a system

3

Doing things with a computer you're not supposed to

3

the guy who types really fast in movies

3

A person who uses a system or function in manner in which it was not intended to be used

3

Script Kiddie

3

Exploiting vulnerabilities in a system for personal gain or with explicit permission

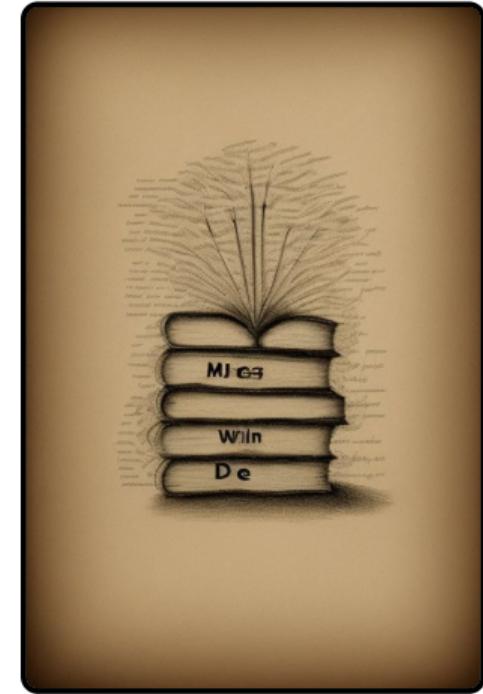
3

An entity that gains unauthorised access to a system.

3

An individual or group that manipulate vulnerabilities to breach systems and gain information either for good or bad purposes.

3



# Terminology | Hacker

O 4/5

someone who attempts to gain access to a system

2

Someone with bad intentions

1

An adversary

2

A very broad term

1

Someone who gains unauthorized access

someone who delights in understanding how things work, and in finding creative solutions to problems

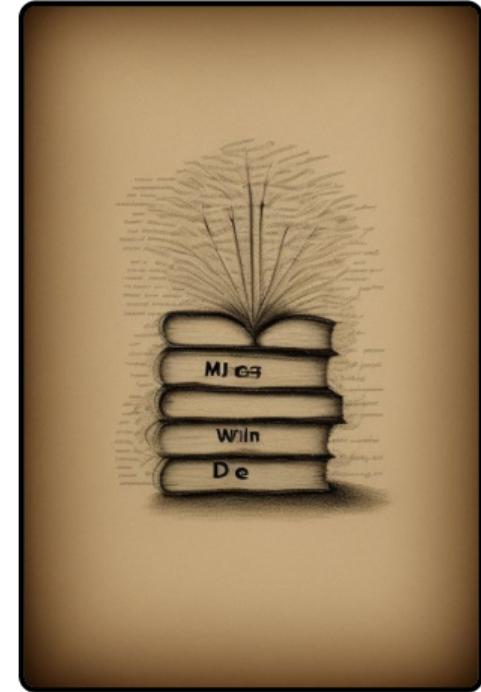
2

Any one who attempts to unauthorisedly access a system, comprise a systems security or shut down a system.

1

Who enjoys disrupting services

A vigilante



# Terminology | Hacker

O 5/5



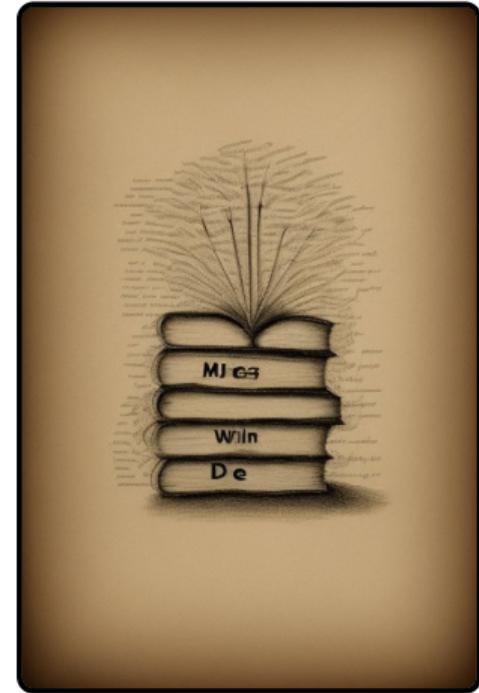
Department of Computer Science

“

*Someone who gets into other people's computer systems without permission in order to find out information or to do something illegal*

”

from the Cambridge Dictionary



# Terminology | What makes an attacker?



UNIVERSITY  
*of York*

Department of Computer Science

Intent

9

A person or group deliberately taking actions to gain information or access outside of the scope of their permissions

8

A person that will maliciously attempt to attack a system by breaching or manipulating security systems. An attacker is usually out for either personal or economic gain.

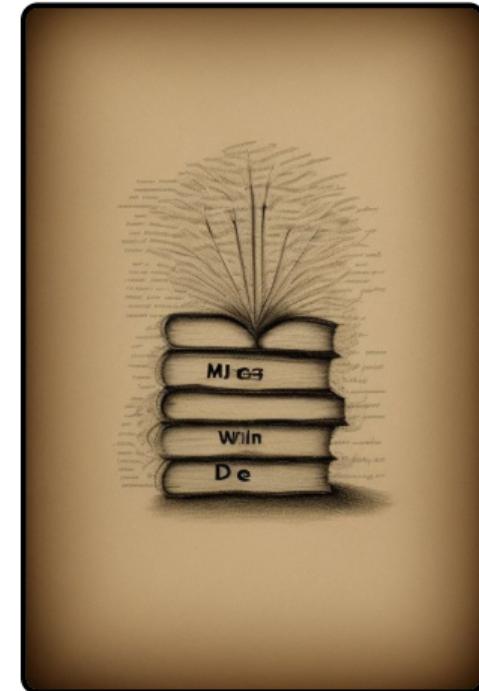
6

Attacker is a person, group, or automated system—that attempts to compromise the security of a system, network, or application by exploiting vulnerabilities to gain unauthorized access, steal data, di

5

A person with intent to harm/damage a system

6



# Terminology | What makes an attacker?



UNIVERSITY  
of York

Department of Computer Science

a hacker  
with bad  
intentions

5

Will there be a small  
break since it's a 2  
hour session?

5

Someone who acts upon the  
vulnerabilities discovered in a system

3

A damaging result eg. a system  
going down, or information privacy  
being compromised

3

Any benefit or personal grudge. a  
person / group etc, or sometime  
just for fun

3

Fun!

3

Gaining access to a system  
with malicious intent

3

Can you close the  
door please, it's cold

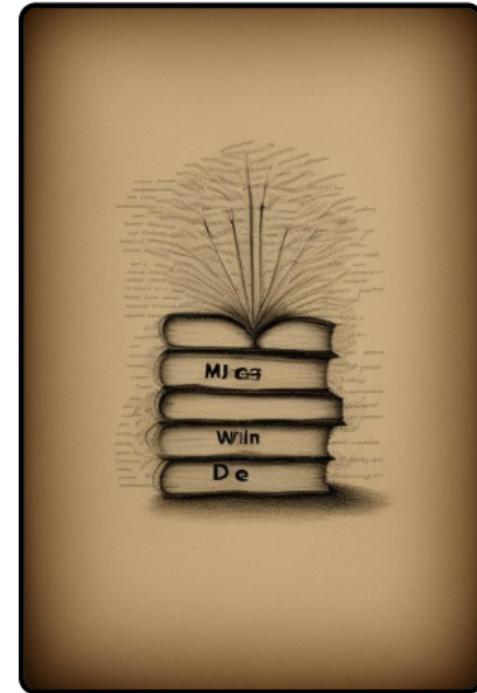
2

Revenge

2

getting a bit  
silly with it

2



# Terminology | What makes an attacker?

C ¾



Department of Computer Science

A person/system that attempts to bypass a system's security

2

An adversary with the goal of compromising a system through any means

2

Someone who poses a threat to a network or system, where some valuable resource could be taken advantage of

2

bad  
Money  
Wealth

2

An attacker is made by the amalgamation of bad life experience + good tech skills

2



Salvos bouncers

2

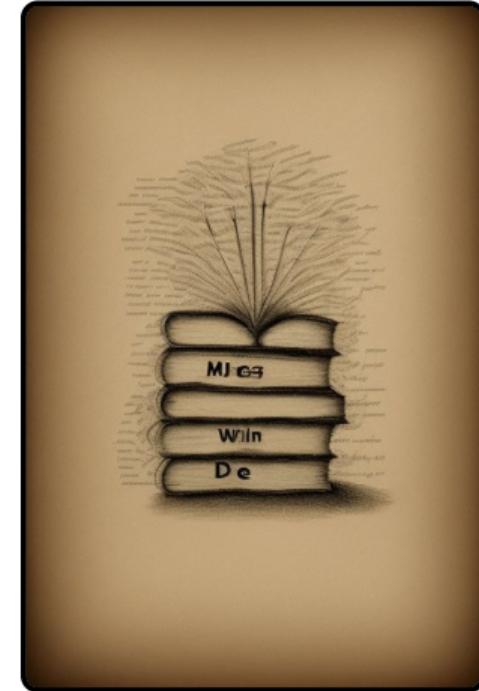


Permission

1

Curiosity

1



# Terminology | What makes an attacker?

O 4%



Department of Computer Science

Knowledge to able to pursue attacks with intent of causing harm.

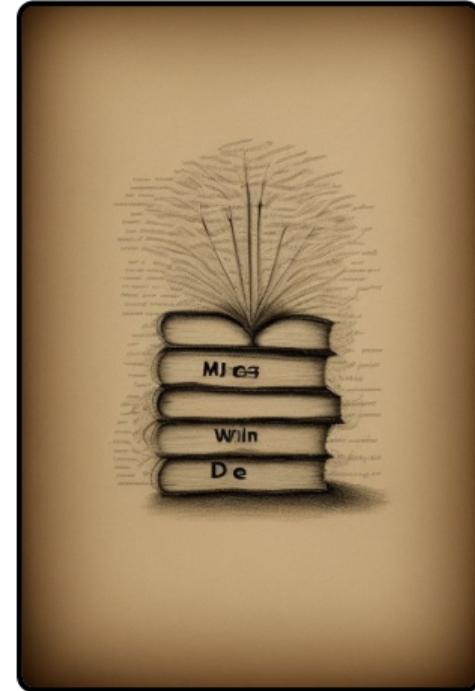
1

Someone who initiates combat

black hat vs grey hat vs white hat

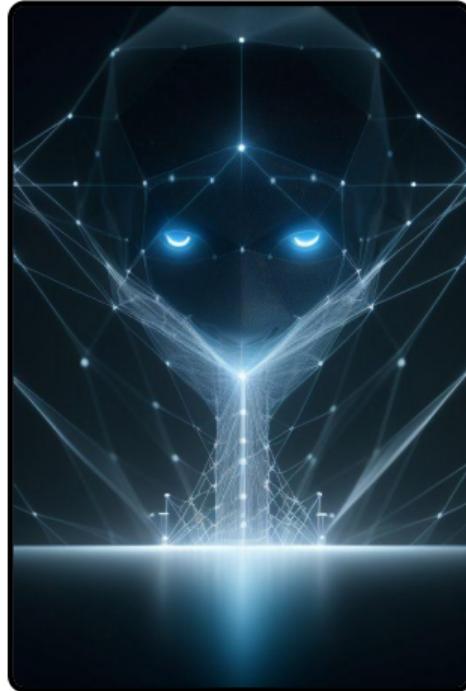
I like to use these four elements to wrap up what *makes an adversary*.

“ Motivation, vulnerability, target, and lack of consent ”



# Definition of Ethical Hacking

- ▶ **Penetration testing** consists of security testing mimicking real-world attacks to identify vulnerabilities in applications, systems, or networks
- ▶ Can pentesting be malicious hacking?
- ▶ We use **Ethical hacking** as an umbrella term that goes beyond pentesting and includes vulnerability assessments, security audits, and other proactive measures to ensure the overall security of an organization



# Importance of Ethical Hacking

## Impact and severity of vulnerabilities...

...can be highlighted by strategically deploying a **honey pot**.

- ▶ An ethical hacker could track malicious activities but also reveal vulnerabilities of poorly configured systems.

- 
- ▶ Protecting systems and networks
  - ▶ Identifying vulnerabilities proactively



# Key Concepts in Cyber Security: Pillars

Certain ethical hacking occurrences may not result in harm, or at least not in the traditional sense. However, these instances are frequently categorized within the conventional computer security threat model.

- ▶ Confidentiality, Integrity, Availability (CIA)
- ▶ Principle of Least Privilege
- ▶ Defense-in-Depth

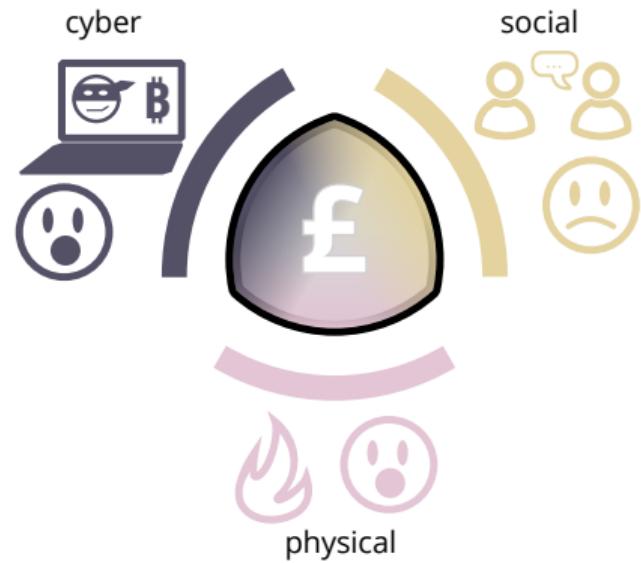


# Key Concepts in Cyber Security: Impact

**social** Erode public trust, disrupt lives, and manipulate societal narratives through disinformation campaigns and identity theft.

**cyber/digital** Compromise sensitive data, disrupt essential services, and create vulnerabilities in interconnected systems, enabling further attacks.

**physical** Real-world consequences: power grid failures, industrial sabotage, life-threatening incidents in healthcare and transportation.



# Attack Surface, Vectors and Vulnerabilities

Attack surface increases with devices and connections

Attack vectors are techniques that can be used  
social engineering, ransomware, ...

Vulnerabilities and poor configurations can be exploited



- 
- ▶ **MITRE ATT&CK® – Adversarial Tactics, Techniques, and Common Knowledge**  
classification of real-world cyberattacks and intrusions
  - ▶ **CVE – Common Vulnerabilities and Exposures)**  
collection of publicly known software vulnerabilities

# Hacker Categories

|   | Motivation   | Legality   |
|---|--|--|
| <br><b>black hat</b> | Malicious, seeking personal gain or harm                       | Generally illegal and unethical  |
| <br><b>gray hat</b>  | Falls between black and white hat, often with ambiguous ethics | May involve technically illegal activities with uncertain ethical boundaries |
| <br><b>white hat</b> | Ethical hackers strengthening cybersecurity defenses           | Legal and performed with explicit permission                                 |



# Questions & Answers



<https://padlet.com/robertometere/ehac>

## Further reading...



Alana Maurushat. *Ethical Hacking* – University of Ottawa Press (2019)

# Concepts from Crypto and Networks

---

EHAC | Lecture 2

Dr. Roberto **Metere**

✉ roberto.metere@york.ac.uk

# Objectives

At the end of this lecture you will know about...

- ▶ what elements of Networks are most relevant to our module
- ▶ what elements of Cryptography are most relevant to our module

# Communication process



## Basic elements

- ▶ Sender
- ▶ Receiver
- ▶ Message
- ▶ Channel
- ▶ Code
- ▶ Context

Both Cryptography and Networks relate to Communications

# Elements of Cryptography

# Encoding and decoding



UNIVERSITY  
of York

Department of Computer Science

## Encoding

What is it? Converting data from one form to another.

What for? For transmission, storage, or *encryption*

ASCII Character encoding

'A' = 65 = 1000001<sub>2</sub>

base64 Binary-to-text encoding **pwd** field of zoom links, e.g.,  
S0h3aXhxY2hkVHU1MnVwRHAYNU1Pdz09

ROT13 alphabet letters are shifted by 13 positions



# Encoding and decoding | Let's practice

0 2/2



Department of Computer Science

Let's encode and decode a string with base64

```
$ echo "" | base64
```

```
$ echo "" | base64
```

```
$ echo "" | base64 -d
```

```
$ echo "" | base64 -d
```

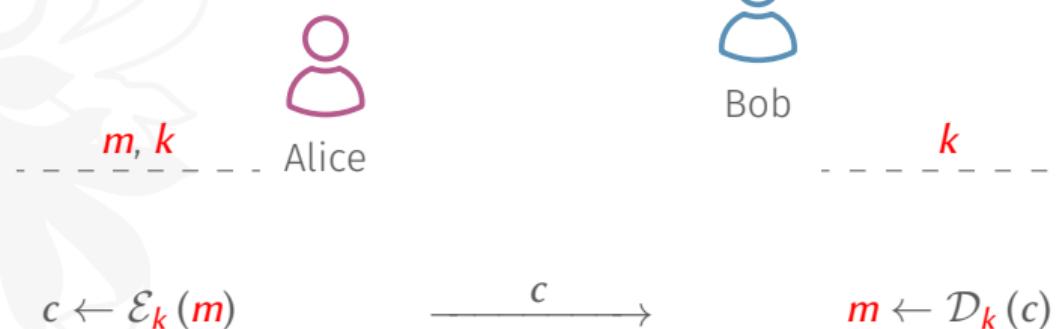


# Symmetric or Private-key encryption



UNIVERSITY  
*of York*

Department of Computer Science



A **private-key encryption scheme** is a triplet of PPT algorithms  $(\mathcal{K}, \mathcal{E}, \mathcal{D})$

- ▶ key generator  $\mathcal{K} : \mathbb{N} \rightarrow K$
- ▶ encryption  $\mathcal{E} : K \rightarrow \{0, 1\}^* \rightarrow \{0, 1\}^*$  deterministic or probabilistic algorithm
- ▶ decryption  $\mathcal{D} : K \rightarrow \{0, 1\}^* \rightarrow \{0, 1\}^*$  deterministic algorithm

# Symmetric encryption | Let's practice

O ½



Department of Computer Science

## Let's encrypt and decrypt a text file with AES-256-CBC

```
$ echo "" > my-ll-secret.txt; cat my-ll-secret.txt  
$ openssl enc -aes-256-cbc -in my-ll-secret.txt -out my-ll-secret.enc  
enter AES-256-CBC encryption password:  
$ cat my-ll-secret.enc | base64  
$ openssl enc -d -aes-256-cbc -in my-ll-secret.enc -out my-ll-secret.dec  
enter AES-256-CBC decryption password:  
$ diff my-ll-secret.txt my-ll-secret.dec
```

# Asymmetric or Public-key encryption



UNIVERSITY  
of York

Department of Computer Science



$$c \xleftarrow{\$} \mathcal{E}_{k_p}(m) \xrightarrow{c} m \xleftarrow{\$} \mathcal{D}_{k_s}(c)$$

A **public-key encryption scheme** is a triplet of PPT algorithms  $(\mathcal{K}, \mathcal{E}, \mathcal{D})$

- ▶ key generator  $\mathcal{K} : \mathbb{N} \rightarrow K_p \times K_s$   
often, a public key  $k_p \in K_p$  is a function of the private key  $k_s \in K_s$ ,  $k_p = f(k_s)$ .
- ▶ encryption  $\mathcal{E} : K_p \rightarrow \{0, 1\}^* \rightarrow \{0, 1\}^*$  probabilistic algorithm
- ▶ decryption  $\mathcal{D} : K_s \rightarrow \{0, 1\}^* \rightarrow \{0, 1\}^*$  deterministic algorithm

# Asymmetric encryption | Let's practice

0 2/3



Department of Computer Science

Let's first generate a keypair with the RSA algorithm

```
$ openssl genrsa -out key.private 512; cat key.private  
$ openssl rsa -in key.private -out key.public -outform PEM -pubout; cat key.public
```

# Asymmetric encryption | Let's practice

O 3/3



## Let's encrypt and decrypt a text file with our freshly generated RSA keys

```
$ openssl pkeyutl -encrypt -inkey key.public -pubin -in my-ll-secret.txt -out my-ll-secret.enc  
$ cat my-ll-secret.enc | base64  
$ openssl pkeyutl -decrypt -inkey key.private -in my-ll-secret.enc -out my-ll-secret.dec  
$ diff my-ll-secret.txt my-ll-secret.dec
```

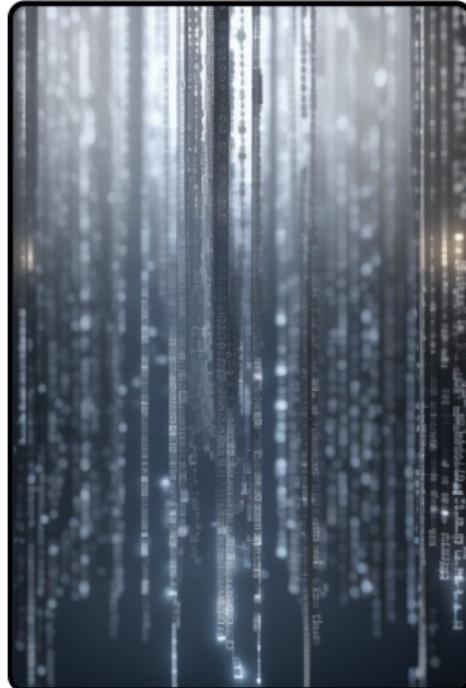
- ▶ The file size cannot exceed a bunch of bytes

# Hybrid encryption

## How it works

- ▶  Public-key encryption can provide a secret channel where **private keys are shared between hosts**
- ▶  Private-key encryption is then used for **intensive communications**

- ▶  might be Diffie-Hellman
- ▶  might be AES
- ▶ An everyday example is the SSL/TLS protocol



# Hash functions

## # Hash functions

What is it? Map input data into a fixed-size bitstring, called a **hash** or a **digest**

What for? Password storage, data integrity, digital signatures

MD5 historically used but weak

SHA-1 deprecated for critical applications

SHA-256 widely used for secure applications



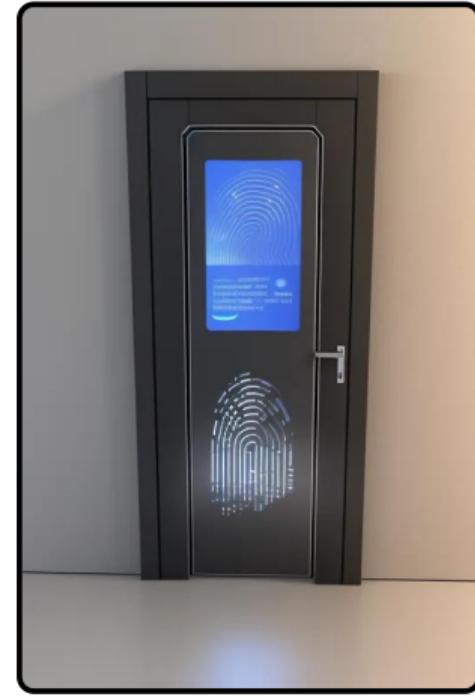
# Hash functions | Let's practice



Department of Computer Science

## Let's calculate some hashes

```
$ echo "" | md5sum  
$ printf "" | md5sum  
$ echo "" | sha1sum  
$ echo "" | sha256sum
```



# Hash functions | MD5 is weak

0 3/3



Department of Computer Science

How weak is MD5? (clash: 253dd04e87492e4fc3471de5e776bc3d)



[https://vle.york.ac.uk/bbcswebdav/xid-25529834\\_2](https://vle.york.ac.uk/bbcswebdav/xid-25529834_2)



[https://vle.york.ac.uk/bbcswebdav/xid-25529833\\_2](https://vle.york.ac.uk/bbcswebdav/xid-25529833_2)



# Diffie-Hellman Man-In-The-Middle Attack



UNIVERSITY  
of York

Department of Computer Science



$$x \xleftarrow{\$} E$$
$$X \leftarrow g^x$$

$$\xrightarrow{X}$$

$$\tilde{x} \xleftarrow{\$} E$$

$$\tilde{X} \leftarrow g^{\tilde{x}}$$

$$\tilde{y} \xleftarrow{\$} E$$

$$\tilde{Y} \leftarrow g^{\tilde{y}}$$

$$\xrightarrow{\tilde{X}}$$
$$\xleftarrow{Y}$$

$$y \xleftarrow{\$} E$$

$$Y \leftarrow g^y$$

$$\tilde{Y}^x$$

$$k_A = \tilde{Y}^x$$
$$k_A = \tilde{X}^{\tilde{y}}, k_B = \tilde{Y}^{\tilde{x}}$$

$$k_B = \tilde{X}^y$$

where  $E = \{2, \dots, p - 1\}$  is the sampling set for secret exponents.

# Diffie-Hellman Man-In-The-Middle Attack

O ½



Department of Computer Science

Alice |  $k_A, m$

$$c \leftarrow \mathcal{E}_{k_A}(m)$$

$$\xrightarrow{c}$$

$\mathcal{A}$  |  $k_A, k_B$

$$m = \mathcal{D}_{k_A}(c)$$

$$c' \leftarrow \mathcal{E}_{k_B}(m)$$

$$\xrightarrow{c'}$$

Bob |  $k_B$

$$m = \mathcal{D}_{k_B}(c')$$

| <b>Weakness in DH</b>  | <b>Why it's vulnerable</b> | <b>Fix/Protection</b>      |
|------------------------|----------------------------|----------------------------|
| No authentication      | Anyone can impersonate     | Digital signatures         |
| Public values in clear | Easily replaced by MITM    | Use certificates (PKI)     |
| No integrity checks    | Cannot detect tampering    | Authenticated key exchange |

# Elements of Networks

# Communication elements



Sender Users' mobile phone

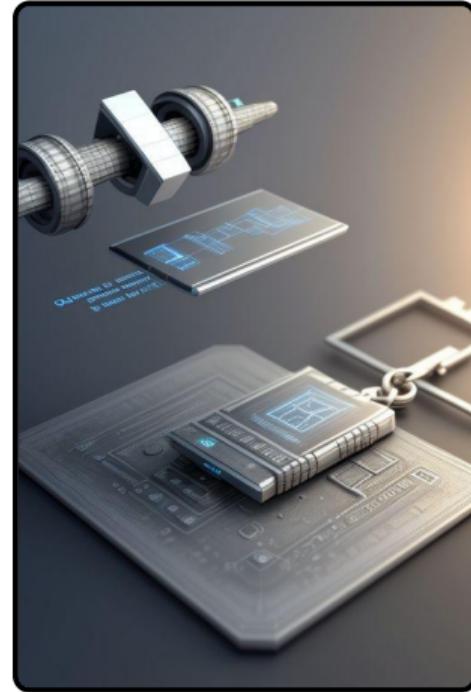
Receiver Vehicle on-board computer

Message Charge battery

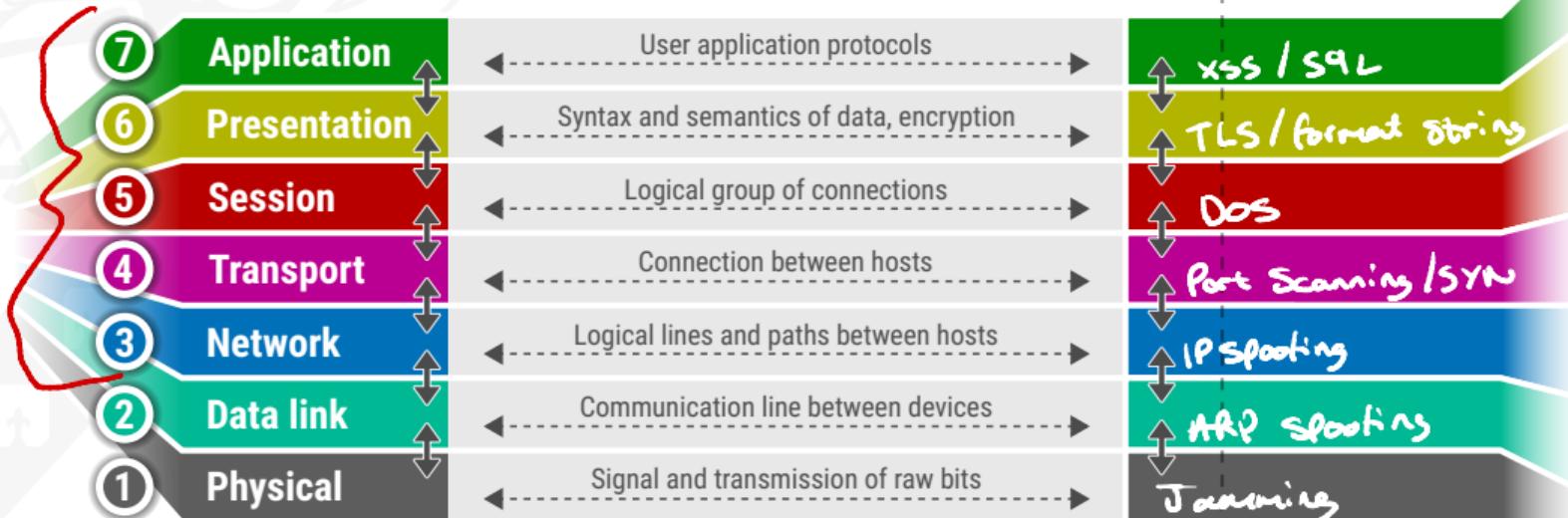
Channel Bluetooth

Code Predefined commands

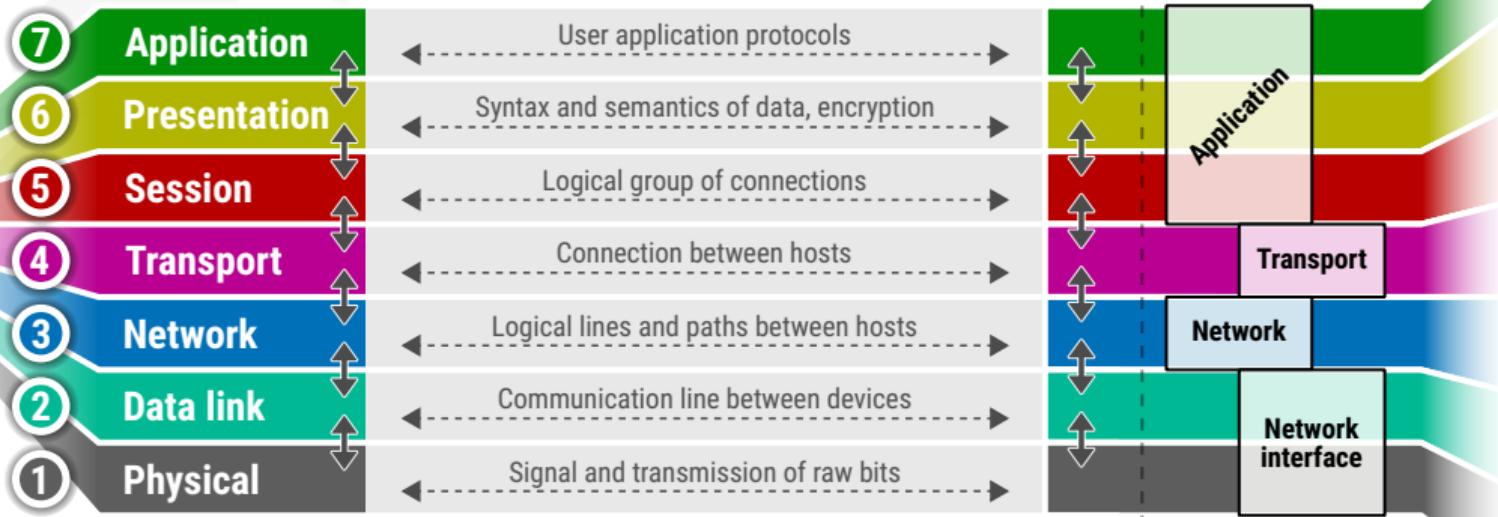
Context User is authenticated



# ISO-OSI model and TCP/IP



# ISO-OSI model and TCP/IP



## Other Networking Concepts



UNIVERSITY  
*of York*

Department of Computer Science

- ▶ VLAN (virtual local area network) segmentation
  - ▶ Firewalls
  - ▶ Intrusion detection/prevention systems
  - ▶ Cyber Intelligence

# Network Security that use Cryptography

- ▶ VPN (virtual private network)
  - ▶ DMZ (demilitarized zone) not inherently crypto
  - ▶ Security protocols HTTPS, SSH, TLS, DNSSEC, ARP, WiFi, ...



# Other Networking Concepts | Let's practice

0 ½



## Some network tools...

```
$ arp -en  
$ resolveip ssh.york.ac.uk  
$ ssh user@ssh.york.ac.uk  
$ curl https://www.york.ac.uk/ 2>/dev/null | head -1
```

Fetch Data



# Hypertext Transfer Protocol (HTTP)

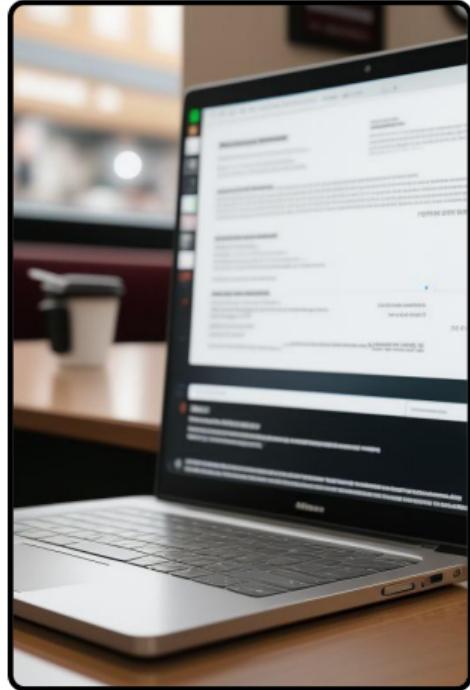
HTTP is **stateless** (each request is independent and unrelated to previous requests) and **connectionless** (a new connection is established for each request and closed after the response).

## Sessions

- ▶ Cookies
- ▶ URL rewriting
- ▶ Hidden form fields
- ▶ Header session tokens

## Inputs

- ▶ Buffer overflow      input is too long
- ▶ Sanitisation      of input, input type
- ▶ Injections      server-side code, SQL, ...
- ▶ Cross-Site Scripting (XSS)      injected malicious client-side scripts
- ▶ Cross-Site Request Forgery (CSRF)      cross-domain unauthorized requests



# Hypertext Transfer Protocol (HTTP)

## Methods and status codes

**GET** Retrieve data from the server.

**POST** Submit data related to a specified resource.

**PUT** Update a resource on the server.

**DELETE** Request the removal of a resource.

... ...

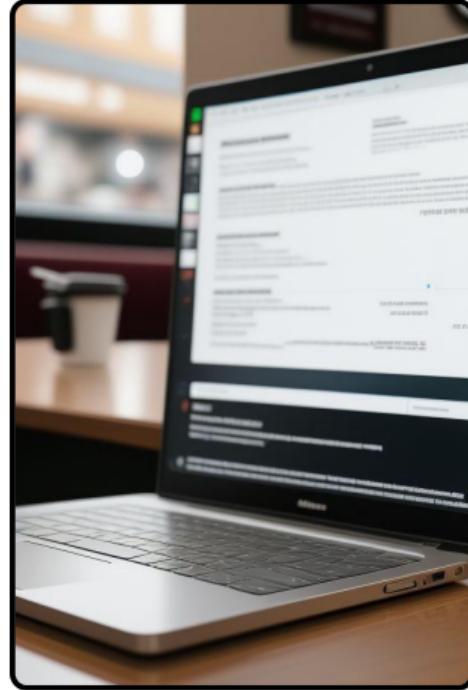
**1XX** Informational

**4XX** Client errors

**2XX** Success

**5XX** Server errors

**3XX** Redirection



# Recap



<https://www.menti.com/al8d7fjjsauq>

# Beyond EHAC

O ½



Department of Computer Science

## Online Courses and Specialisations

- ▶ Cybrary
- ▶ Udemy
- ▶ Coursera
- ▶ Various certifications...

## Practical Labs and Platforms

- ▶ Hack The Box (HTB)
- ▶ TryHackMe
- ▶ OverTheWire



# Beyond EHAC



UNIVERSITY  
*of York*

Department of Computer Science

|                            | TryHackMe                                  | OverTheWire                                 | HackTheBox                                 |
|----------------------------|--|---|--|
| <b>Level of Difficulty</b> | Beginner to Intermediate                   | Beginner to Advanced                        | Intermediate to Advanced                   |
| <b>Content Variety</b>     | Rooms, tutorials, and virtual environments | Wargames, CTFs, and challenges              | Machines, challenges, and labs             |
| <b>Labs &amp; Machines</b> | Emphasis on guided labs and scenarios      | Primarily focused on wargames and exercises | Virtual machines with real-world scenarios |
| <b>Collaboration</b>       | Supports collaboration on challenges       | Individual or team-based challenges         | Individual or team-based challenges        |
| <b>Realism</b>             | Mix of real-world scenarios and CTFs       | Realistic scenarios and hacking challenges  | Simulated real-world environments          |
| <b>Pricing</b>             | Freemium model with free and paid plans    | Free to use                                 | Free and VIP plans                         |

# Questions & Answers



<https://padlet.com/robertometere/ehac>

## Further reading...

-  Michael Sikorski and Andrew Honig – no starch press (2012)  
*Practical Malware Analysis: A Hands-On Guide to Dissecting Malicious Software.*

# Blue Team, Red Team and Penetration Testing

---

EHAC | Lecture 3

Dr. Roberto **Metere**

✉ roberto.metere@york.ac.uk

# Objectives

At the end of this lecture you will know about...

---

- ▶ what are the main responsibilities of the Blue Team
- ▶ what are the main responsibilities of the Red Team
- ▶ what are the phases and objectives of penetration testing

# The Blue Team and the Red Team

# The Blue Team

## Main responsibilities

- ▶ **Identify vulnerabilities** within an organization's systems and networks.
- ▶ **Implement security measures** to protect assets from unauthorized access.
- ▶ **Monitor** network for potential threats or attacks.
- ▶ Ensure that all systems **remain secure**.



# The Red Team

## Main responsibilities

- ▶ **Simulate real-world cyberattacks** on the target system or network.
- ▶ **Identify vulnerabilities** within an organization's security measures and provide **recommendations**.
- ▶ **Exploit** vulnerabilities to gain access to the target system or network.
- ▶ Test the effectiveness of an organization's security measures by attempting to **breach their defenses**.
- ▶ **Social engineering** testing and educate employees



# Blue and Red Teams | Overlaps (our answers)

0 ¼



Department of Computer Science

Identifying  
vulnerabilities

18

May use similar tools- to find  
weaknesses to fix and weaknesses to  
exploit

8

Both are aware of  
common exploits.

6

both protecting the  
network

13

Work to improve / maintain  
security measures

9

Red team communicates their findings  
to the blue team to improve defences

7

they aim to improve security, albeit  
through different approaches; red team  
by attacking, and the blue by defending

6

Similar qual-  
ifications

5



# Blue and Red Teams | Overlaps (our answers)

0 2/4



Department of Computer Science

Purple team when working together

5

"Testing" security measures in place

4

both need to research new exploits and attack methods

4

Analysing previous network and systems attacks. Blue team from a purely analytical pov, red team in an attempt to recreate and simulate such attacks.

4

passive vs active

3

Both being paid (hopefully)

3

Intent

3

Red teamers often know some Blue teaming tactics to increase the effectiveness of penetration tests

3



# Blue and Red Teams | Overlaps (our answers)

O 8/4



Department of Computer Science

testing effectiveness

2

Both working for an organisation

2

often work together in cybersecurity to improve an organization's defense strategies

2

Avoiding Zero-day exploit

2

Network vulnerability scanning

1

knowledge of inner working of the system.

1

Work for the same customer (who also own the system)

1

built a better Internet environment together

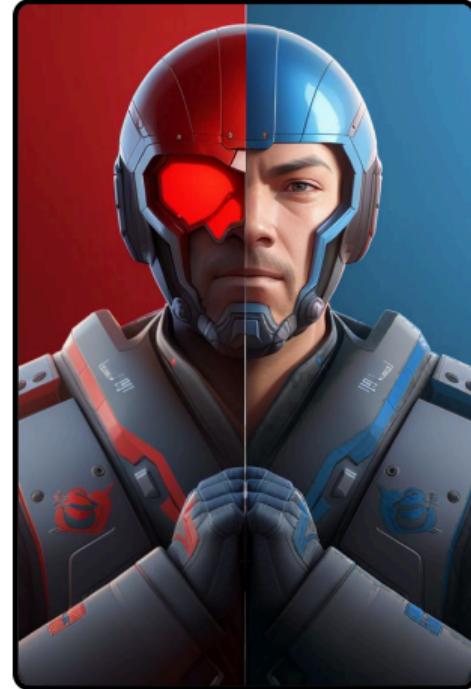
1

Work together on building a full-proof barrier to protect the System.

1

Offensive vs defensive

1



# Blue and Red Teams | Let's practice

O 4%



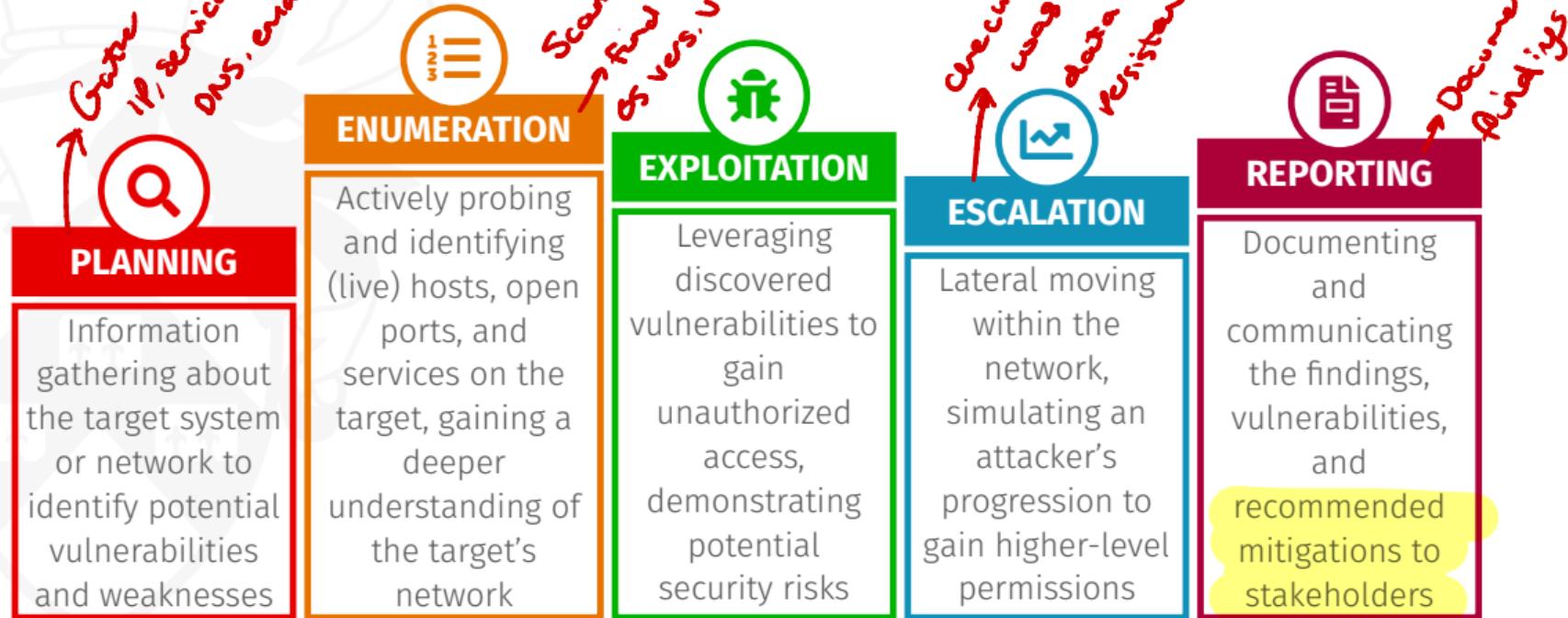
Department of Computer Science

```
$ nmap -Pn -p 80,443,22 google.com
...
PORT      STATE     SERVICE
22/tcp    filtered ssh
80/tcp    open      http
443/tcp   open      https
...
$ cat /etc/ssh/sshd*g | grep -v '^#' | grep -iE "(port|pass|root)"
PermitEmptyPasswords yes
$ printf "GET / HTTP/1.1\r\nHost: york.ac.uk\r\n\r\n" | nc
york.ac.uk 80
HTTP/1.1 302 Moved Temporarily
Location: https://www.york.ac.uk/
Strict-Transport-Security: max-age=31536000
Server: BigIP
Content-Length: 0
Connection: keep-alive
```



# Penetration testing

# Stages of penetration testing



# Planning stage

- 
- ▶ Target some data, server, employees, ...
  - ▶ Legal boundaries consequences of **real** attacks
  - ▶ Notify people involved social engineering
  - ▶ Gather information about the target(s)



## PLANNING

Information gathering about the target system or network to identify potential vulnerabilities and weaknesses

# Planning stage | Let's practice



Perform a SYN scan (**-sS**), OS and service version detection  
**(-A)** on all ports of the target IP address

```
$ nmap -sS -p 22,21,80 -A york.ac.uk
...
21/tcp filtered ftp
22/tcp filtered ssh
80/tcp open     http-proxy F5 BIG-IP load balancer http
               proxy
...
OS fingerprint not ideal because: Missing a closed TCP
port so results incomplete
No OS matches for host
...
```



# Planning stage | Let's practice

0 3/3



Department of Computer Science

```
$ ping -c1 <address>
PING nasa.gov (192.0.66.108) 56(84) bytes of data.
64 bytes from 192.0.66.108: icmp_seq=1 ttl=56 time=34.5 ms
```

...

```
$ whois google.com
```

...

```
Updated Date: 2025-01-29T10:41:03Z
Creation Date: 2000-03-02T14:40:13Z
Registry Expiry Date: 2026-03-02T14:40:13Z
```

...



## PLANNING

Information gathering about the target system or network to identify potential vulnerabilities and weaknesses

PEDEEG2

|   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|
| P | E | D | E | E | G | 2 |
| L | n | o | x | s | c |   |
| R | u | c | p | c | p |   |
| B | u | v | l | a | o |   |
| E | m | o | l | r |   |   |
| T | e | i | a | b |   |   |
| A | n | t | t |   |   |   |
| C | t |   |   |   |   |   |
| C |   |   |   | o |   |   |

- ▶ Vulnerability scanning OS, versions, configurations, ...
- ▶ Active port scans, replay, guess passwords, phishing, spyware, ...
- ▶ Passive monitoring persons, monitoring network traffic...
- ▶ Human dumpster diving, social engineering, shoulder surfing



## ENUMERATION

Actively probing and identifying (live) hosts, open ports, and services on the target, gaining a deeper understanding of the target's network

# Enumeration stage | Let's practice

0 2/2

Gather information from a Windows system: such as shares, users, and groups.

```
$ enum4linux-ng -As
```

```
...  
[*] Checking LDAP  
[-] Could not connect to LDAP on 389/tcp: timed out  
[*] Checking LDAPS  
[-] Could not connect to LDAPS on 636/tcp: timed out  
[*] Checking SMB  
[-] Could not connect to SMB on 445/tcp: timed out  
[*] Checking SMB over NetBIOS  
[-] Could not connect to SMB over NetBIOS on 139/tcp: timed out  
[!] Aborting remainder of tests since neither SMB nor LDAP are  
accessible
```

Completed after 20.10 seconds



## ENUMERATION

Actively probing and identifying (live) hosts, open ports, and services on the target, gaining a deeper understanding of the target's network

# Exploitation stage

- 
- ▶ Buffer overflow
  - ▶ Cross-Site Scripting (XSS)
  - ▶ Code/SQL Injections
  - ▶ Sniffing traffic and poisoning
  - ▶ Misconfiguration
  - ▶ Kernel flaws
  - ▶ Race conditions
  - ▶ Bad permissions



# Exploitation stage | Let's practice

0 2/2



## SQL injection

pong Can you play pong? <http://tiny.cc/dpouwz>

A solution is to append the following to the address bar

?email=a' OR '1' LIKE '1

Thanks to all students who helped me overcome my bad SQL injection skills with this working suggestion!



The graphic features a green circular icon with a white figure of a person carrying a shield. Below it is a green rectangular box with the word "EXPLOITATION" in white. To the right of the box is a white rectangular area with a thin green border containing the following text:

Leveraging discovered vulnerabilities to gain unauthorized access, demonstrating potential security risks

## Escalation stage

- ▶ **Consolidate access** to systems by exploiting vulnerabilities and gaining control over multiple entry points
- ▶ Achieve **privilege escalation** by leveraging weaknesses in user permissions or system configurations to gain higher-level access
- ▶ **Extract sensitive data** (exfiltration), such as credentials, confidential information, or intellectual property, from the targeted systems.

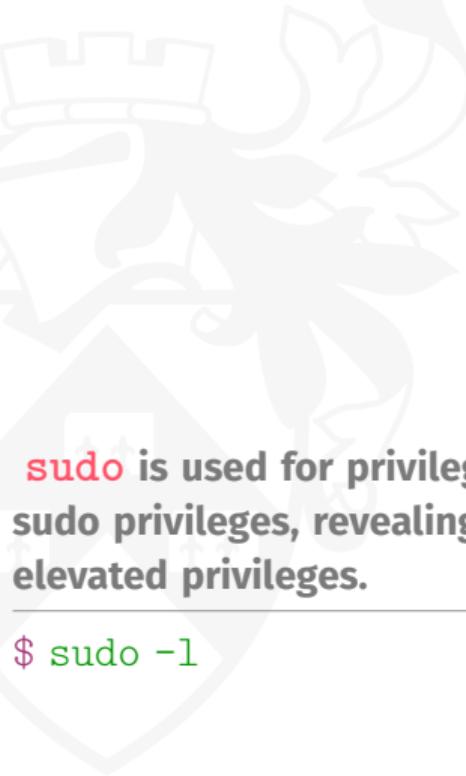


### ESCALATION

Lateral moving within the network, simulating an attacker's progression to gain higher-level permissions

# Escalation stage | Let's practice

0 ½



**sudo** is used for privilege escalation. Let's check current user's sudo privileges, revealing what commands can be executed with elevated privileges.

```
$ sudo -l
```



**ESCALATION**

Lateral moving within the network, simulating an attacker's progression to gain higher-level permissions

# Reporting stage

0 ½



Department of Computer Science

- ▶ **Organize findings** in perceivable format – tables, charts, graphs
- ▶ Detailed **explanations of each vulnerability**  
discovery, severity, impact, and remedies
- ▶ Recommendations for **improving security** measures and  
**preventing attacks** in the present and the future
- ▶ **Maintain confidentiality** by only sharing the report with  
authorized personnel within the organization



## REPORTING

Documenting and  
communicating  
the findings,  
vulnerabilities,  
and  
recommended  
mitigations to  
stakeholders

## Reporting stage | Let's practice

0 2/2

Big / common answer can be sanitisation



Department of Computer Science



### REPORTING

Documenting and communicating the findings, vulnerabilities, and recommended mitigations to stakeholders

Generate a report of potential vulnerabilities and misconfiguration by performing comprehensive tests against a web server

```
$ nikto -h <target>
```

# Types of Penetration Tests

|                                   | Black box testing     | Gray box testing           | White box testing       |
|-----------------------------------|-----------------------|----------------------------|-------------------------|
| <b>Knowledge of System</b>        | limited/none          | partial                    | comprehensive           |
| <b>Information Access</b>         | external perspective  | partial internal           | full internal           |
| <b>Scope of Knowledge Sharing</b> | no details shared     | limited details shared     | full details shared     |
| <b>Simulates</b>                  | external attackers    | insider or partial insider | internal team or admins |
| <b>Testing Time</b>               | longer, comprehensive | moderate                   | shorter, focused        |

# Recap



<https://www.menti.com/alwzfbcgatpd>

# Questions & Answers



<https://padlet.com/robertometere/ehac>

## Further reading...

- 
-  The Penetration Testing Execution Standard | <http://www.pentest-standard.org>
  -  Michael Sikorski and Andrew Honig – no starch press (2012)  
*Practical Malware Analysis: A Hands-On Guide to Dissecting Malicious Software.*

# Introduction to Reverse Engineering

---

EHAC | Lecture 4

Dr. Roberto **Metere**

✉ roberto.metere@york.ac.uk

# Objectives

At the end of this lecture you will know about...

---

- ▶ main concepts of computer architectures
- ▶ what can produce a buffer overflow
- ▶ what are the main obfuscation and anti-debugging techniques

# Definitions | Reverse Engineering



UNIVERSITY  
*of York*

Department of Computer Science

gnireenigne

11

Please can I have  
the checkin code  
again Roberto :(

6

Tracing back the path  
of how it was  
originally build to find  
its characteristics and  
behaviors

7

Taking something  
apart to figure  
out how it works

6

Reverse engineering is the process of  
taking apart a fully functioning product  
to find out what components work to  
achieve the final output of the product.

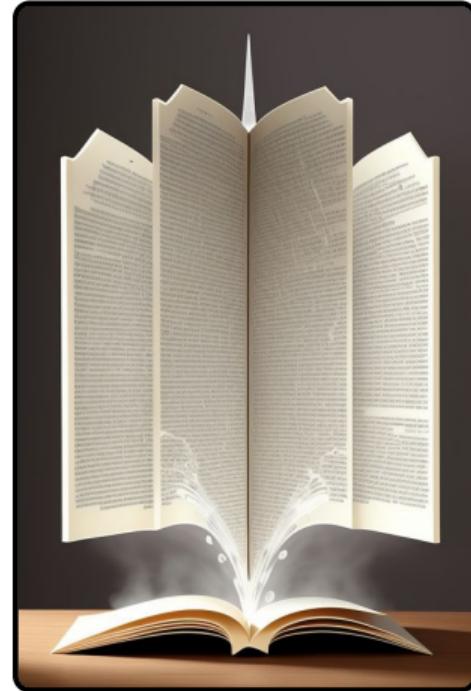
7

Decinstructing  
software to see  
how it's designed  
and how it  
functions

6

Figuring out the  
inner workings of  
compiled  
software

6



# Definitions | Reverse Engineering



UNIVERSITY  
*of York*

Department of Computer Science

Examining a completed piece of hardware or software systematically, to try and figure out how it works

4

Reverse engineering is the analytical deconstruction of systemic methodologies to extrapolate foundational paradigms and reconstruct intrinsic functionalities in a non-linear iterative process.

4

deconstruction to see how something was originally built

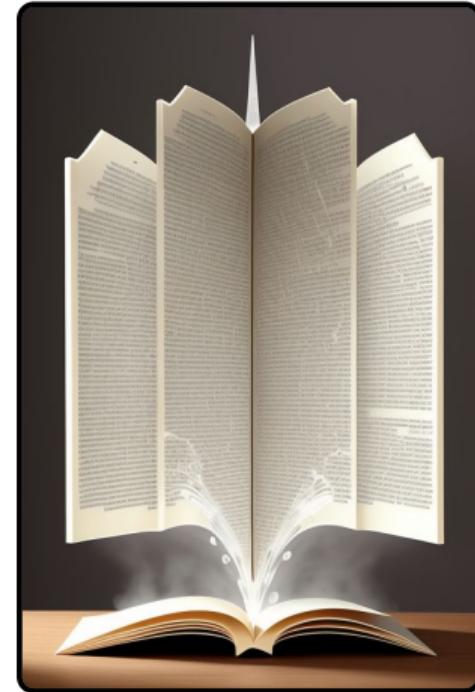
4

Uncompiling

4

Able to understand a stolen technology and able to create something without any prior knowledge

3



# Definitions | Reverse Engineering

O ¾



Department of Computer Science

Working backwards to achieve a certain goal such as re-building something

3

figure out what something does

2

Engineering in reverse

2

Deconstructing to figure out how to reconstruct and replicate

3

Turning an executable back into higher-level (human readable) code

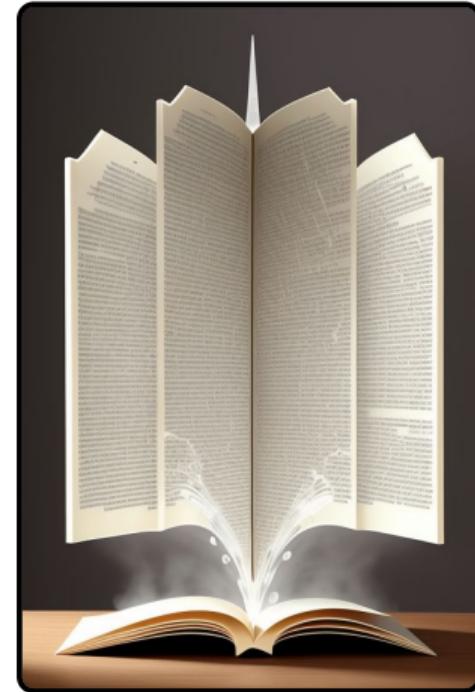
2

looking at the parts of a finished system (e.g. a program) to figure out how it works.

3

Breaking something down to learn from it

2



# Definitions | Reverse Engineering

O ¼



Department of Computer Science

break it and see how it works from the inside

1

Taking something apart to look at what makes it tick

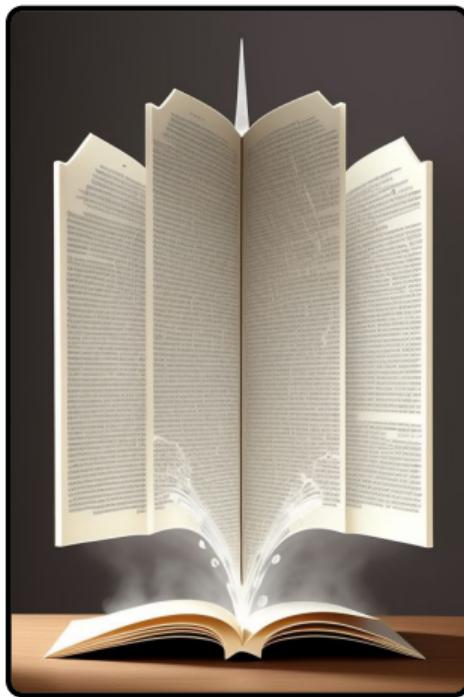
1

Analysing the biology of a system

Reverse the code back to hack the target

“ the act of copying the product of another company by looking carefully at how it is made ”  
from The Cambridge Dictionary

Apart from the “gnireenignE” fun answer, I like more the definitions we gave in class.



# Definitions | Binary file

O ¼



Department of Computer Science

A binary file is a type of file that contains data in a format that is not meant to be read as text. Unlike text files, which are composed of human-readable characters (like letters, numbers, and punctuation), binary files are encoded using binary code (a sequence of 0s and 1s).

8

file not encoded with a type of human-readable encoding

5

file with 0s and 1s

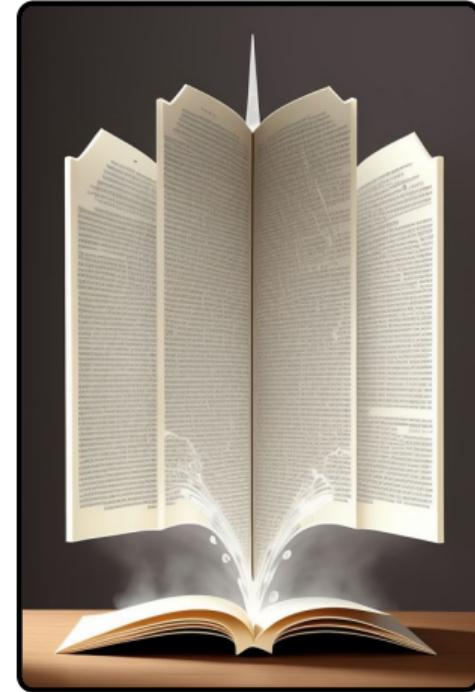
5

A low-level file containing the compiled source code of a program.

8

Every file if you look hard enough

5



# Definitions | Binary file



UNIVERSITY  
of York

Department of Computer Science

A file that's in 1 and 0 and can store data like application, images and other stuff

4

A file that stores data in a binary format (0's and 1's).

4

all of them?

3

01100110  
01101001  
01101100  
01100101

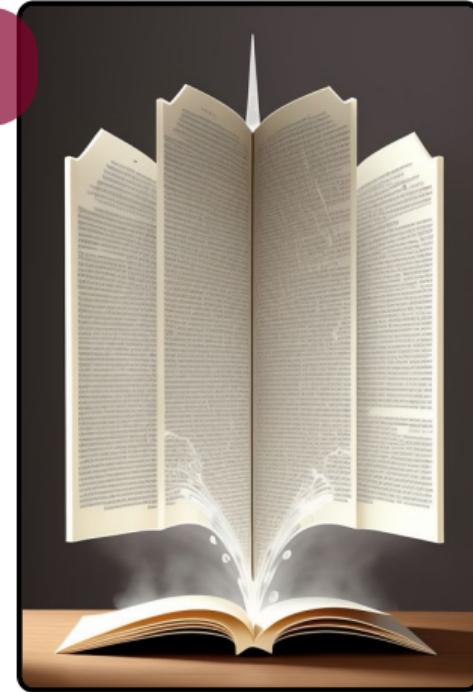
4

Not human-readable file

4

Can be used to allow someone to use your code but in a format such that they cannot understand and alter it in a way that leaves it functional afterwards

3



# Definitions | Binary file



UNIVERSITY  
*of York*

Department of Computer Science

A file containing code contains functions translated very low-level non-readable code. Needs to be turned into machine code to be used by a specific architecture

3

10011010 01010100 11100011 00010011 00011100 10100110  
01110000 11011010 01010100 01100100 1 (compiled program)

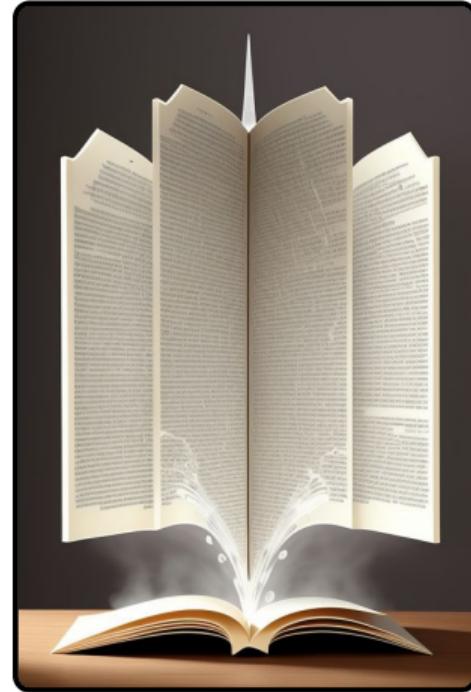
2

Requires a specific software to interpret and use its content

2

a compiled program containing only opcodes (in this context, It Depends:tm:)

1



# Definitions | Binary file

O ¼



Department of Computer Science

binary file

1

a file with the  
extension .bin

1

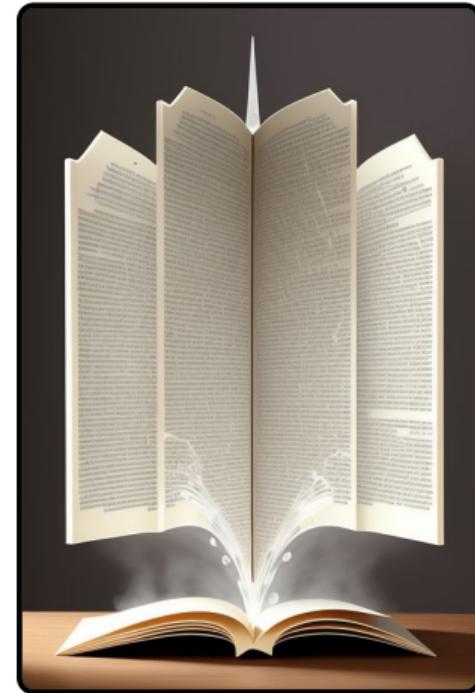
Low level code that the  
machine can understand  
better

1

A file that uses binary code  
formatting as opposed to ascii  
formatting or hex or something?

1

A binary file something that is low level  
but also can be humanly analyzed using  
tools



# x86 Assembly

## Basic concepts

- ▶ A **CPU** has few memory locations called **registers** fast
- ▶ Can access to **main memory** locations slow
- ▶ CPUs can be **instructed** to perform different operations on memories instruction set/ISA



# Computer architecture



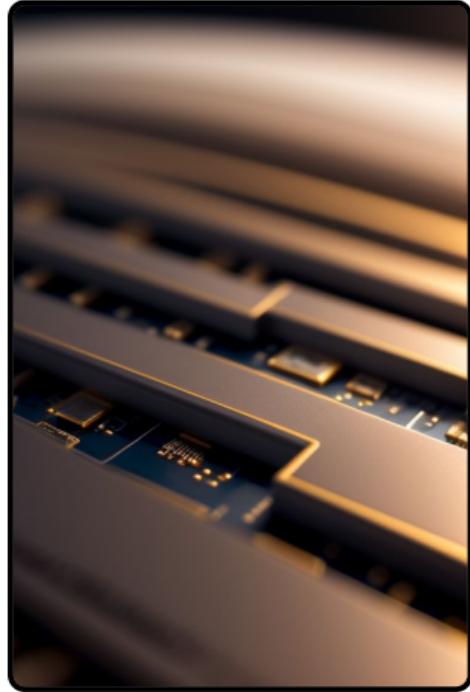
UNIVERSITY  
of York

Department of Computer Science

## Registers

| Description |                     |  |
|-------------|---------------------|--|
| EAX         | Accumulator         | <i>used in arithmetic operations</i>     |
| EBX         | Base register       | <i>often used as a pointer</i>           |
| ECX         | Counter             | <i>used in loop operations</i>           |
| EDX         | Data register       | <i>used in arithmetic operations</i>     |
| ESI         | Source Index        | <i>used in string operations</i>         |
| EDI         | Destination Index   | <i>used in string operations</i>         |
| ESP         | Stack Pointer       | <i>points to the top of the stack</i>    |
| EBP         | Base Pointer        | <i>used as a reference in procedures</i> |
| EIP         | Instruction Pointer | <i>points to the next instruction</i>    |

Also, flag registers and some others.



 x86 Instructions on Wikipedia

What's the value of **eax**?

---

```
mov eax, 0x4 ; eax = 4
mov ebx, eax ; ebx = eax = 4
shl eax, 0x1 ; eax = eax * exp(2,1) = 8
add eax, ebx ; eax = eax + ebx = 12
xor eax, 0x8 ; eax = eax^8 = 4
```



# Endianness

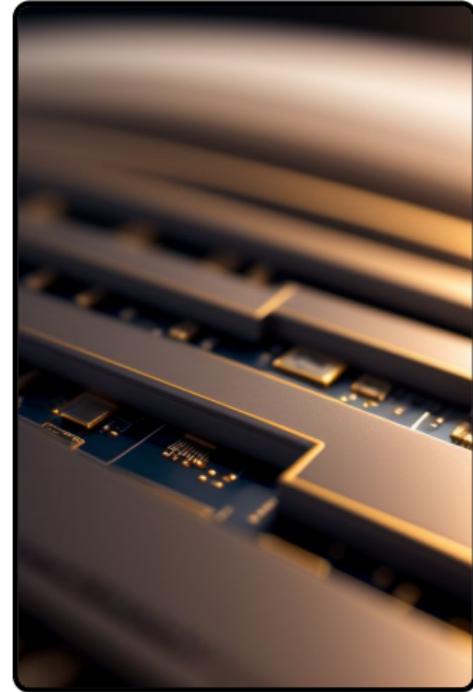
x86 architecture, including Intel and AMD processors, predominantly uses Little-Endian.

MIPS is Big-Endian.

ARM is Bi-Endian (can work with either).

**32-bit integer 0x12345678 is stored in memory as**

- ▶ Little-Endian: 78 56 34 12
- ▶ Big-Endian: 12 34 56 78

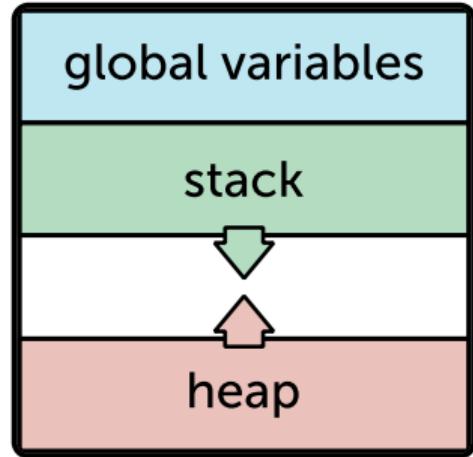


# Principles of Buffer Overflow

# Principles of Buffer Overflow

## Principles

- ▶ (Too) long inputs that overwrite unintended parts of the memory runtime!
- ▶ The severity depends on the source code and the ability of the hacker to adjust to the **right payload**



# Avoid detection

# Obfuscation principles

Make it difficult to understand while preserving functionality

- ▶ code
- ▶ strings
- ▶ control flow
- ▶ data
- ▶ Inserting irrelevant or misleading code (including unnecessary branches)
- ▶ Decrypting values at runtime
- ▶ Metamorphism



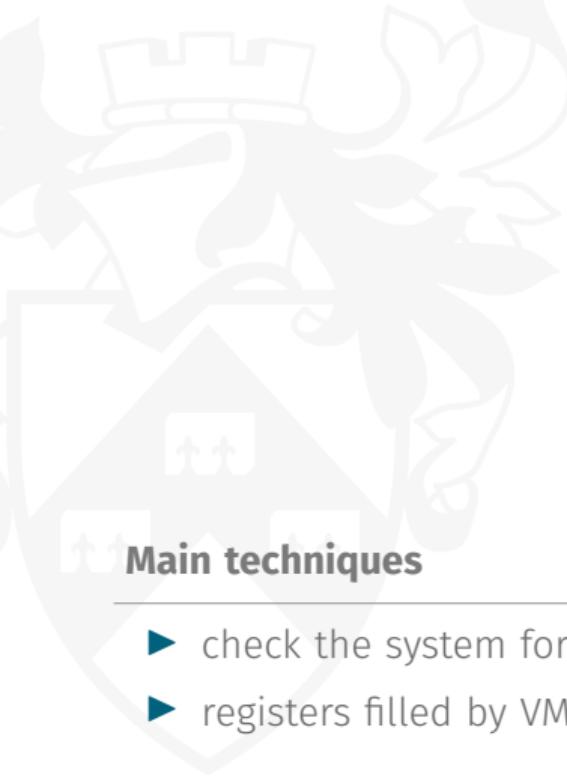
# Anti-debugging techniques

## Main techniques

- ▶ Linux: one process only can call `ptrace`!
- ▶ Windows: built-in `isDebuggerPresent` debug flags in registers
- ▶ detect breakpoints timing
- ▶ self-modifying code that alters its behavior when analyzed
- ▶ fake code paths
- ▶ exception codes that indicate debugger presence



# Anti-virtual environment techniques



## Main techniques

- ▶ check the system for typical VM services
- ▶ registers filled by VMs



# Questions & Answers



<https://padlet.com/robertometere/ehac>

## Further reading and exercises...

- 
-  Reverse Engineering for Beginners by Dennis Yurichev | <http://beginners.re>
  -  Michael Sikorski and Andrew Honig – no starch press (2012)  
*Practical Malware Analysis: A Hands-On Guide to Dissecting Malicious Software.*

# Reverse Engineering, Advanced Techniques

---

EHAC | Lecture 5

Dr. Roberto **Metere**

✉ roberto.metere@york.ac.uk

# Let's start with something that we already know



<https://www.menti.com/alzgpjoi2ppf>

# Objectives

At the end of this lecture you will know about...

- ▶ coding paradigms, interpretation vs compilation
- ▶ disassembly and decompilation

# What is an “object file”?

0 1/3



Department of Computer Science

A file generated as part of the process of compilation. It may contain executable machine code, intermediate bytecode, or other intermediate representations or metadata.

10

An object file is a file that contains machine code, which is the low-level code that a computer's processor can execute.

7

An \*\*object file\*\* is a file containing machine code and data generated by a compiler or assembler during the compilation process. (from deepseek)

7



# What is an “object file”?

O 2/3



Department of Computer Science

Compiled form of source code  
that serves as an intermediate  
step in the compilation  
process

6

That first one is  
Wikipedia

5

An object file is a file that contains machine code or bytecode, as well as other data and metadata, generated by a compiler or assembler from source code during the compilation or assembly process.

5

It contains machine code that can be  
linked to create an executable program

4

.o extension

4



# What is an “object file”?

machine code file generated by compiler

3

An object file is a type of binary file produced by a compiler after compiling a source code file, but before linking it with other object files to create an executable program.

3

Object files basically consist of compiled and assembled code, data, and all the additional information necessary to make their content usable. In the process of building an operating system, you will

3

File with machine code

2

file with machine code

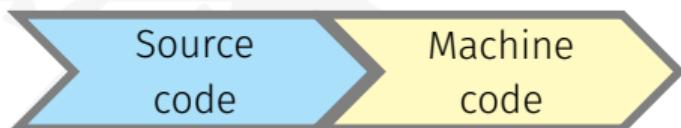


# (Ahead-of-Time) Compilation



UNIVERSITY  
*of York*

Department of Computer Science



- 
- ✓ Compiled code is fast
  - ✗ Compiled code is not portable (platform dependent)
  - ✗ Compilation time can be slow



# (Ahead-of-Time) Compilation | Let's practice

0 ½



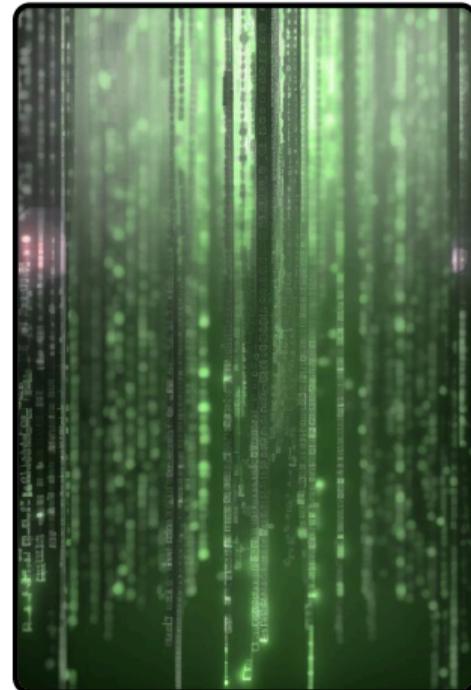
Department of Computer Science

test.c

```
#include <stdio.h>
#include <stdlib.h>
int main() {
    printf("Hello World!");
    exit(0);
}
```

```
$ gcc test.c -o test
$ ./test
```

Hello World!

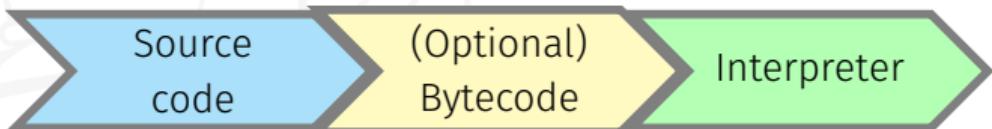


# Interpretation



UNIVERSITY  
*of York*

Department of Computer Science



- ✖ Interpretation is line by line: runtime is generally slow
  - ✓ Compiled code is portable (platform independent)
  - You need a platform dependent interpreter (or VM)

## Just-in-time compilation

- ✓ Can be faster than ahead-of-time compiling
  - Execution can be slow the *first* time code gets compiled

# Interpretation | Let's practice

0 2/3



Department of Computer Science

test.php

```
<?php  
  
echo "Hello World!";
```

```
$ php test.php  
Hello World!
```

A screenshot of a browser's developer tools, specifically the JavaScript console or debugger. It shows a stack trace with several frames from the file 'entropy.js'. The code in the current frame is highlighted in yellow and shows a function named 'activateEntropyHandler' which toggles an 'entropy' variable and animates a section's width. The stack trace includes frames from 'Object' and 'Function' objects.

# Interpretation | Let's practice

0 3/3



c-then-php.sh

```
#!/bin/bash
gcc test.c -o test
./test
php test.php
```

```
$ ./c-then-php.sh
```

Hello World!Hello World!

A screenshot of a browser's developer tools, specifically the JavaScript console or debugger. It shows a stack of frames, with the bottom frame containing a script. The script has several lines highlighted in yellow, indicating they are currently being executed or have been executed. The code includes functions like `activateEntropy()`, `entropy.toggle()`, and `\$(...).animate({width: '100%'}, duration: 150)`.

# Disassembly

0 1/3



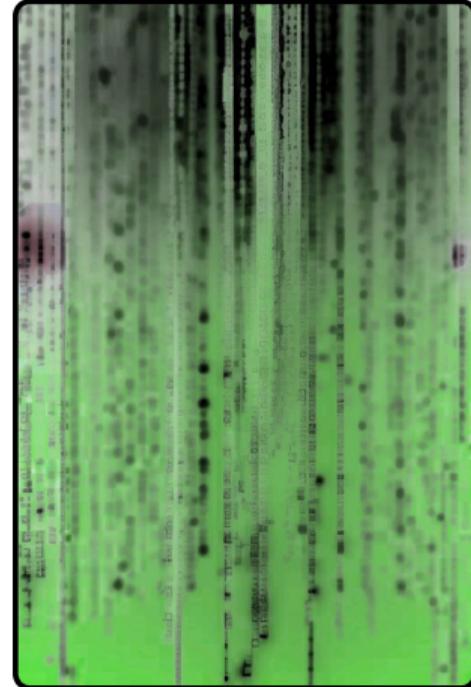
UNIVERSITY  
of York

Department of Computer Science

Machine code → Assembly code

Disassembling an object file allows...

- ▶ a programmer or analyst to understand the flow of the program, analyze individual functions, identify instructions, and **study the logic of the code**
- ▶ for insights into the inner workings of a program, helping with troubleshooting, optimization, and **security assessments**



# Disassembly



UNIVERSITY  
of York

Department of Computer Science

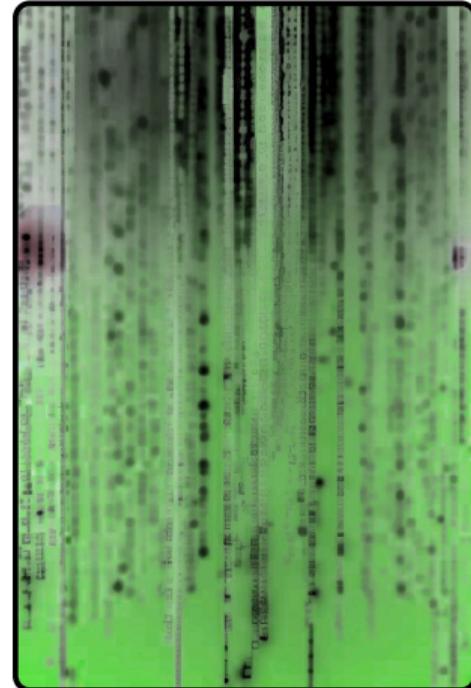
```
$ objdump -d protected
protected:      file format elf64-x86-64
```

Disassembly of section .init:

```
0000000000001000 <_init>:
1000: f3 of 1e fa        endbr64
1004: 48 83 ec 08       sub    $0x8,%rsp
...
...
```

In Windows you can use

```
dumpbin /DISASM{[:[BYTES|NOBYTES]]} <file>
```



# Disassembly

0 3/3

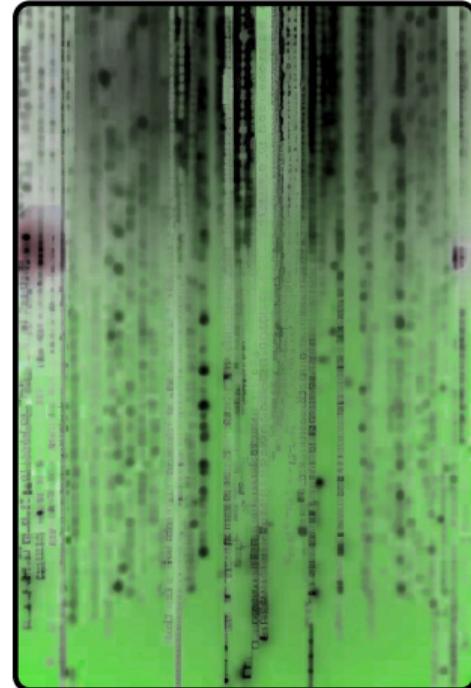
## Advanced analysis tools are available

- ▶ Debuggers GDB and DDD

```
$ gdb ./protected  
(gdb) disas main
```
- ▶ Graphical disassemblers and more IDA, Ghidra, radare2, ...

Let's see together how can we crack the license of our *mock* software **protected**

- ▶ Focus on position 12a2, that jumps somewhere else beyond the call to the function **protectedFunction**.



# Decompilation



UNIVERSITY  
*of York*

Department of Computer Science

Decompilation is the process of translating an object code to source code in a high level programming language.

In other words, given a language, we try to generate a source code, with the idea that, after re-compiling it, we would obtain an object code that is **functionally equivalent** to the original.

*So in general, we cannot retrieve the original source code.*



```
function activateEntropyCurrent() {
    entropy.toggle();
    if (entropy.isActive()) {
        $(`input.entropy`).prop('checked', true);
        $(`section.entropy`).animate({width: '100%'}, {
            duration: 150
        });
        function() {
            $(`input.entropy`).prop('checked', false);
            $(`section.entropy`).animate({width: '0%'}, {
                duration: 150
            });
        }
    }
}
```

# Decompilation | Let's practice



UNIVERSITY  
of York

Department of Computer Science

```
$ java HelloWorld
Hello World $ procyon HelloWorld.class
public class HelloWorld
{
    public static void main(final String[] array) {
        System.out.println("Hello World");
    }
}
```

A screenshot of a browser developer tools window, likely the JavaScript console or debugger. It shows several lines of JavaScript code, with line numbers 37, 38, and 39 visible. The code appears to be a function named 'activateEntropyCurrent()' which toggles the state of an 'entropy' object and animates its width. The code is partially cut off at the bottom.

```
function activateEntropyCurrent() {
    entropy.toggle();
    if (entropy.isActive()) {
        $(`input.entropy`).prop('checked', true);
        $(`section.entropy`).animate({width: '100%'}, duration: 150);
    } else {
        $(`input.entropy`).prop('checked', false);
        $(`section.entropy`).animate({width: '0%'}, duration: 150);
    }
}
```

# Questions & Answers



<https://padlet.com/robertometere/ehac>

## Further reading...

- 
-  Jon Erickson – No starch press (2008)  
*Hacking: the art of exploitation.*

# Forensic Response and Analysis

Introduction – part 1

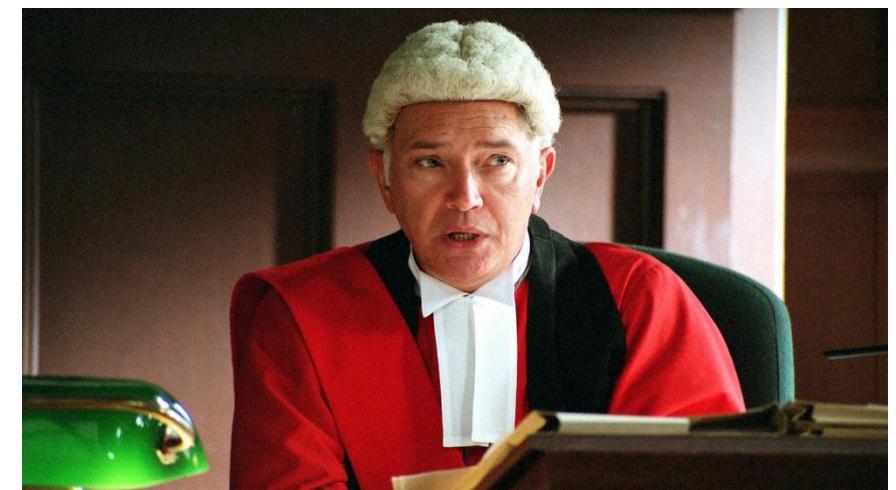
Angus M Marshall BSc PhD CEng FBCS CITP FRSA

# Intro – why Forensic Analysis?

- In order to improve, we need to understand
- When things go wrong, we need to examine the failures and develop ways of treating them for the future
- This needs a systematic approach which maximises the information available to us
- Remember – incomplete information leads to uncertainty

# Footnote: “Forensic”

- Strictly speaking, “Forensic\*” relates to courts and court procedures
- Forensic science is the application of scientific principles and methods to produce evidence which can be relied on court.
- Increasingly, though, “Forensic” is used as a synonym for “Investigative” or “Investigation”, especially in incident response and cyber security.
- We will accept this usage, but remember the higher standard required for court.



So...



- An incident has occurred...
- What happens next?



Or ...?

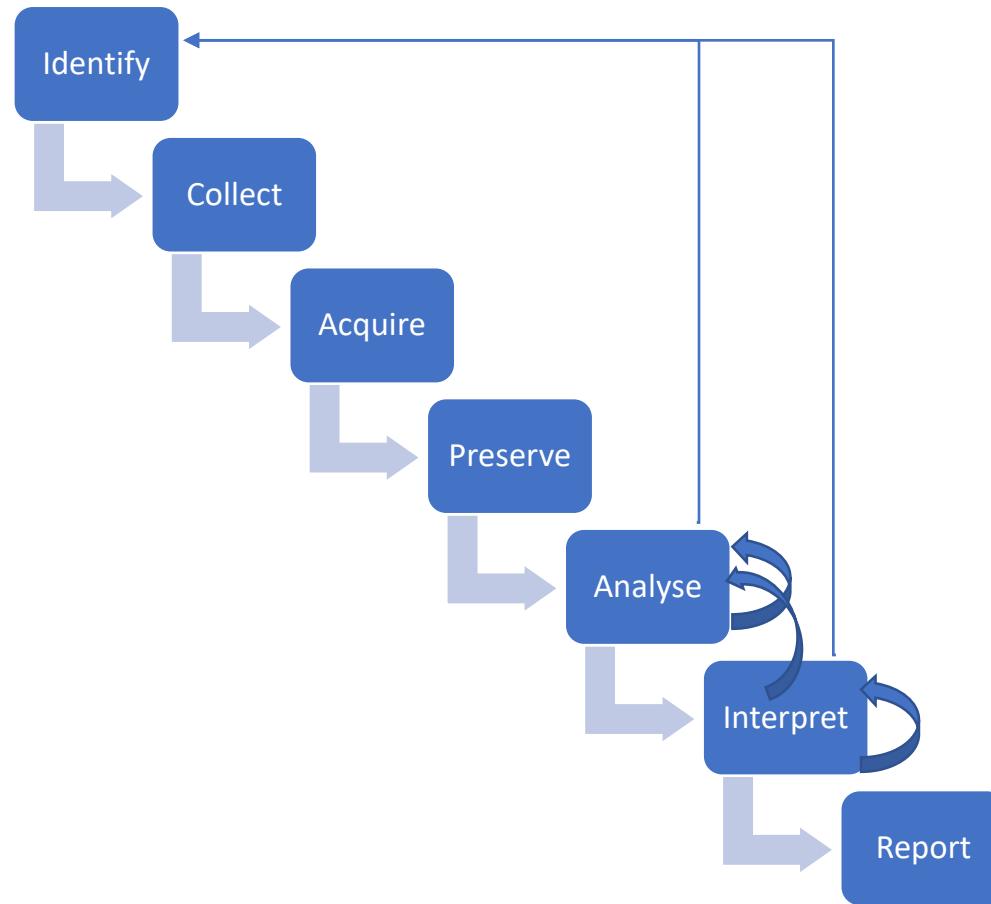
# Implement a response plan

- Conflict:
- Remediation vs. Investigation

# An Investigative Model

ISO/IEC 27037, 27041, 27042, 27043

# Investigation Phases



# Goals

1. Maximise Potential Digital Evidence (PDE) available
2. Minimise potential for accusations of tampering ("*spoliation*")
3. Comply with relevant procedures & standards
  - internal and external
  - Intelligence gathering vs. evidential
  - criminal case. Vs. civil
  - internal investigation
  - eDiscovery process
  - Potential for escalation to a higher level

# Wait a minute!

- *Potential* digital evidence?
- Why not digital evidence?

# Think about a physical crime scene



- How do you know what is evidential and what isn't?
- How do you establish the locus?
- How easy is it to move the boundary outwards?

# Evidence is a product of analysis

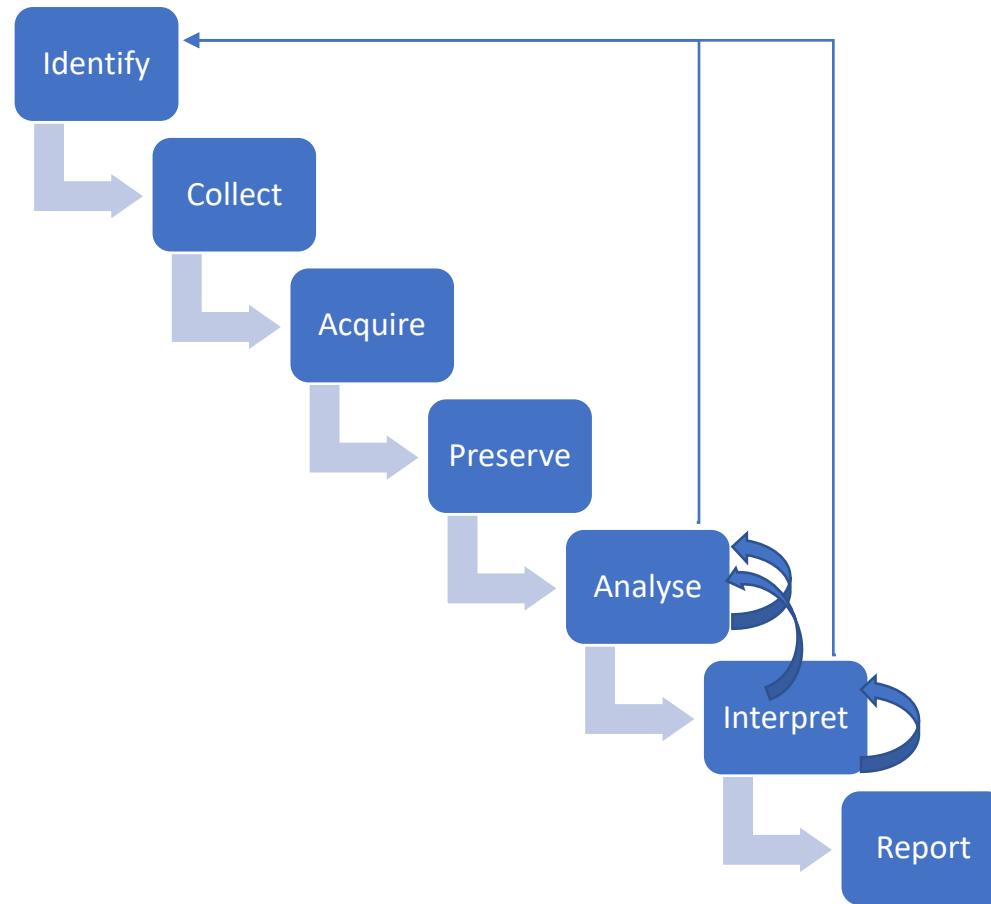
- Analysis allows us to identify which POTENTIAL evidence is relevant to the investigation – turning it into evidence
- We don't have time to analyse everything as we find it, so we aim to preserve as much potential evidence as possible, and analyse it later.
- Analysis & interpretation often identify other potential evidence that can help – if we haven't preserved that, it may be too late.



*"Whoa Dammit!!!"*

We're getting ahead of ourselves  
Let's start again.

# Investigation Phases

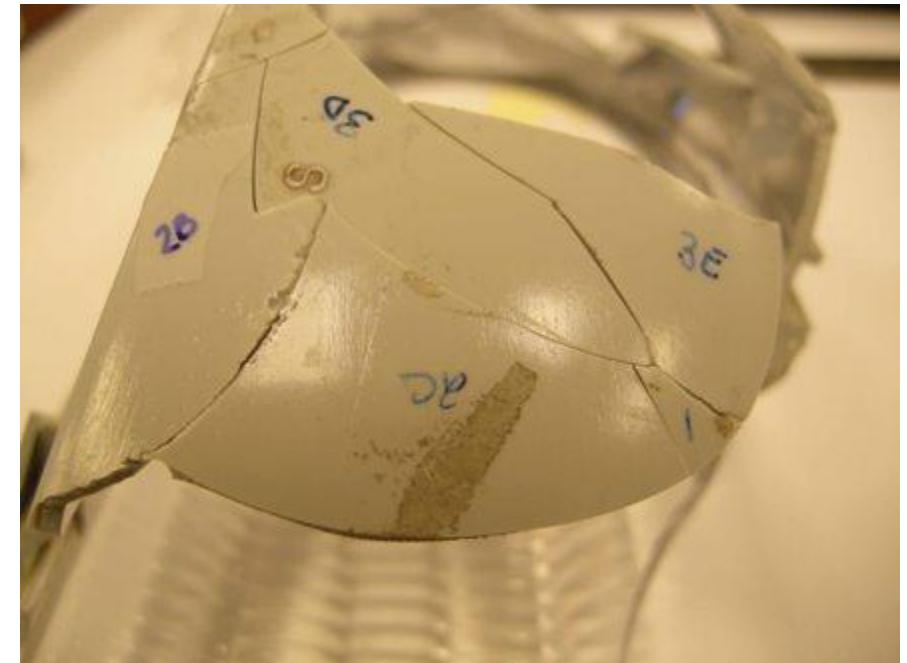


# Identify

- Identify sources of potential digital evidence
  - Typically, where data which may have evidential value is stored
- Locard's principle: "Every contact leaves a trace"
  - When two things come into contact a bit of each transfer to the other
  - Fibres, fingerprints, DNA, splinters, GSR etc.
  - Marshall's digital corollary: "but not necessarily for very long – if at all"
- Physical evidence "speaks for itself" – it is what it is
  - The explanation for why it is where it is makes or breaks the case

# The very best evidence

- “Physical fit”
- Can we do this with digital systems?



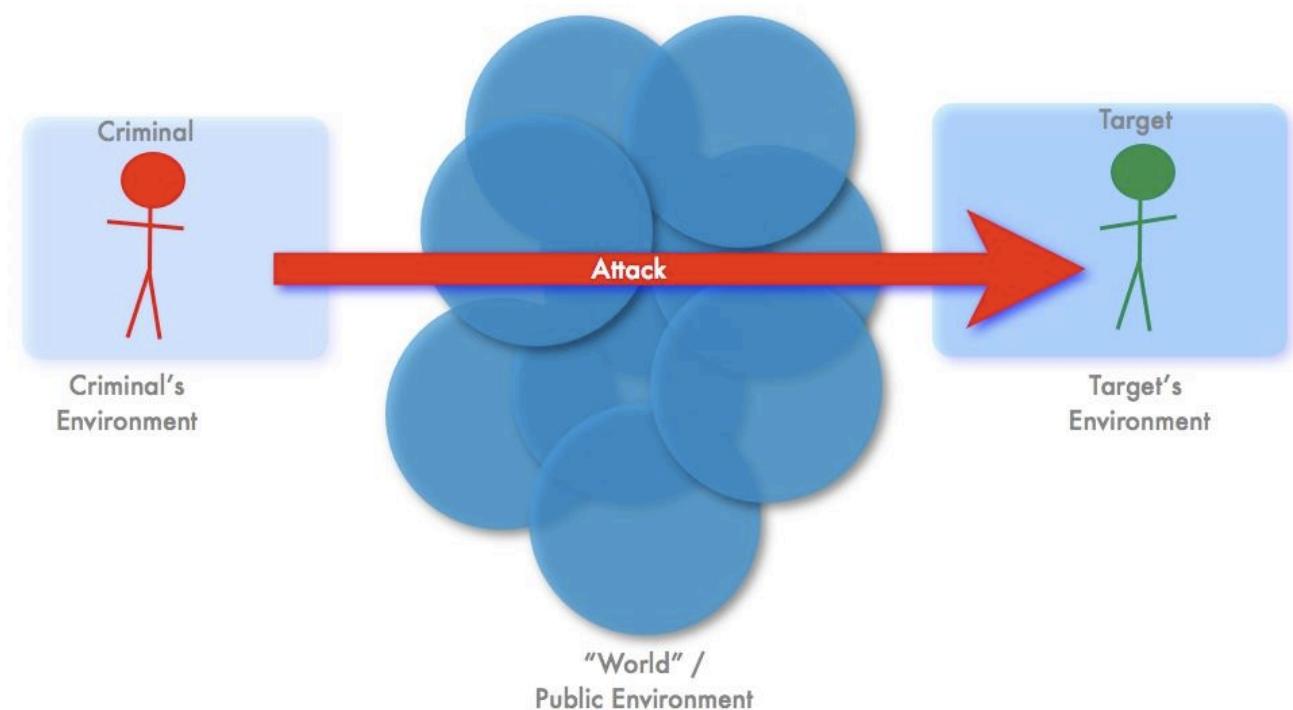
# Digital Evidence challenge

- We can't do “physical fit” and we can't rely on Locard
- Therefore we rely heavily on expert opinion & experience.
- Everything produced as evidence can be argued about.
- BUT – digital sources can be rich in potential evidence, the first challenge is identifying where it is

Thinking about incidents (or  
crimes)

# A little bit of criminology

- Victimation theory: a crime can only happen when a motivated & capable criminal comes into contact with a suitable victim, in the absence of a suitable guardian



# Can help to think in terms of the roles played

- Witness – observes the crime
- Tool – helps the criminal to do something
- Accomplice – is part of the gang
- Victim – is directly affected
- Guardian – was supposed to deter/prevent the crime

Which, if any, of these may have potential evidence of the incident?



# Exercise 1

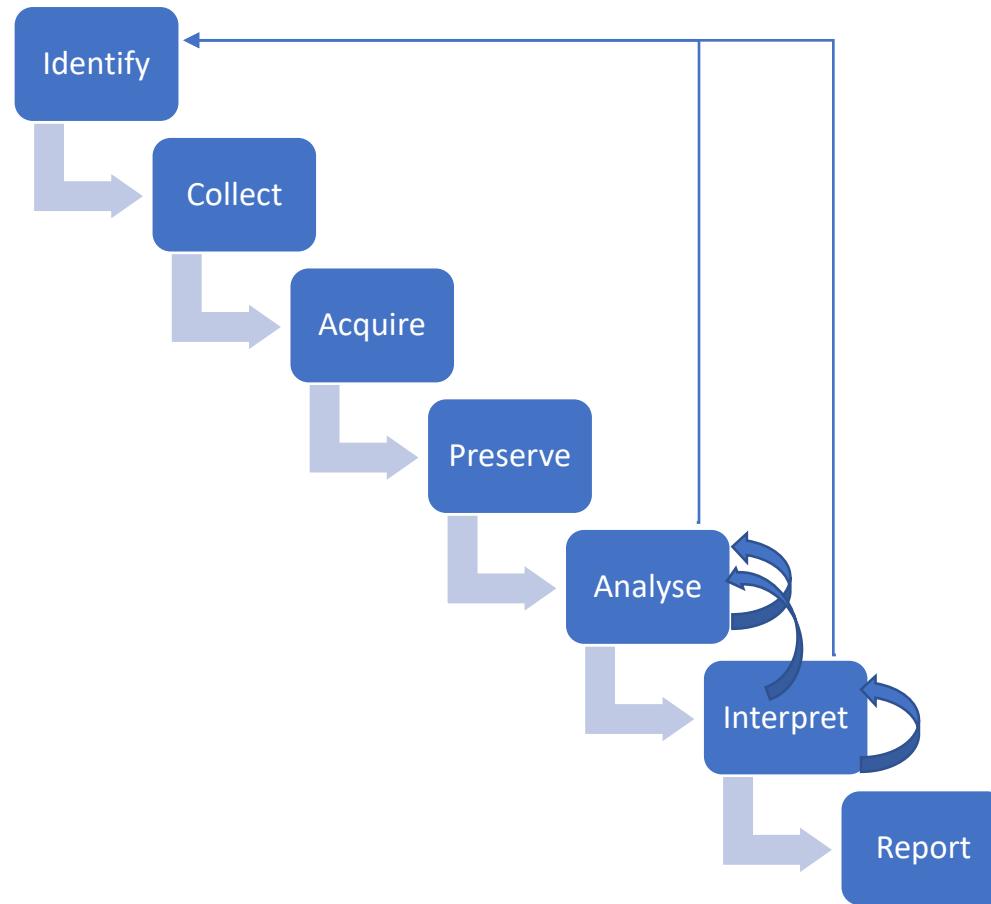
Messages appearing on social media indicate that an A level exam paper has “escaped” from the exam board’s cloud storage system and is being offered for sale on the dark web.

- Consider how this might have happened, and identify the various system elements in the scenario and the roles they fulfil.
- State the nature of any evidence that each might hold, and give an indication of how long its lifespan might be.

# Identification

- If risk assessment has been done correctly, the asset register can also help by listing
  - Devices
  - Processes
  - Data sources
  - Data stores
  - Network elements
  - etc.
- The more we understand about which devices exist and what they do, the quicker we can locate them

# Investigation Phases





# Data Types & Artefacts

Evidence Type

Location / Use



Deleted Files

MFT entries (NTFS), file carving

Metadata

File timestamps, authoring tools

Registry

USB usage, software installs, user actions

Event Logs

Logon/logoff, time changes, system events

Ink Files

Recently opened files, file locations

Recycle Bin

Deleted files & metadata (\$I and \$R files)

# Collect

- Collect, in ISO/IEC 27037, refers to physically taking control of the device in order to extract data from it.
  - “dead box” forensic processing
  - Implies seizure & removal of the device
- What are the problems with this?

# Powers to seize

- In order to seize – you may have to search.
  - Do you have authority to do that?
  - Are your search & seize powers restricted by policy? law? wording of the authorising document?
  - Can you take personal devices or only corporate?
  - Are your actions proportionate?
    - i.e. is your action likely to cause further damage to innocent parties?

# Categories of evidence source

- In practice, there appear to be 3 distinct categories of evidence source
- “Principle 1” devices → Safe to image (HDD/USB)
- “Principle 2” devices → Risky to image (mobile phones)
- Remote sources  
↳ cloud or online services

# Principle 1? Principle 2?

- Named after the ACPO (now NPCC) principles.
  - Found in the ACPO Good Practice Guide
  - Generally agreed worldwide as the foundation of good practice

# ACPO Principles

- **Principle 1:** No action taken by law enforcement agencies or their agents should change data held on a computer or storage media which may subsequently be relied upon in court.
- **Principle 2:** In exceptional circumstances, where a person finds it necessary to access original data held on a computer or on storage media, that person must be competent to do so and be able to give evidence explaining the relevance and the implications of their actions.
- **Principle 3:** An audit trail or other record of all processes applied to computer based electronic evidence should be created and preserved. An independent third party should be able to examine those processes and achieve the same result.
- **Principle 4:** The person in charge of the investigation (the case officer) has overall responsibility for ensuring that the law and these principles are adhered to.

# Paraphrasing

- Principle 1: “Thou shalt not risk damaging the original data”
- Principle 2: If you can’t avoid potentially damaging the data, be sure you’re qualified & competent, and be ready to explain your actions
- Principle 3: Document your process so that someone else can repeat it. - ***CONTEMPORANEOUS NOTES, CHAIN OF CUSTODY, CONTINUITY OF EVIDENCE***
- Principle 4: The case officer (or investigation manager or similar term) must make sure that these principles and the law (and policies) are followed.

# Chain of custody

- Who had control of the item?
  - Who/where did they get it from? When?
  - Who did they hand control over to? When? Why?
  - What happened to it whilst someone had control?
- 
- A break in the chain = a time of unknowns = scope for tampering

# Continuity of evidence

- Essentially the same as chain of custody, but add
  - Where did new evidence come from (result of processing)?
  - At what time?
  - Who produced it?

# So, back to device categories

- Principle 1 – data can be extracted without risk of damage (e.g. hard discs, SSDs, USB devices etc.).
  - May be capable of being seized and taken away.
- Principle 2 – extraction process creates a risk of damage (typical of mobile phones & tablets).
  - May only get a partial copy. May only get a “logical” copy instead of a physical.
- Remote devices – the physical device is not accessible, but data can be accessed across a network. (e.g. Cloud, social media)
  - Probably only ever get a logical copy – and may not get the same view that the user or controller get.

# So –”collect”

- How realistic is it?
- When can it be used?
- When should it be used?



## Exercise 2 – in teams of 3

- You have been asked to assist in the seizure of devices from a suspect's office.
- When you arrive, you notice that one of the computers appears to show Veracrypt active, accessing an encrypted volume. You know that, if left inactive for a time after the unlock password has been entered, will “forget” the password and lock the drive again. You can see a partial listing of the encrypted volume that suggests the files in it are particularly relevant to the investigation.
- What do you do?
- How do your actions comply with the ACPO principles?

# Triage

- Principle 2 process
  - A fast examination of the original PDE source to determine if it looks like it contains relevant material
  - Automated triage typically relies on file signatures (e.g. hashes of known bad files – typical in cases involving imagery of child abuse)
    - Requires hash sets to be produced
    - Can also use “exclude” sets (e.g. NIST known file sets) to reduce final results.
    - cf “de-NISTing” (or de-KUFing) in eDiscovery
  - Manual triage – skill and experience of the operator

# Source Assessment

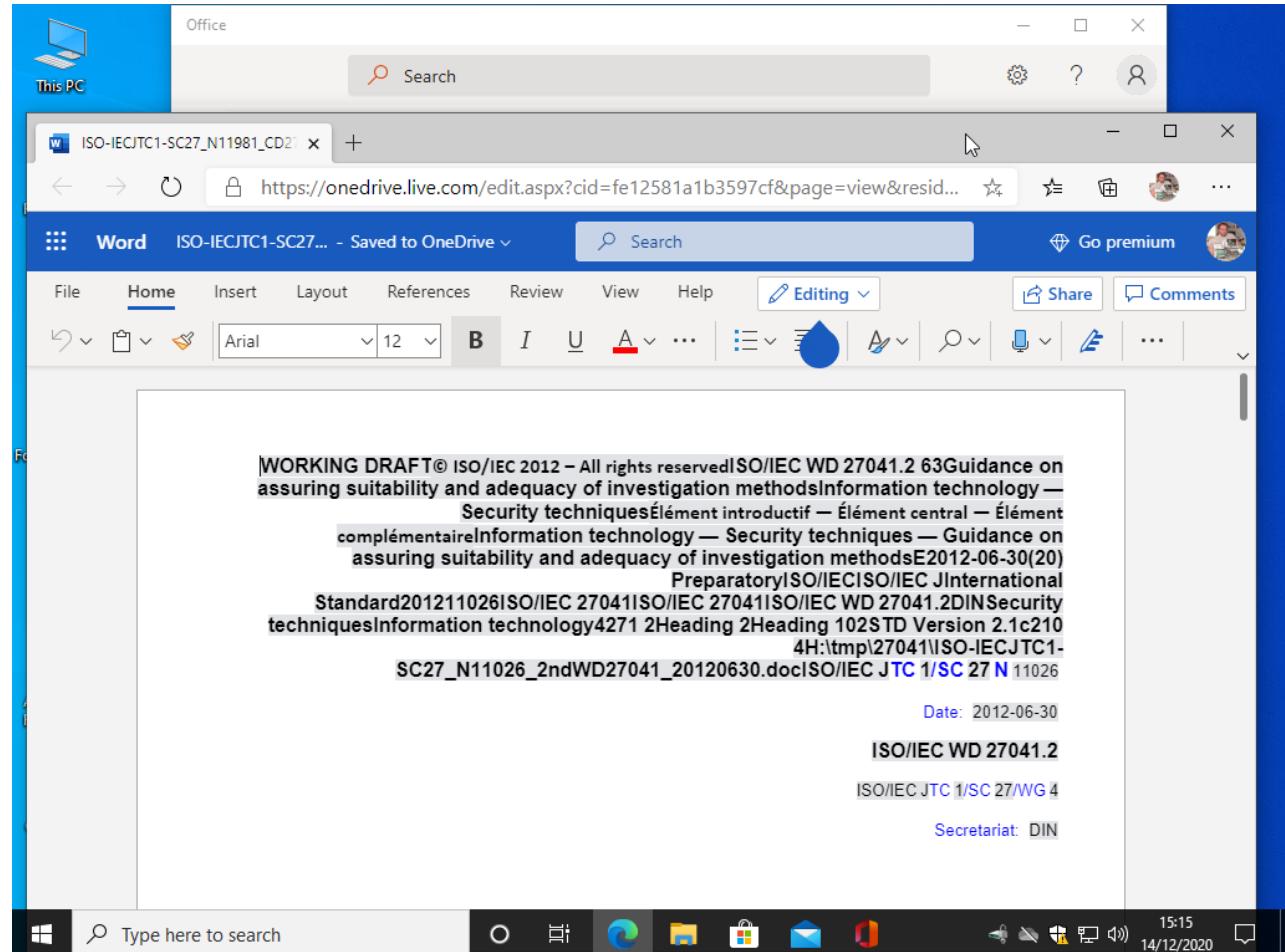
- Required to ensure compliance with ACPO principles
- Follows “crime scene” assessment principles
  - Visual inspection first
  - Then assessment of storage devices

# Source Assessment : Visual

- Start : record detail of the item
  - Type
  - Markings
  - Serial Numbers
  - Anything connected, attached, inserted
  - Use sketches & photographs

# Visual assessment – challenge

- Write a description of this machine, based on the screenshot.



# Source Assessment

- After recording details of the device :
  - Identify storage devices
    - Floppy drive
    - CD/DVD
    - Removable memory devices
  - Check for media

# Source Assessment

- Catalogue media & package appropriately for later examination

# Scene Examination Statement

- Date of examination
- Information provided before your examination of the scene
- Description of the scene
- Description of relevant evidence
- Any interpretation or opinion
- Recommendations for further work

# Source Assessment

- If necessary, using appropriate tools, open the device to
  - identify internal storage
    - hard discs etc.
  - carefully remove them, recording details of connections, serials numbers, settings etc.
  - package for later examination.

# Lab. Statement/ Report Headings

- Receipt of Items (not always appropriate)
- Information/ Background/ Circumstances
- Purpose
- Technical issues
- Examination and Results – a logical order
  - Subdivided into sections
  - Items relating to individuals
  - Items relating to scene/s
  - Ordered relating to best evidence
- Interpretation of Findings
- Conclusion/s

# Other Inclusions

- Use of Assistants
- Disclosure
- Evidence submitted but not examined
- Reference to scene photographs or statements from other scientists

# Collect

- Often seen as “the gold standard”
- Generally desirable if devices can be physically accessed and removed
- Allows state to be preserved and re-acquisition & examination to be carried out
- Drives can be extracted and imaged at will.
- BUT – not always possible, so it may be necessary to
  - Leave the device alone (documenting the reasons for doing this)
  - Go straight to acquisition



# This week's lab.

- You will be processing a simple crime scene to collect a PDE source and then carrying out a physical source assessment of that PDE source.

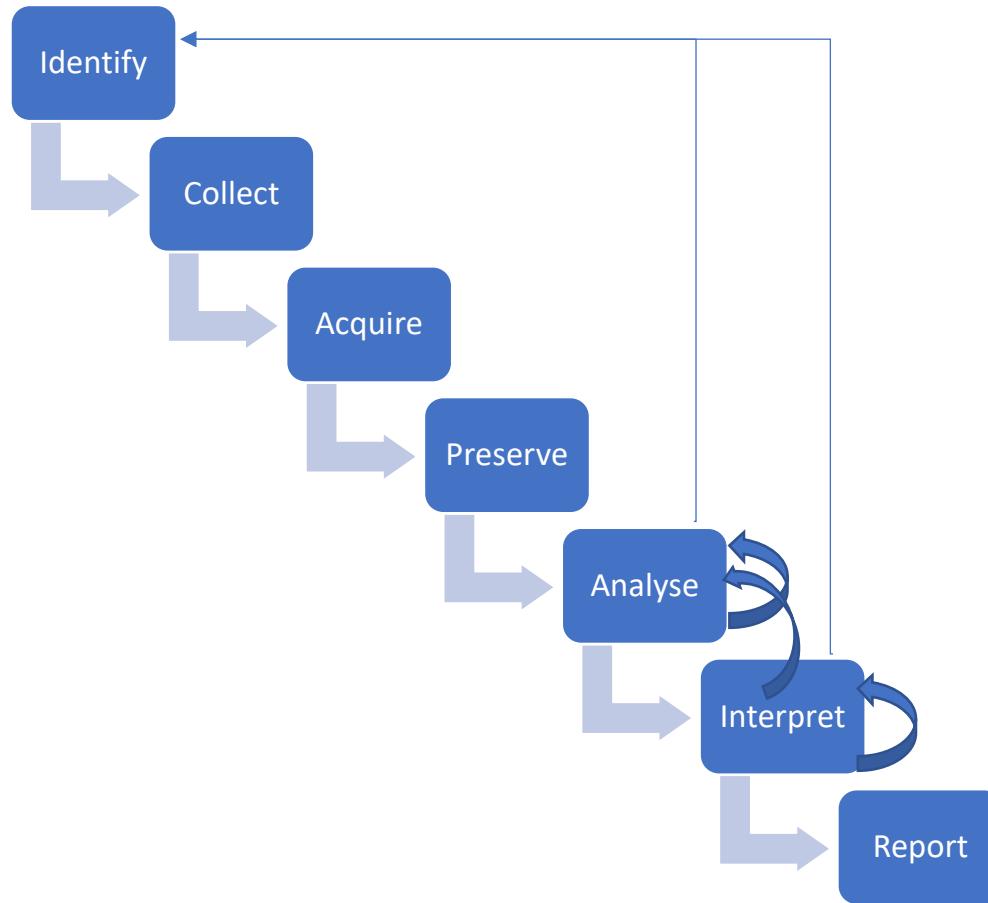
# Forensic Response and Analysis

Part 2

# Collect

- Often seen as “the gold standard”
- Generally desirable if devices can be physically accessed and removed
- Allows state to be preserved and re-acquisition & examination to be carried out
- Drives can be extracted and imaged at will.
- BUT – not always possible, so it may be necessary to
  - Leave the device alone (documenting the reasons for doing this)
  - Go straight to acquisition

# Investigation Phases



# Acquire

- The process of extracting data from a source in order to produce a copy which can be subjected to further processing. (aka “imaging”)

# Copying storage devices?

Going beyond ^C^V and drag ‘n’ drop

# Examining devices

- Applying ACPO principle 1 means that we usually have to produce a verifiably accurate copy of the device to be examined
- How ?

# Old method

- When HAL was less-common
- Device to device copy using an IDENTICAL target device
- Not always successful
  - subtle difference between devices (bad block positions etc.)

# Current Method

- Apply streams concept and perform a bitwise copy
  - Contents of device copied to some medium
    - file, partition etc.

# Why is this valid ?

- Interpretation of data is independent of hardware
  - Bitwise copy can be shown to be equivalent to the device at all levels above device driver
  - stream is already an abstract view of the device

# Verifying equivalence

- Compute hashes
  - a mathematical operation using all data in the input
  - provides a near-unique “signature” for the stream
  - very low probability that any two different streams will generate the same hash
  - e.g. MD5 gives 128 bit = 1 in  $2^{128}$  chance of collision

# Proposition

- $\text{hash}(\text{device}) = \text{hash}(\text{copied stream}) \Rightarrow$  copied stream can be used as if it is the device for investigation
- Is this acceptable ?

(But Angus, MD5 collisions can be manufactured – what about that?)

# But what if we can't copy the whole device?

- Existing damage
- Unable to access (e.g. cloud storage)
- Social Media
- Best efforts – logical copies, contemporaneous notes and hashes of whatever we do get. (Pragmatic view – some PDE is better than none).

# Acquisition

- Considerations
  - Modern O/S have a bad habit of auto-mounting any storage device attached to them
    - What effect does this have? (in an evidential context)
    - Can it be prevented?



# Write blockers

- NIST requirements
  - The tool shall not allow a protected drive to be changed.
  - The tool shall not prevent obtaining any information from or about any drive.
  - The tool shall not prevent any operations to a drive that is not protected
- Don't have to be hardware
  - Commercial software
  - Open source software
  - Customised O/S (registry hacks, kernel mods., disabling services)

# Question

- Apart from storage devices, should we consider imaging anything else?

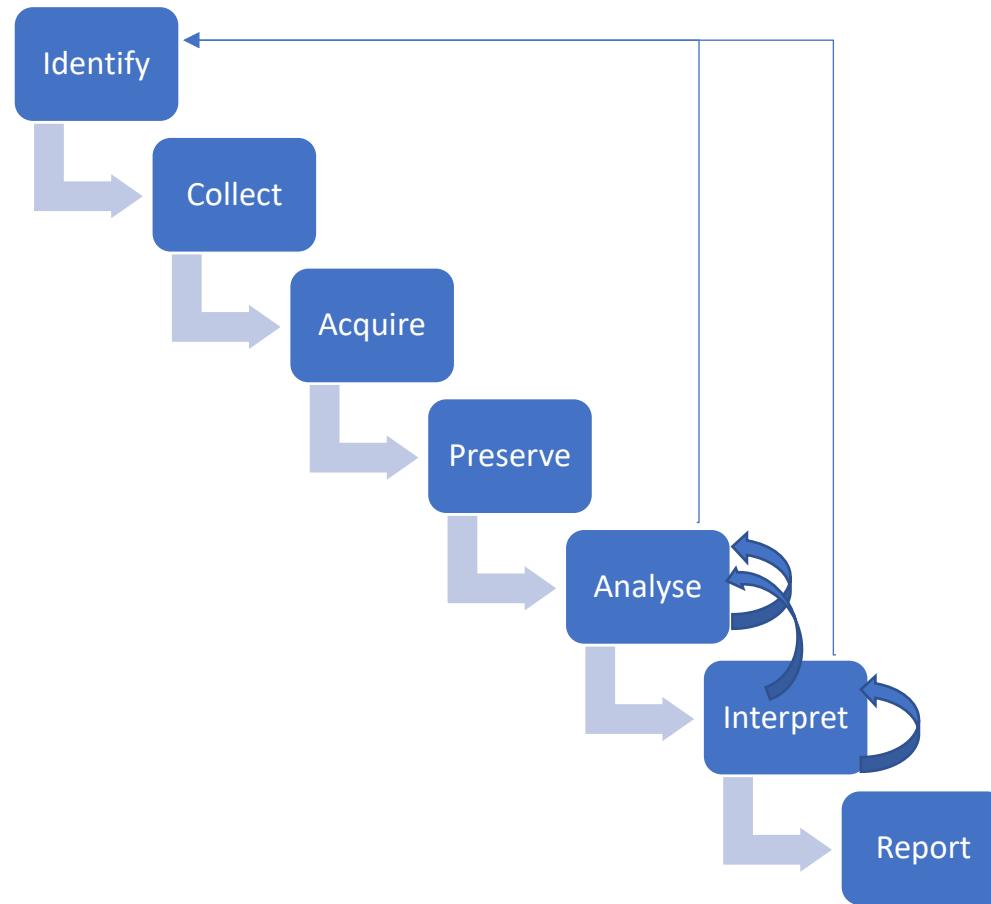
# Physical vs. Logical copies

- Physical – the copy is considered identical to the original. All data on the device is present and can be accessed – including deleted data, meta-data, hidden areas etc.
- Logical – the copy is as complete as the device will allow us to get. May not represent how the data are actually stored on the device. Probably does not include deleted data or inaccessible areas. For cloud storage “de-duping” guarantees that the logical view does not match the actual storage.

# Exercise

- Consider Snapchat.
- A 13-year old girl has been chatting to a stranger. The stranger has sent her an unsolicited picture of a naked adult male.
- What could be done to preserve the potential evidence in this situation?

# Investigation Phases



# Preserve

- Maintain the image in such a way as to avoid damage/spoliation
- Usual data preservation rules apply – redundancy, backups etc.
  - With hashing
  - Write-once media should be considered
  - Generally work on copies of the initial image (“working copy” vs. “master copy”)
- How long to preserve for?

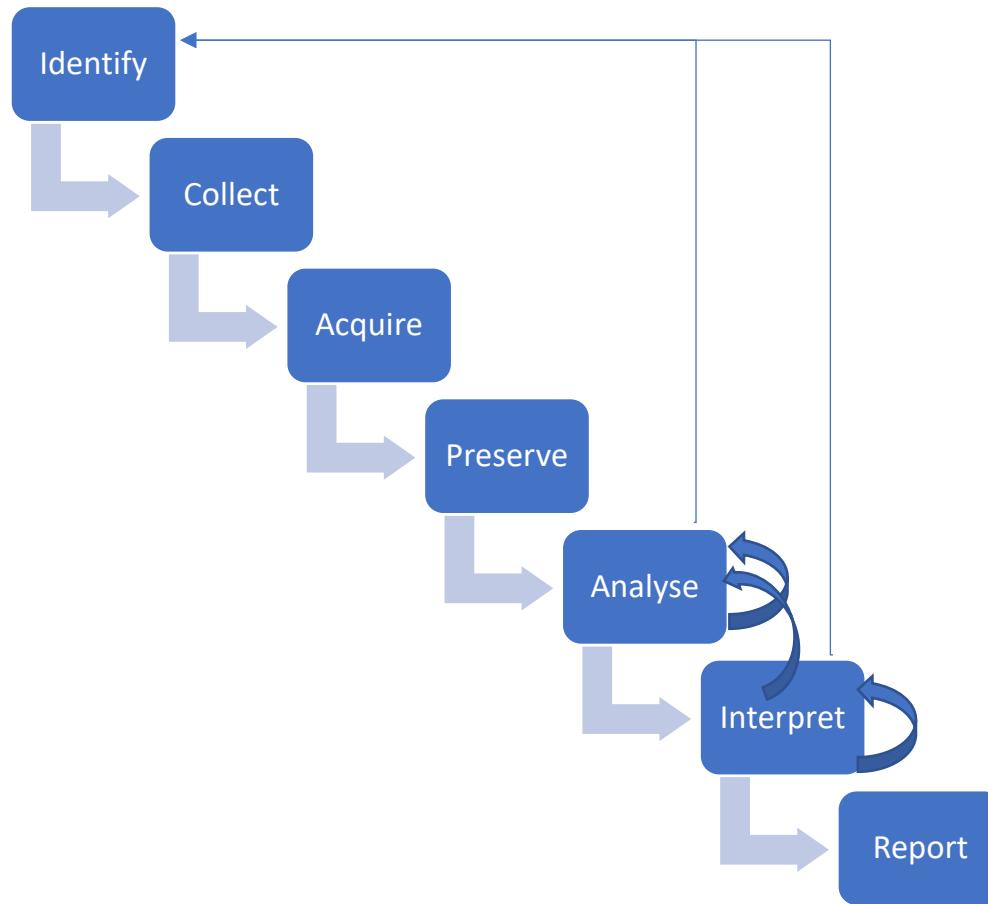
# Question

- In a case where a single employee has been accused of illegal downloading, you have been asked to image a RAID array connected to a server, which is critical to the company's operation. The imaging process is likely to take 24 hours to complete and the server cannot be taken offline.
- What do you do?

# Question

- If you have to run investigative/imaging software on the suspect system, can you trust the software?

# Investigation Phases



# Analysis

- Processing the potential digital evidence, to determine which (if any) is relevant to the investigation.
- Requires knowledge of system architecture, software behaviours, user behaviours, system component interactions...
- Let's spend some time on those...

# Aim

- To understand how data appears on a system, and to consider how its relevance to an investigation can be determined

# ABC

- Assume nothing
- Believe nothing
- Challenge everything
- Pay attention to detail & search for corroboration

# Investigative Evaluation

- What is known
- What is not known
- Consistencies
- Conflicts

Remember

A.B.C

# 5WH formula

- Recognised process to aid investigators develop an investigative mindset
- So what is this formula?

# What, Where, Who, When, Why ,How

# Aide memoire

| What do we know at the moment? | What information do we need to find out? (5WH) | How are we going to go about finding this information? |
|--------------------------------|--|--|
|                                |  |  |
|                                |  |  |
|                                |  |  |
|                                |  |  |
|                                |  |  |
|                                |  |  |
|                                |  |  |
|                                |  |  |
|                                |  |  |
|                                |  |  |

# Fundamentally

- Data doesn't just "appear"
  - It has to be stored, deliberately, as a result of some human-directed process.
  - The human in charge may be authorised or unauthorised
  - It represents the results of some activity
  - That activity may be relevant or irrelevant

# Data deposition categories

|                          | <b>Knowing</b> | <b>Unknowing</b> |
|--------------------------|----------------|------------------|
| <b>Authorised user</b>   |                |                  |
| <b>Unauthorised user</b> |                |                  |



# Exercise

- Suggest which of the 4 categories each of these belongs to:
  - A word processed document saved in a user's "Documents" folder
  - A cached copy of an image from a website
  - A log file entry showing that a user logged in to their account
  - Malware introduced to a system through use of a compromised removable storage device
  - Ransomware installed through clicking on an attachment in email

# Assumption (or we'll be here until October)

- You know enough about disk organisation and file system to understand partition tables, File Allocation Tables/file indexes etc.

# Examining filesystems

- Using a tool and pressing the “find evidence” button is not enough.
- You need to be able to
  - Check the tool’s results
  - Know how well the tool works
  - Explain the data found
  - Spot discrepancies and deal with them
  - Deal with incomplete data (fragments)

# Key File System Operations : User View

- **CRUD**

- Create
- Read
- Update
- Delete

# Create a file

- Identify Free space for the file on the device
  - Check file size against free list  
OR
  - Attempt to write and find insufficient space (?)
- Mark units as used and add entry to the file system table

# Create a file

- Identify Free space for the file on the device
  - Check file size against free list  
OR
  - Attempt to write and find insufficient space (?)
- Mark units as used and add entry to the file system table

# Read a file

- Locate filename in the index/directory
- Locate allocation units
- Read data from AUs and present back to O/S
- Record time of reading in the index

# Update a file

- Locate current position of file
- Write the new contents into the space already occupied, extending the file if necessary
- Update the Modification time in the index

# Deleting a file

- Locate the file
- Mark the allocation units as free (change state in bitmap, add to free list etc. etc. etc.)
- Mark file as deleted  
(e.g. by changing first character to FF)
- BOCTAOE...

# Journalling Filesystems

- FAT, ext2 and most other filesystems are “direct write” systems
  - i.e. data is written to the disk ASAP.
- An alternative is to use Journalling
  - “delayed write”

# Journalling FileSystems

- In addition to writing data & file meta-data to the disk, the filesystem maintains a “journal” of events in the life of the files.
  - Future - “I’m about to do this to the file”
  - Historical - “I did this to the file”

# Why Journal ?

- File writing needs two operations - write the data, write the meta-data about the file (times etc.)
- If the system crashes, something can be lost
- Journalling records what should have happened, and whether or not it did happen...

# Logical Formatting

- Sits above the Physical Formatting
- Groups sectors into filesystem units :
  - Allocation Units, blocks, inodes
  - Adds a filesystem information area
    - Free units, used units, file details
  - Unique to each filesystem

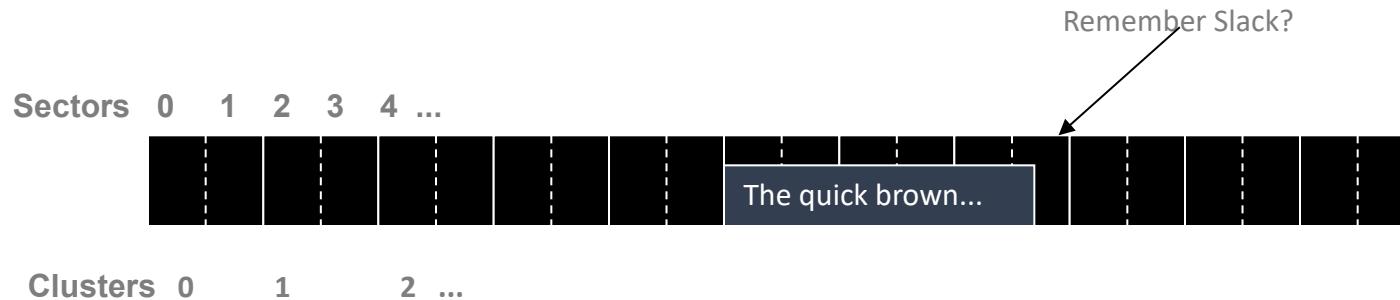
# File System

- Is the total collection of files available to the computer system at any time
- Across all connected storage devices
- Managed by the Operating System

# Logical FileSystem : EXAMPLE

- Physical uses sectors of 512 bytes & has 80000 available on disk.
- Logical FS can only deal with 20000 units
  - Allocates 4 sectors per unit
    - (1 unit=2048 bytes)
    - Larger SLACK space needing filled...

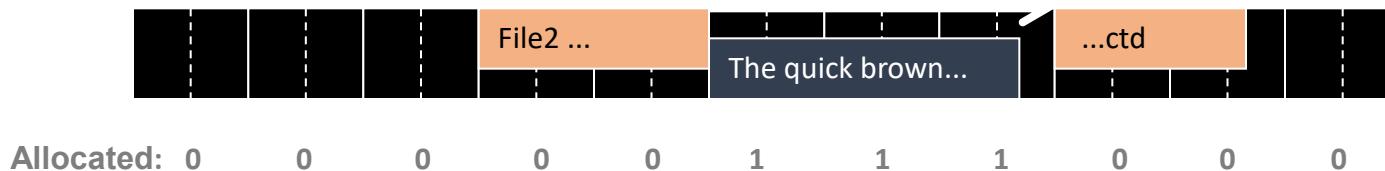
# Clusters



- *Clusters (or blocks)* are a fixed number of sectors, they are the *unit of allocation* for files (usually clusters are 4k bytes – 8 sectors, may be smaller.)
- There may be a separate area for file system management which is not counted in clusters.
- The area available for data may be known as *data area* or *cluster heap*.

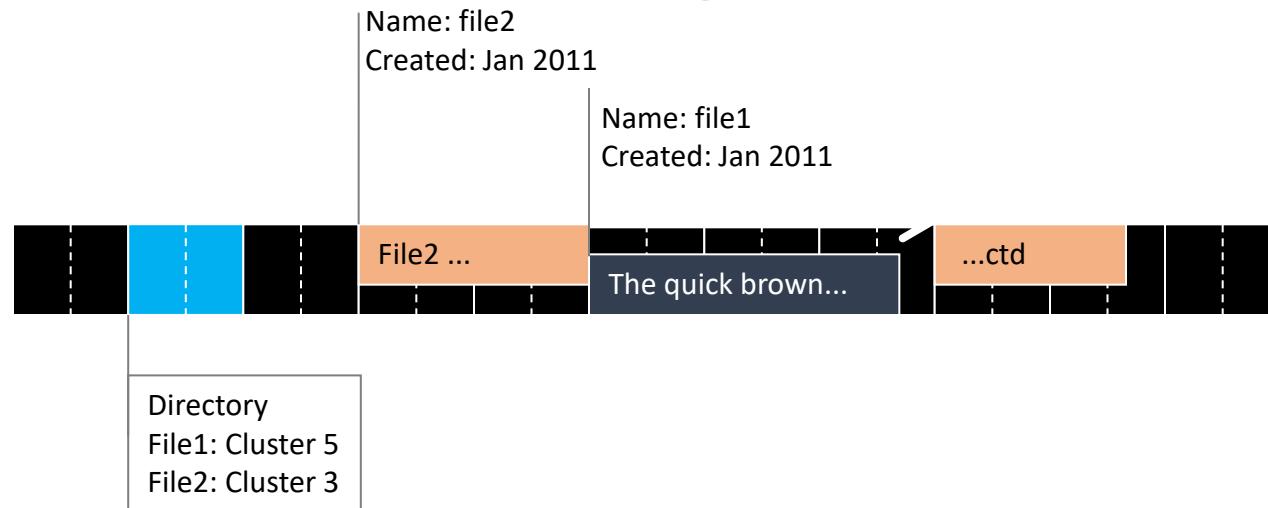
# Allocated Blocks

What is the status of File2?



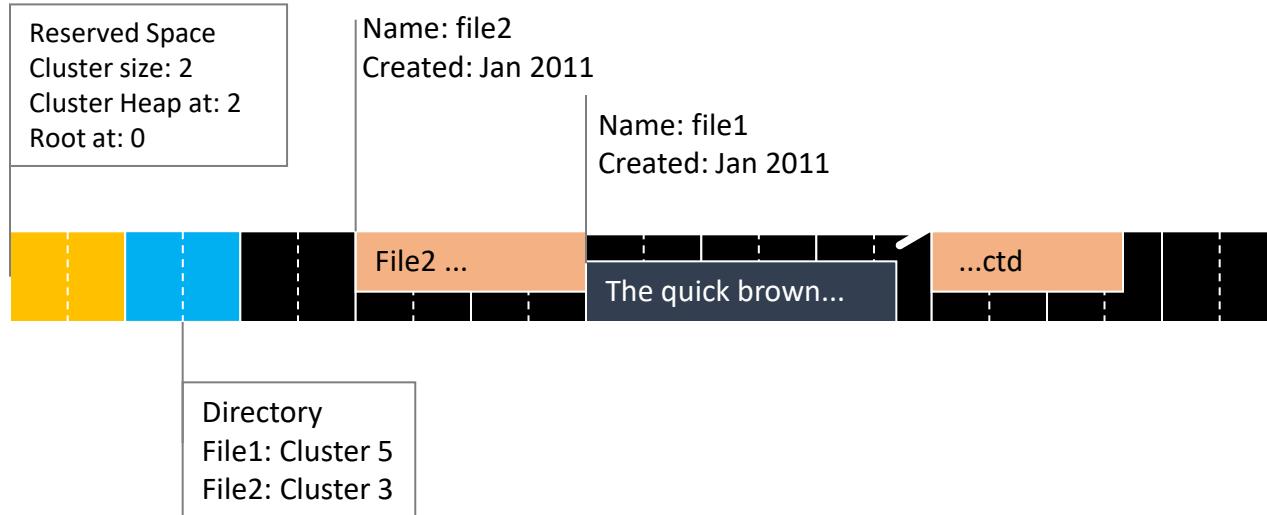
- An *allocation map* tells the file system what space is free.
- A file is allocated a series of *extents*, which are linked to form a single logical *stream*.
- A single file name may be associated with several streams.

# Metadata and Indexing



- File metadata includes date-time information.
- There is usually a ‘root’ directory, and some ‘files’ serve as sub-directories.
- Metadata may be held within a file record, in a directory, or a mixture of both, depending on the file system.

# File System Metadata



- The start of a file system often includes a reserved space for a specification e.g.:
    - Cluster size.
    - Initial navigation (e.g. Root directory location).

# Summary: a file system has

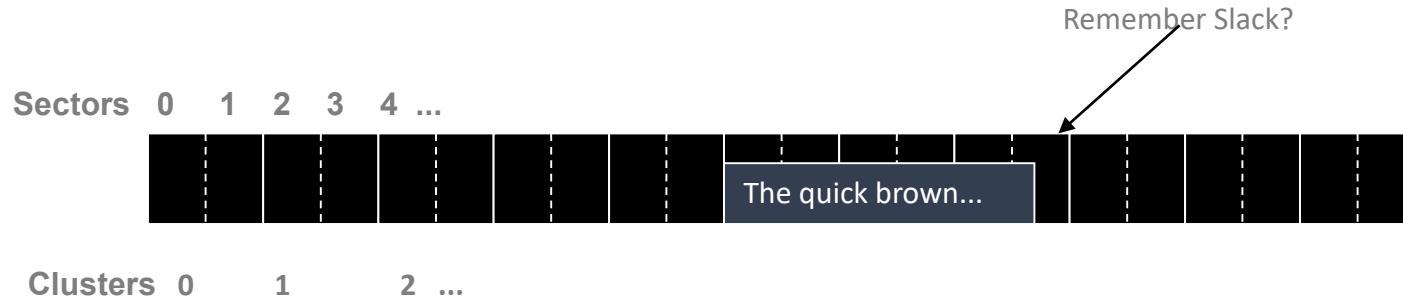
- Information about its configuration.
- A data area in which files are allocated.
- Allocation information.
- Directory or index structures.
- Information to link file/stream extents.
- File metadata.
- File content.

# This week's lab.

- You'll see why verification functions & write-blockers are important
- You'll use a basic forensic tool to look at data on a simple filesystem

As I was saying ...

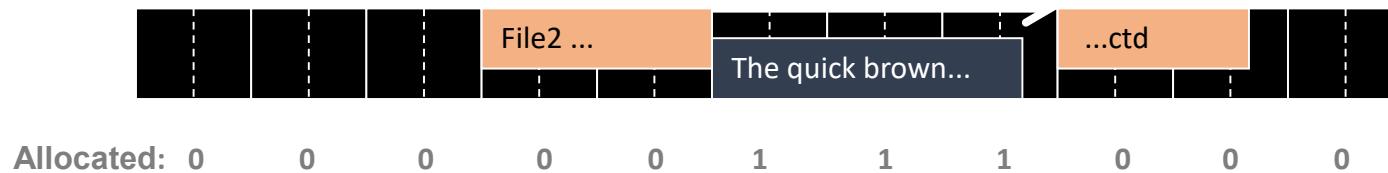
# Clusters



- *Clusters (or blocks)* are a fixed number of sectors, they are the *unit of allocation* for files (usually clusters are 4k bytes – 8 sectors, may be smaller.)
- There may be a separate area for file system management which is not counted in clusters.
- The area available for data may be known as *data area* or *cluster heap*.

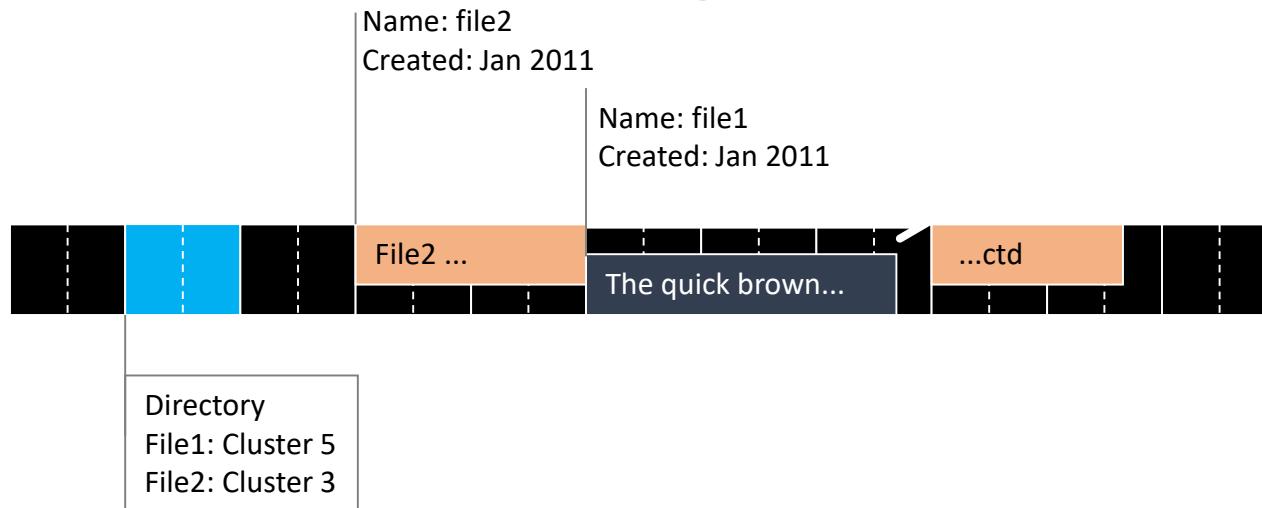
# Allocated Blocks

What is the status of File2?



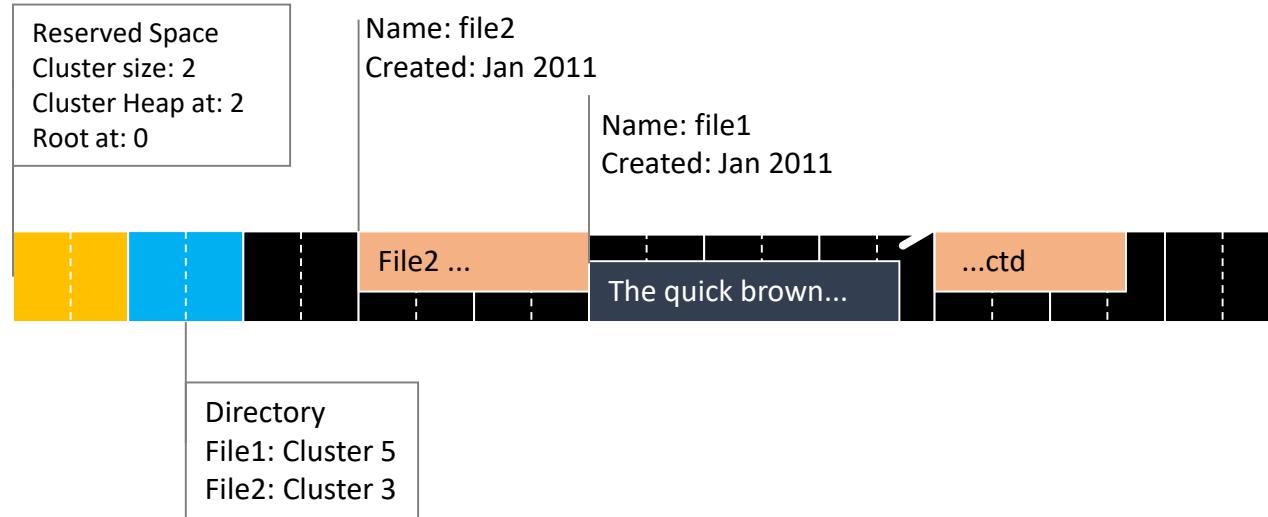
- An *allocation map* tells the file system what space is free.
- A file is allocated a series of *extents*, which are linked to form a single logical *stream*.
- A single file name may be associated with several streams.

# Metadata and Indexing



- File metadata includes date-time information.
- There is usually a ‘root’ directory, and some ‘files’ serve as sub-directories.
- Metadata may be held within a file record, in a directory, or a mixture of both, depending on the file system.

# File System Metadata



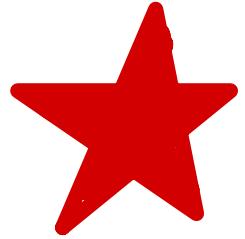
- The start of a file system often includes a reserved space for a specification e.g.:
  - Cluster size.
  - Initial navigation (e.g. Root directory location).

# Summary: a file system has

- Information about its configuration.
- A data area in which files are allocated.
- Allocation information.
- Directory or index structures.
- Information to link file/stream extents.
- File metadata.
- File content.

# Forensic Response and Analysis

Part 3



# Data Recovery

Examining the filesystem present in the image (copy)

Searching for items of interest

# Why

To summarise some of the forensic issues associated with the file system.

To extend our knowledge of file handling and the identification of file use and metadata:

Recognising files which are no longer indexed by the file system.

The ‘deletion’ mechanism provided by operating systems such as Windows (aka ‘recycle bin’).

Other evidence of file use and metadata.

The examples here are specific to Microsoft Windows 7 and later releases; earlier versions of Windows and other operating systems have similar mechanisms but the detail is different.

# Data Analysis

Determining the type of data recovered

Determining likely significance

e.g. files in “My Documents” vs. files in “Temporary Internet Files”

# Data Interpretation

- Deriving meaning from the available data
- What has the machine been used for ?
- When was it used ?
- How did data arrive ?

# File System EVIDENCE

# Dates and Times

- Timelines are fundamental to many cases.
- Different file systems store date-time in different ways:
  - NTFS stores dates in UTC (GMT).  +1 to BST
  - FAT stores times in ‘local time’.
- The usual problem for the analyst is to convert times to actual time, for which you need to know:
  - Any clock offsets in the computer (BIOS, Network).
  - The TimeZone set in the computer.
  - Any daylight saving offsets applied to the timezone.

# Beware of Assumptions

- Consider a UK computer, which was left in the Seattle timezone by the user who simply changed the clock to correct the apparent computer time to the UK.
- Other easy confusions include the date format output by different tools (month/day/year) and forgetting to allow for daylight saving times.

User at keyboard in UK at wall clock time: 1200

Although computer shows correct time (1200)  
because it is set to Pacific Standard Time (PST) it  
records the time in the file system as  $12 + 8 = 2000$   
UTC

Sloppy analysis would conclude that the files had been changed at 8pm in the evening, not at midday.

# Behaviour of Time Records

- We need to link an action to a time:
  - Which means that we need to know what actions result in updating which metadata fields.
  - The subject is complex – you may need to account for how specific applications or network configurations behave.
    - Often you will need to model and check critical behaviour.
- The following approximate rules apply to the file system (not, for example, to Internet Explorer)

# CMA times in Windows MFT

- **Created** time is set when a file is:
  - Newly created (Including 'save as' operations from applications that create new files).
  - Copied within the file system.
  - Imported from an external source (Internet, Network, Portable storage).
- **Modified** time is set when a file is:
  - Content updated in the same location.
- **Accessed** time is not normally used (in desktop PCs, from Windows 7, usually set to the same as created time).

*copying  
created  
modified  
accessed  
or changed  
or written  
or saved*

# Time caveats

- Copy may update the created time without the modified time:
  - Not uncommon to find a modification date before the created date.
- Move, Cut-and-Paste, Drag-and-Drop are different to ‘copy’ and may not change these dates.
- MFT Record date provides extra evidence of when file metadata was changed.
- Be sure you know what the filesystem supports AND HOW YOUR TOOLS PRESENT IT. (CD-ROMs are interesting)

# Deleted files

- If the MFT record has not been overwritten:
  - The full filename and file metadata are still available.
  - The location extents provide file location.
- Be careful:
  - That the file location contains the expected file, not overwritten.
    - Check overlaps with other files.
    - File is intact and has correct start and end.
    - File decodes...
  - That the generation number of the folder to which you ascribe the file is equal, or 1 higher than that given for the parent of the file.
    - Otherwise the folder MFT record may have been reused – you may assign the file to the wrong folder.

# Alternate Data Streams

- NTFS can support several ‘streams’ – ie different data items associated with the same file name.
  - Streams other than the main file are referred to as ‘alternate data streams’.
  - Although we didn’t do this in the practical, you may have noticed that MFT attributes are named – usually the first stream in a file is not named, the others have a name indicating their function.
  - One use of secondary streams is to record ‘zone’ information.
  - It is possible to refer to an alternate data stream with a special filename, e.g: Body.png:Zone.Identifier .
- Malware components are sometimes hidden in the alternate data streams of otherwise innocent files.

# Zones

- Zones are used for protection – e.g. to indicate that files are from an untrusted source..

| Zone | Source    |
|------|-----------|
| 1    | Intranet  |
| 2    | Trusted   |
| 3    | Internet  |
| 4    | Untrusted |

- Files imported from untrusted or Internet sources are usually labelled with a ‘Zone.Identifier’
- In investigations the zone may provide information about the source of the file.

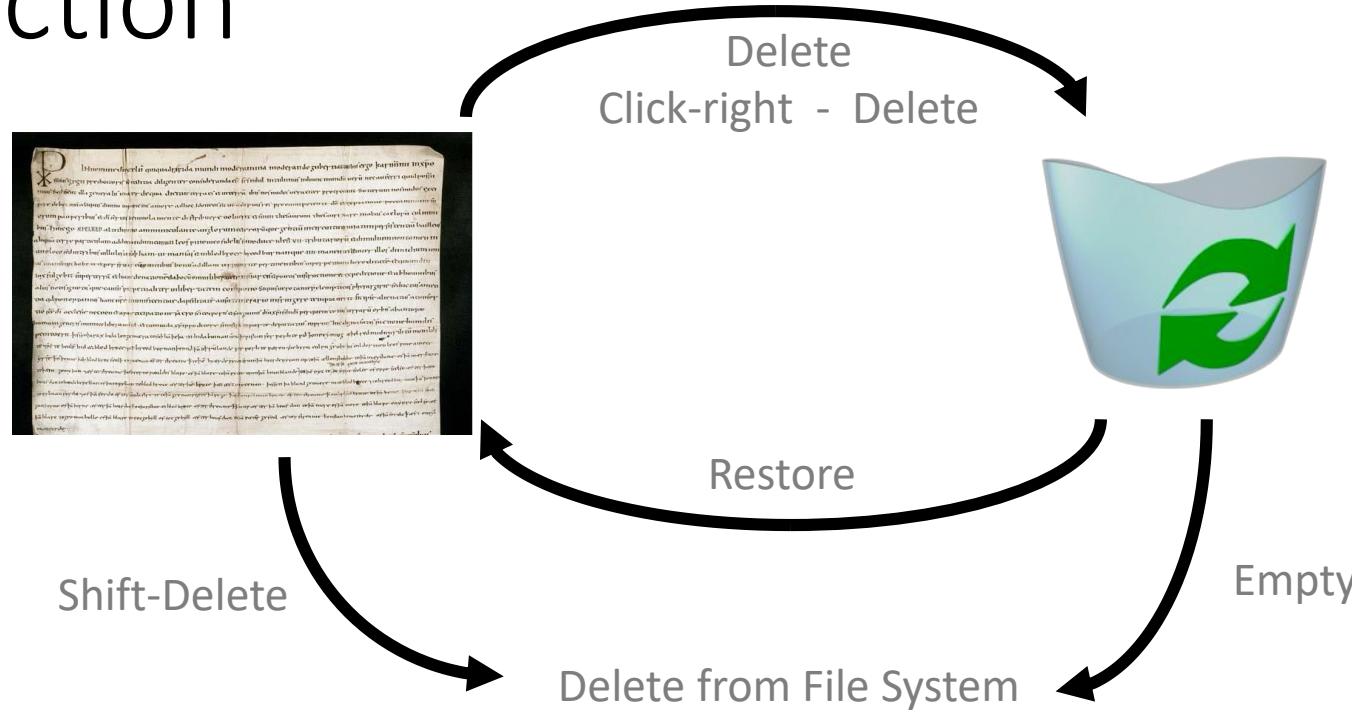
# File Recovery

What if: the actual file has not been overwritten, but the \$MFT record has been reused?

The file is in ‘file system unallocated’ space.

# The ‘recycle bin’

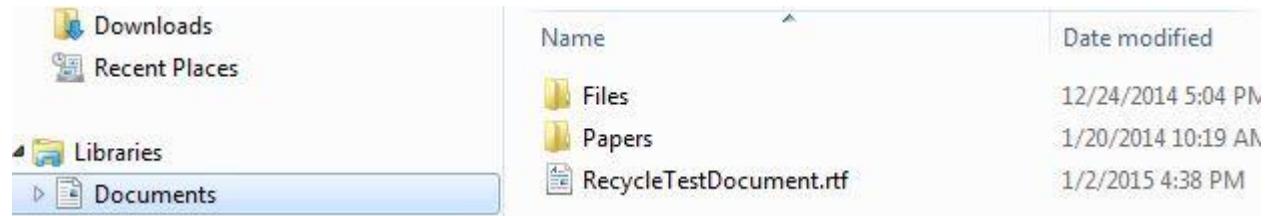
# Introduction



In Windows, applies only to operations from Windows Explorer.  
Files that are 'deleted' are simply moved to another folder (recycle bin).  
Of course, not quite that simple ...

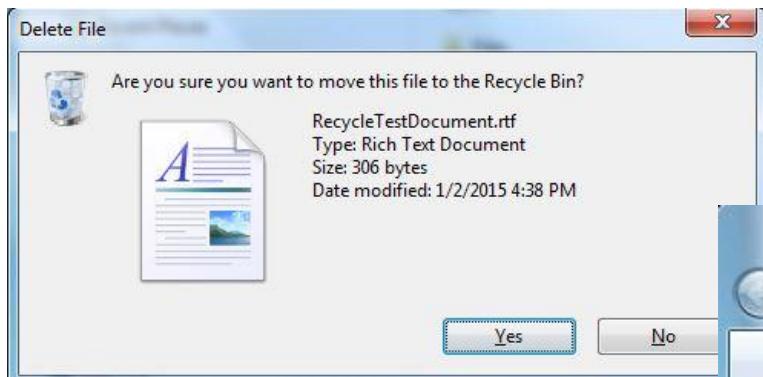
# Example

Original Test Document



| Name                    | Date modified      |
|-------------------------|--------------------|
| Files                   | 12/24/2014 5:04 PM |
| Papers                  | 1/20/2014 10:19 AM |
| RecycleTestDocument.rtf | 1/2/2015 4:38 PM   |

'Delete'

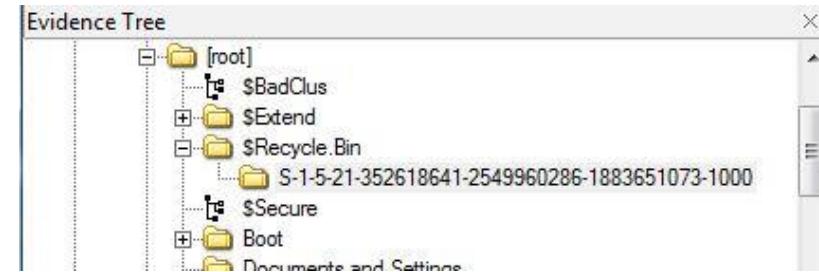


User view of Recycle Bin



# Forensic View of Recycle Bin

Documents held under a user SID in \$Recycle.Bin\



..... pairs of files, renamed, same extension .....

\$I... metadata, including original path

| Name         | Size | Type         | Date Modified      |
|--------------|------|--------------|--------------------|
| SIOYWFXJ.rtf | 1 KB | Regular File | 1/2/2015 4:42:0... |
| SROYWFXJ.rtf | 1 KB | Regular File | 1/2/2015 4:38:5... |
| desktop.ini  | 1 KB | Regular File | 10/27/2013 10:...  |

|     |   |                  |
|-----|---|------------------|
| 000 | 01 00 00 00 00 00 00 00-32 01 00 00 00 00 00 00 | .....2.....      |
| 010 | 00 12 D9 0C AB 26 D0 01-43 00 3A 00 5C 00 55 00 | ..Ù-«æB-C::-\U-  |
| 020 | 73 00 65 00 72 00 73 00-5C 00 73 00 74 00 75 00 | s-e-r-s-\s-t-u-  |
| 030 | 64 00 65 00 6E 00 74 00-5C 00 44 00 6F 00 63 00 | d-e-n-t-\D-o-c-  |
| 040 | 75 00 6D 00 65 00 6E 00-74 00 73 00 5C 00 52 00 | u-m-e-n-t-s-\R-  |
| 050 | 65 00 63 00 79 00 63 00-6C 00 65 00 54 00 65 00 | e-c-y-c-l-e-T-e- |
| 060 | 73 00 74 00 44 00 6F 00-63 00 75 00 6D 00 65 00 | s-t-D-o-c-u-m-e- |
| 070 | 6E 00 74 00 2E 00 72 00-74 00 66 00 00 00 00 00 | n-t..-r-t-f----- |
| 080 | 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 00 | .....            |

\$R... the actual file

| Name         | Size | Type         | Date Modified      |
|--------------|------|--------------|--------------------|
| SIOYWFXJ.rtf | 1 KB | Regular File | 1/2/2015 4:42:0... |
| SROYWFXJ.rtf | 1 KB | Regular File | 1/2/2015 4:38:5... |
| desktop.ini  | 1 KB | Regular File | 10/27/2013 10:...  |

|     |   |                   |
|-----|---|-------------------|
| 000 | 7B 5C 72 74 66 31 5C 61-6E 73 69 5C 61 6E 73 69 | {\rtf1\ansi\ansi  |
| 010 | 63 70 67 31 32 35 32 5C-64 65 66 66 30 5C 64 65 | cpg1252\deff0\de  |
| 020 | 66 6C 61 6E 67 31 30 33-33 7B 5C 66 6F 6E 74 74 | flang1033{\fontt  |
| 030 | 62 6C 7B 5C 66 30 5C 66-6E 69 6C 5C 66 63 68 61 | b1{\f0\fni1\fcha  |
| 040 | 72 73 65 74 30 20 43 61-6C 69 62 72 69 3B 7D 7D | rset0 Calibri;}}  |
| 050 | 0D 0A 7B 5C 2A 5C 67 65-6E 65 72 61 74 6F 72 20 | ..\{*generator    |
| 060 | 4D 73 66 74 65 64 69 74-20 35 2E 34 31 2E 32 31 | Msftedit 5.41.21  |
| 070 | 2E 32 35 31 30 3B 7D 5C-76 69 65 77 6B 69 6E 64 | .2510;}\viewkind  |
| 080 | 34 5C 75 63 31 5C 70 61-72 64 5C 73 61 32 30 30 | 4\uc1\pard\sa200  |
| 090 | 5C 73 6C 32 37 36 5C 73-6C 6D 75 6C 74 31 5C 6C | \s1276\slmult1\1  |
| 0a0 | 61 6E 67 39 5C 66 30 5C-66 73 32 32 20 54 68 65 | ang9\f0\fs22 The  |
| 0b0 | 20 71 75 69 63 6B 20 62-72 6F 77 6E 20 66 6F 78 | quick brown fox   |
| 0c0 | 20 6A 75 70 73 20 6F 76-65 72 20 74 68 65 20 6C | jups over the l   |
| 0d0 | 61 7A 79 20 64 6F 67 2E-5C 70 61 72 0D 0A 54 68 | azy dog.\par..Th  |
| 0e0 | 69 73 20 69 73 20 61 20-74 65 73 74 20 66 69 6C | is is a test fil  |
| 0f0 | 65 20 74 6F 20 64 65 6D-6F 6E 73 74 72 61 20 72 | e to demonstra r  |
| 100 | 65 63 79 63 6C 65 20 74-68 65 20 72 65 63 79 63 | ecycle the recyc  |
| 110 | 6C 65 20 62 69 6E 20 69-6E 20 77 69 6E 64 6F 77 | le bin in window  |
| 120 | 73 2E 5C 70 61 72 0D 0A-5C 70 61 72 0D 0A 7D 0D | s.\par..\\par..}. |
| 130 | 0A 00   | ..                |

# ‘Delete’ – other comments

- In Windows only NTFS file systems have a recycler.
- If you see a recycle bin on a FAT file system, or on a small USB it is possibly malware camouflage.
- Recycle bins may be found on portable drives formatted with NTFS (unpredictably), if so they may collect SIDs from several computers.
- Dates and times usually stay the same for the \$R file (assuming accessed times are not specifically enabled).
- The Date and Time for the \$I file is the time that the file was placed in the recycle bin.
- These are very easy to review and recover in any editor that allows you to view raw data.



# OTHER EVIDENCE OF FILE USE and METADATA

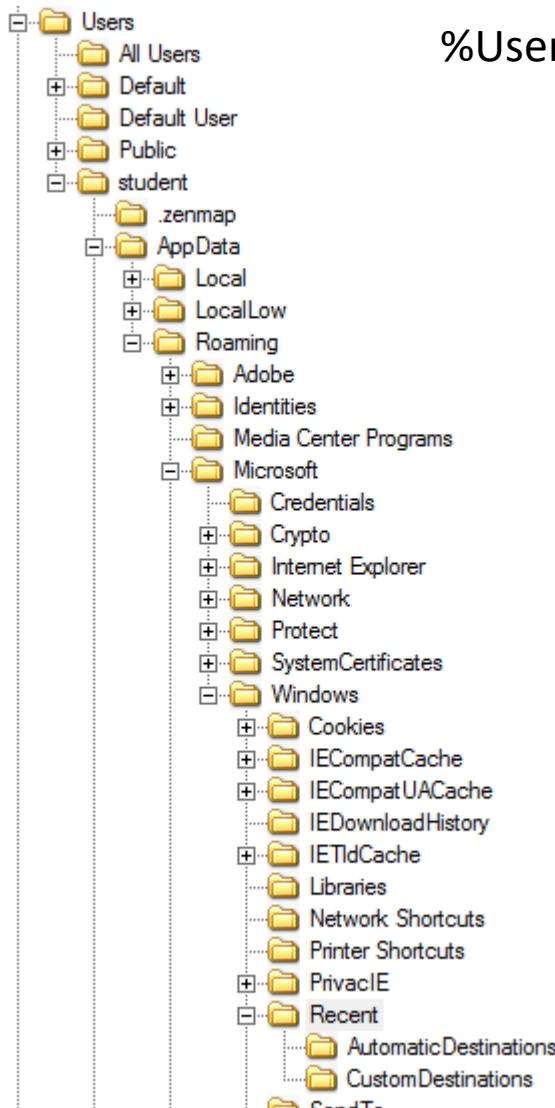
# Use of Files

Depending on how a file is opened or executed, the action may be recorded in a number of places:

- The Registry.
- The Internet Explorer History.
- In Operating System prefetch files.
  - Used to speedup application startup.
  - /Windows/Prefetch
- As shortcuts (lnk files) in users Recent Files Folder.
- In Jump List files, similar location.

The registry and Internet Explorer will be covered later, here we will briefly describe lnk files.

# Recent Files ('lnk' or 'shortcut')



%User%\AppData\Roaming\Microsoft\Windows\Recent

Ink files carry the filename with the additional extension '.lnk'.

|     | ReadMe.txt.lnk.FileSlack                        | 4 KB | File Slack       |                     |
|-----|---|------|------------------|---------------------|
|     | RecycleTestDocument.rtf.lnk                     | 2 KB | Regular File     | 02/01/2015 16:38:52 |
|     | RecycleTestDocument.rtf.lnk.File...             | 3 KB | File Slack       |                     |
| 400 | 00 00 00 1F 00 00 00 05-00 00 00 2E 00 72 00 74 |      | .....r-t         |                     |
| 410 | 00 66 00 00 00 00 00 00-00 00 00 BE 00 00 00 31 |      | -f.....%..1      |                     |
| 420 | 53 50 53 40 E8 3E 1E 2B-BC 6C 47 82 37 2A CD 1A |      | SPS@>+>1G-7*f-   |                     |
| 430 | 83 9B 22 79 00 00 00 08-00 00 00 00 1F 00 00 00 |      | ..y.....         |                     |
| 440 | 33 00 00 43 00 3A 00-5C 00 55 00 73 00 65 00    |      | 3...C.:.\-U-s-e- |                     |
| 450 | 72 00 73 00 5C 00 73 00-74 00 75 00 64 00 65 00 |      | r-s.\-s-t-u-d-e- |                     |
| 460 | 6E 00 74 00 5C 00 44 00-6F 00 63 00 75 00 6D 00 |      | n-t.\-D-o-c-u-m- |                     |
| 470 | 65 00 6E 00 74 00 73 00-5C 00 52 00 65 00 63 00 |      | e-n-t-s.\-R-e-c- |                     |
| 480 | 79 00 63 00 6C 00 65 00-54 00 65 00 73 00 74 00 |      | y-c-l-e-T-e-s-t- |                     |
| 490 | 44 00 6F 00 63 00 75 00-6D 00 65 00 6E 00 74 00 |      | D-o-c-u-m-e-n-t- |                     |
| 4a0 | 2E 00 72 00 74 00 66 00-00 00 00 00 29 00 00 00 |      | .r-t-f.....)     |                     |
| 4b0 | 02 00 00 00 00 1F 10 00 00 01 00 00 00 00 00    |      |                  |                     |

# Ink (recent files) Summary

- Ink files contain a wealth of metadata, including the Created/Modified/Accessed times of the target file.

| Linked path  | Created             | Written             | Last Accessed       | Size [B] |
|--|---------------------|---------------------|---------------------|----------|
| C:\Users\student\Documents\RecycleTestDocument.rtf | 02/01/2015 16:38:52 | 02/01/2015 16:38:52 | 02/01/2015 16:38:52 | 306      |

- The time of the link itself is the time the file was opened (be careful, opening via some applications may not set a shortcut.)
- An important forensic feature is that the Recent folder will also carry shortcuts for files held on removable media.

The tool used here is Mitec Windows File Analyser

# Additional Material

- Kessler, G. File Signatures Table, [http://www.garykessler.net/library/file\\_sigs.html](http://www.garykessler.net/library/file_sigs.html)  
(see also /etc/magic on \*nix systems)
- Parsonage, H. (2010) The meaning of LIFE (Link Files in Forensic Examinations).  
<http://computerforensics.parsonage.co.uk/downloads/TheMeaningofLIFE.pdf>

# The Windows Registry.

# Why ?

Almost every aspect of configuration in a Windows operating system is held in a structure known as ‘The Registry’.

A rich source of evidence:

System configuration (‘Control Panel’), software, time settings.

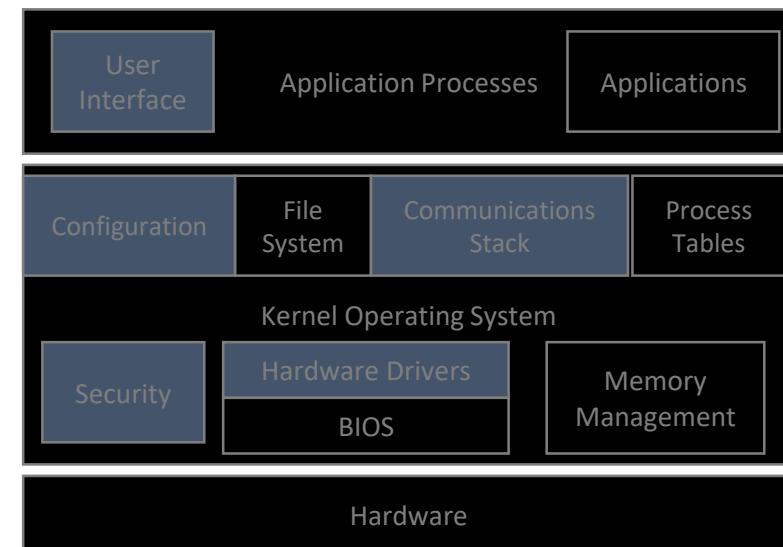
Devices, e.g. connected USB devices.

User accounts.

Communications Settings (IP addresses etc).

User actions.

cf Linux/Unix/Mac OS use of ‘etc’ ‘var’ ‘lib’ directories and Mac .plist files



# Contents

Registry Structure.

Online/Offline differences.

Registry Files and Content.

Identities.

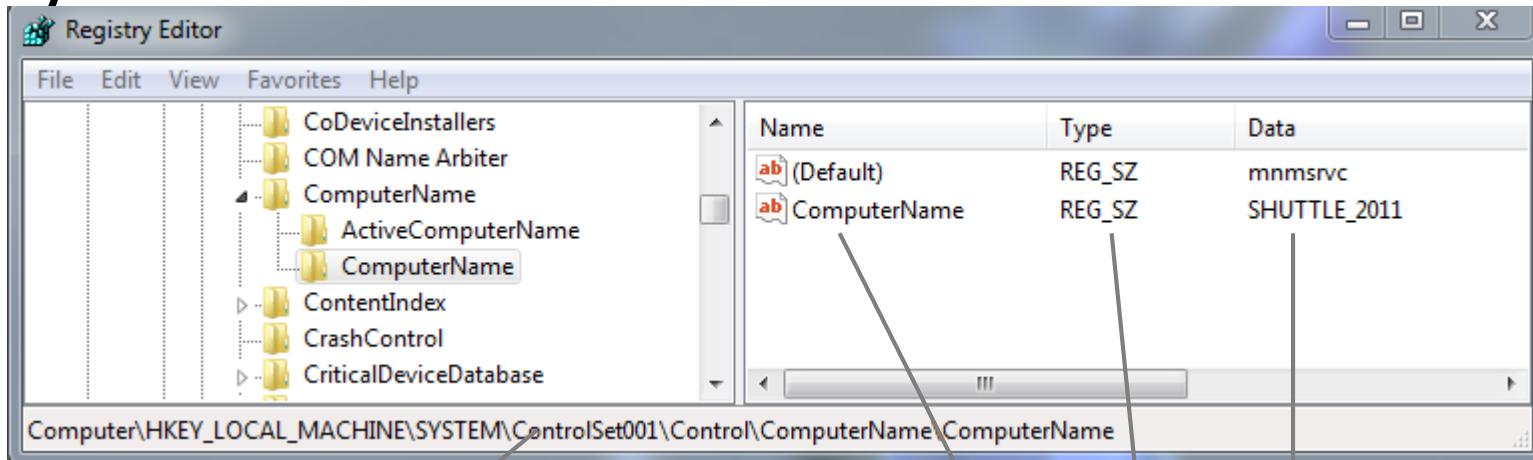
Software, devices, services (GUIDs) .

Users (SIDs).

Example: USB Forensics.

This lecture introduces the registry, its files and data types. Obtaining information from the registry will then be studied in the practicals.

# Registry Structure



Registry is organised as a 'folder' structure.

The folder path is called a **key**

**value - type - data**  
are listed under keys.

This tool is 'regedit'. It is available on all Windows systems and views/edits the **live** registry.

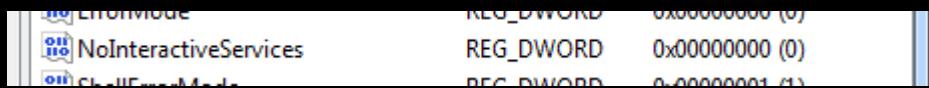
Warning: regedit will allow you to modify (and break) the configuration of your computer.

In CAINE-YE we have "FRED", to edit Windows registries, which can be just as dangerous. (look under Analysis)

# Common Value Types

| Data Type     | Description  |
|---------------|--|
| REG_BINARY    | Raw binary data. The format is specific to the application, often a device driver.     |
| REG_DWORD     | 32 bit integer. Note that True (1) and False (0) are also recorded as dwords.          |
| REG_EXPAND_SZ | A string which contains a variable to be replaced when it is read. e.g: %user%\AppData |
| REG_MULTI_SZ  | Lists in human readable text.  |
| REG_SZ        | A readable text string.  |

Warning – be careful when interpreting boolean registry keys, since the value may be worded as a negative.



e.g. this means that Interactive services are enabled.

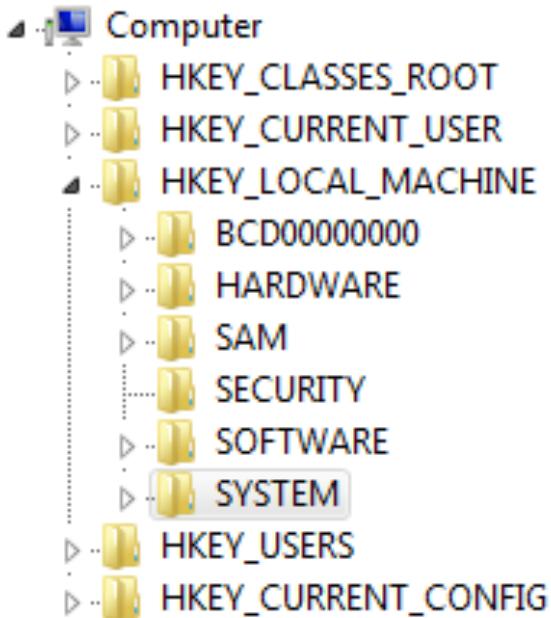
# Dates and Times

Registry keys have an associated ‘Last Write’ time.

- FILETIME format, the same as NTFS times.
- Note that if the key has many values, this does not tell you which value was written at that time.
- Some system processes (e.g. virus checking, enumeration) may set many keys.
- Time is recorded in UTC (GMT) time, with the same issues as the NTFS times.

Do not assume that all computer dates and times are UTC. Times stored as binary data are application dependent.

# Hives



Root keys are compiled from registry files known as ‘hives’.

There is not a 1:1 correspondence between live root keys and hives.

The live registry may also contain keys which are completely volatile (not stored on disk).

volatile keys include:

HKEY\_CURRENT\_USER      compiled from the specific user’s hive

HKEY\_CLASSES\_ROOT      compiled from Software and User hives

HKEY\_CURRENT\_CONFIG      compiled from the System hive

# Comment

- Usually we are concerned with file-based registry hives.
- The live registry may be of value when investigating malware:
  - Remember hiding a user account in MALF by changing a registry key?
  - Malware may set keys to disable security functions, or ensure that the malware is loaded on startup – some of these keys may be volatile.

# System Hives

**%System Root%\System32\Config**

e.g. C:\Windows\System32\Config

|          |   |
|----------|---|
| SAM      | Security Account Manager database of accounts, including user name.   |
| SECURITY | Local security profile, such as user rights.  |
| SYSTEM   | System start-up, operating system behaviour and device information.   |
| SOFTWARE | Database of installed software and its configuration, including registered extensions that populate the HKEY_CLASSES_ROOT live key. |

Note – there are many other files in the Windows system with a registry hive data structure.

# User Hives

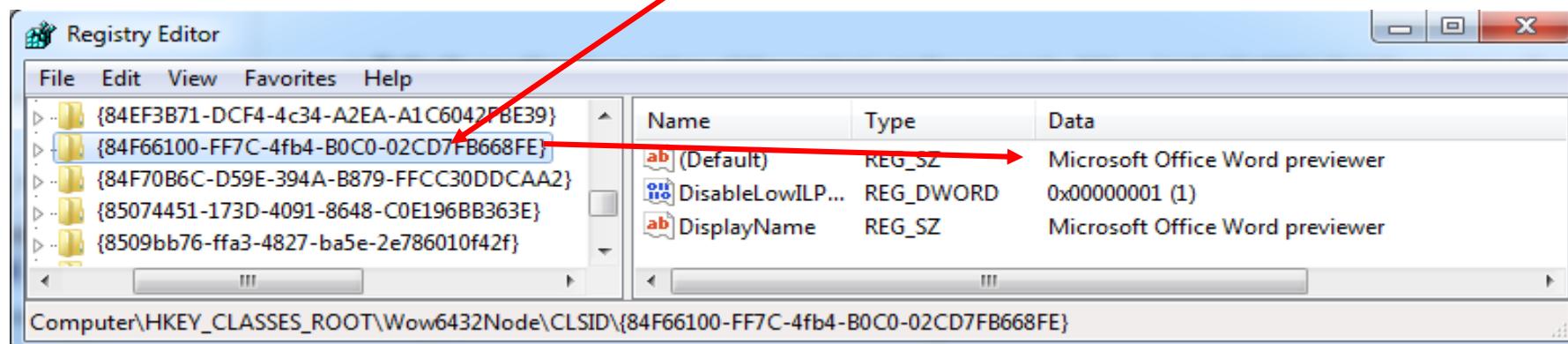
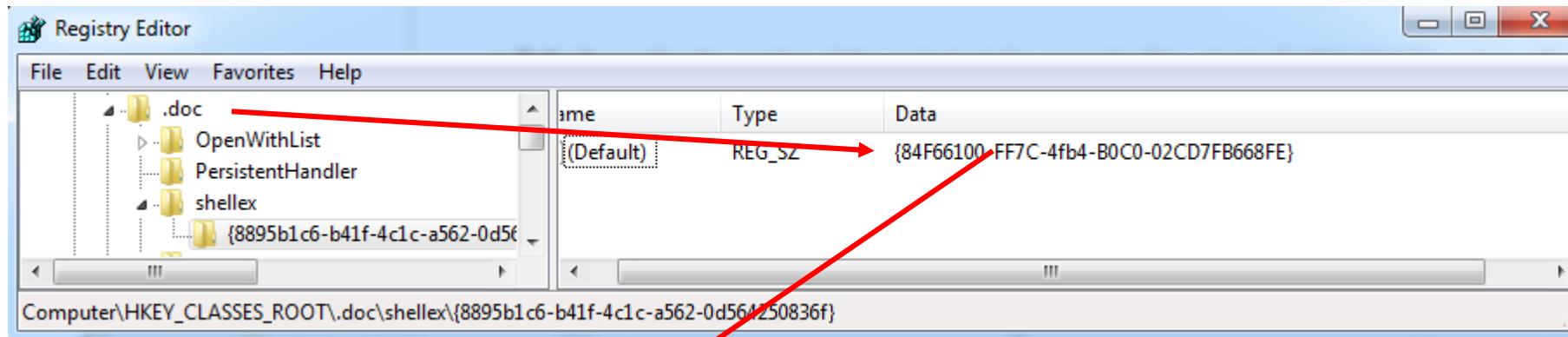
- %user%NTUSER.DAT
  - e.g. C:\users\howard\NTUSER.DAT
  - Primary settings for that specific user, will override any configuration settings in other hives in the live system.
  - This is the most important user configuration file.
- Two other keys sometimes contain useful data:
  - %user%AppData\Local\Microsoft\Windows\userclass.dat
    - Includes settings for such things as display windows and tray notifications.
  - %defaultuser%NTUSER.DAT
    - Default user settings, loaded before specific user settings into the live registry.

# GUID

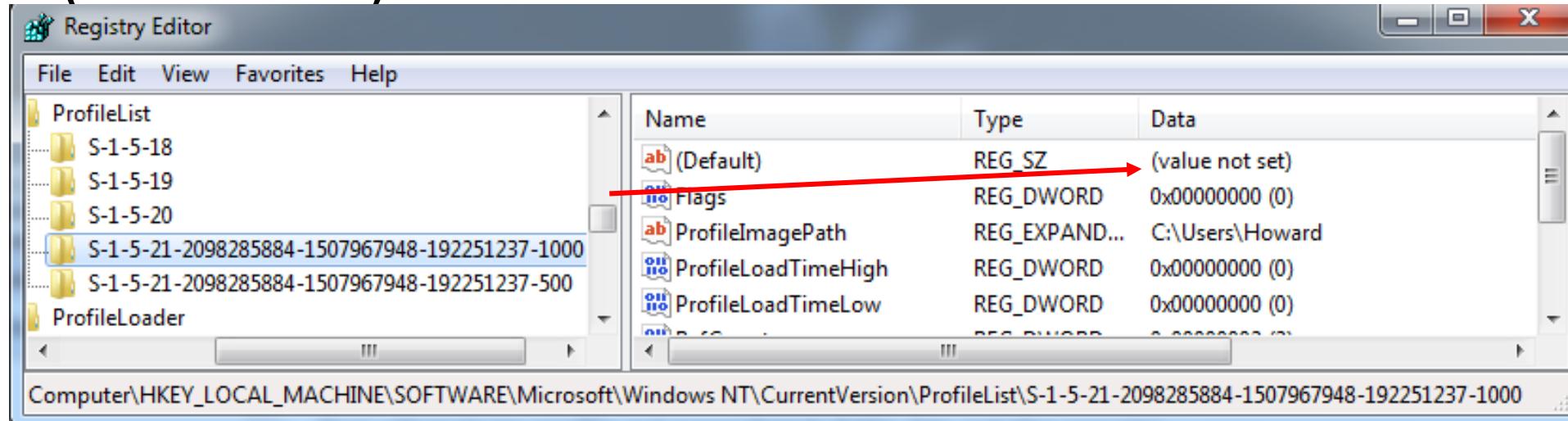
↓  
Devices  
such as  
USB's

SID - users

- GUID are references {.....} used to identify software, devices and services.
- They are used as cross-references within the registry.



# SID (User ID)



S-1-5-21-2098285884-1507967948-192251237-1000

Domain Identifier

Standard prefix, means 'user'.

See [<http://support.microsoft.com/KB/243330>]  
(Short SIDs are service accounts.)

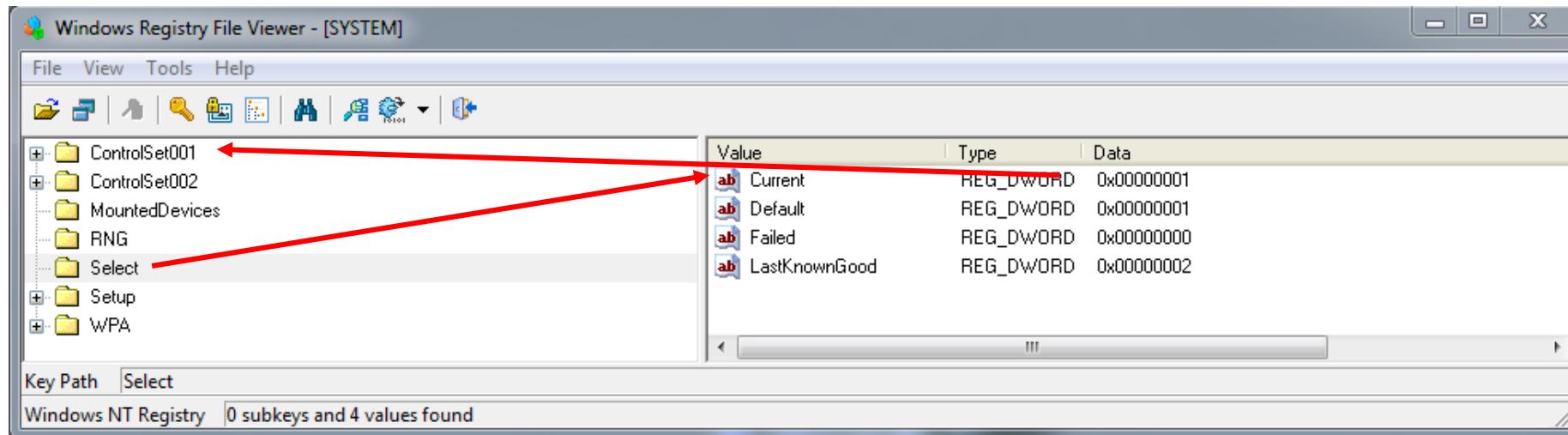
User Relative ID (RID)

Normal users are numbered from  
1000,  
500 is the system administrator.

Using the registry to identify connected flash drives.

## Example – USB Forensics

# SYSTEM registry



The SYSTEM\select key tells us to use ControlSet001

This tool is the Mitec Registry Viewer which allows us to look at the hives (files) rather than the live registry.

[<http://www.mitec.cz/>]

# USB enumerate

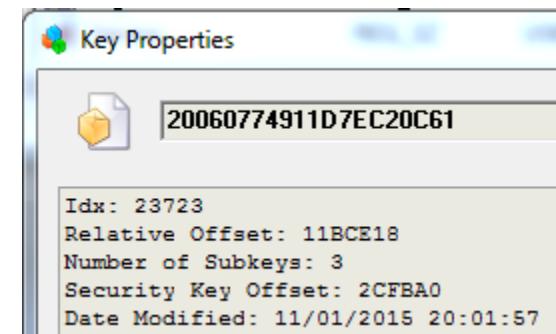
The screenshot shows the Windows Registry File Viewer window titled "Windows Registry File Viewer - [SYSTEM]". The left pane displays a tree view of registry keys under "ControlSet001\Enum\USB\VID\_0781&PID\_556B". One key, "20060774911D7EC20C61", is selected and highlighted with a blue selection bar. The right pane shows a detailed table of registry values for this key:

| Value               | Type         | Data   |
|---------------------|--------------|--|
| DeviceDesc          | REG_SZ       | @usbstor.inf,%genericbulkonly.devicedesc%:USB Mass Storage Device    |
| LocationInformation | REG_SZ       | Port_#0002.Hub_#0004   |
| Capabilities        | REG_DWORD    | 0x000000D4   |
| HardwareID          | REG_MULTI_SZ | USBVID_0781&PID_556B&REV_0126USBVID_0781&PID_556B                    |
| CompatibleIDs       | REG_MULTI_SZ | USB\Class_08&SubClass_06&Prot_50USB\Class_08&SubClass_06USB\Class_08 |
| ContainerID         | REG_SZ       | {500ef7ff-5c58-59ea-b192-9656ef58f950}                               |
| ConfigFlags         | REG_DWORD    | 0x00000000   |
| ClassGUID           | REG_SZ       | {36fc9e60-c465-11cf-8056-444553540000}                               |
| Driver              | REG_SZ       | {36fc9e60-c465-11cf-8056-444553540000}\0076                          |
| Class               | REG_SZ       | USB  |
| Mfg                 | REG_SZ       | @usbstor.inf,%generic.mfg%:Compatible USB storage device             |
| Service             | REG_SZ       | USBSTOR  |

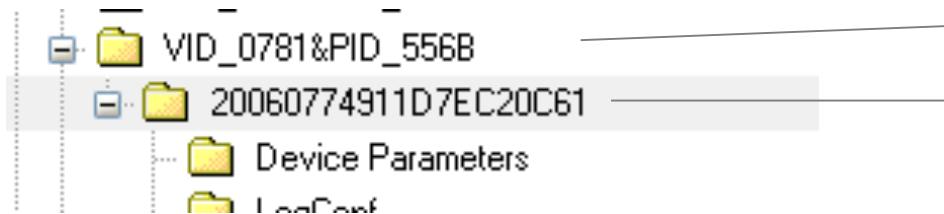
Key Path: ControlSet001\Enum\USB\VID\_0781&PID\_556B\20060774911D7EC20C61

When a USB is first inserted it is ‘registered’ – a permanent registry record is made.

The record time may be updated when the enumerator is run **NOT** necessarily when the device is plugged in again!



# VID & PID



```
0781 SanDisk Corp.  
    0001 SDDR-05a ImageMate CompactFlash Read  
    0002 SDDR-31 ImageMate II CompactFlash Re  
    0005 SDDR-05b (CF II) ImageMate CompactFl  
    0100 ImageMate SDDR-12  
    0200 SDDR-09 (SSFDC) ImageMate SmartMedia  
    0400 SecureMate SD/MMC Reader  
    0621 SDDR-86 Imagemate 6-in-1 Reader  
    0720 Sansa C200 series in recovery mode  
    0729 Sansa E200 series in recovery mode  
    0810 SDDR-75 ImageMate CF-SM Reader  
    0830 ImageMate CF/MMC/SD Reader  
    1234 Cruzer Mini Flash Drive  
    5150 SDCZ2 Cruzer Mini Flash Drive (thin)  
    5151 Cruzer Micro Flash Drive  
    5153 Cruzer Flash Drive  
    5204 Cruzer Crossfire  
    5402 U3 Cruzer Micro  
    5406 Cruzer Micro U3  
    5408 Cruzer Titanium U3  
    540e Cruzer Contour Flash Drive  
    5530 Cruzer  
    5567 Cruzer Blade  
    556c Ultra  
    556d Memory Vault
```

Vendor & Device Ids

Serial Number

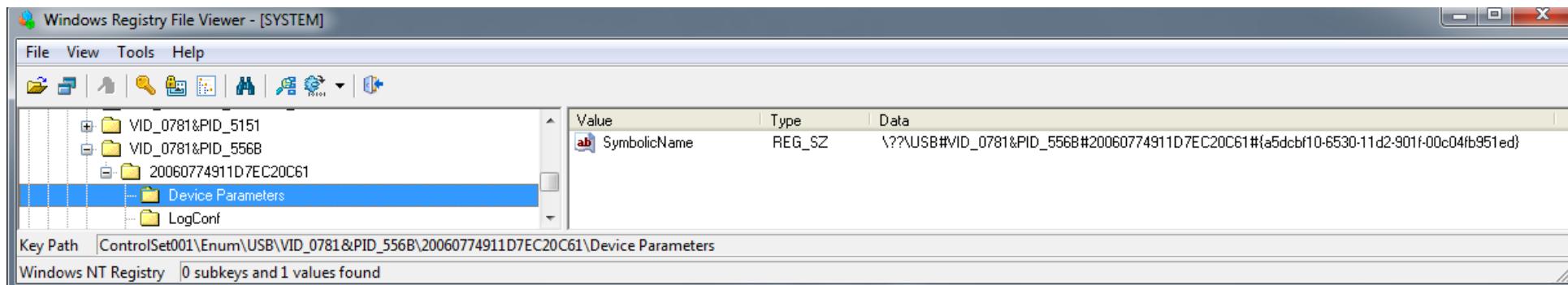
(If the 2<sup>nd</sup> character of the serial is ‘&’ it has been created by Windows, not by the device.)

We can often look up the vendor and device.

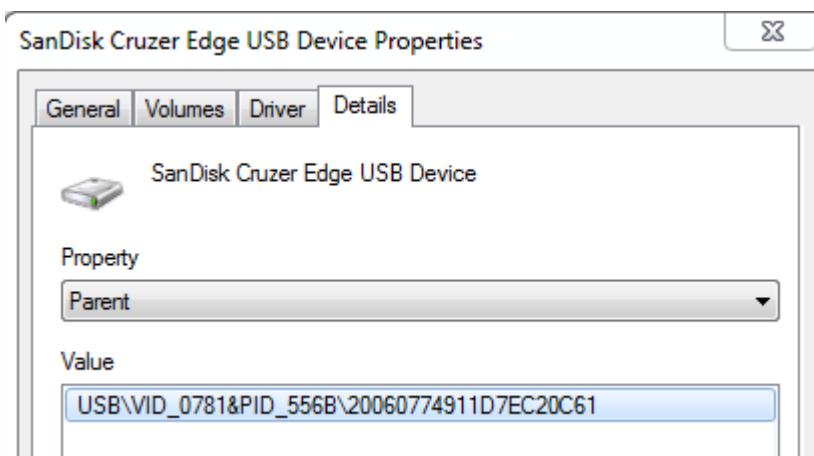
Here we can see the vendor, the device may be Cruzer, but is not listed.

[<http://www.linux-usb.org/usb.ids>]

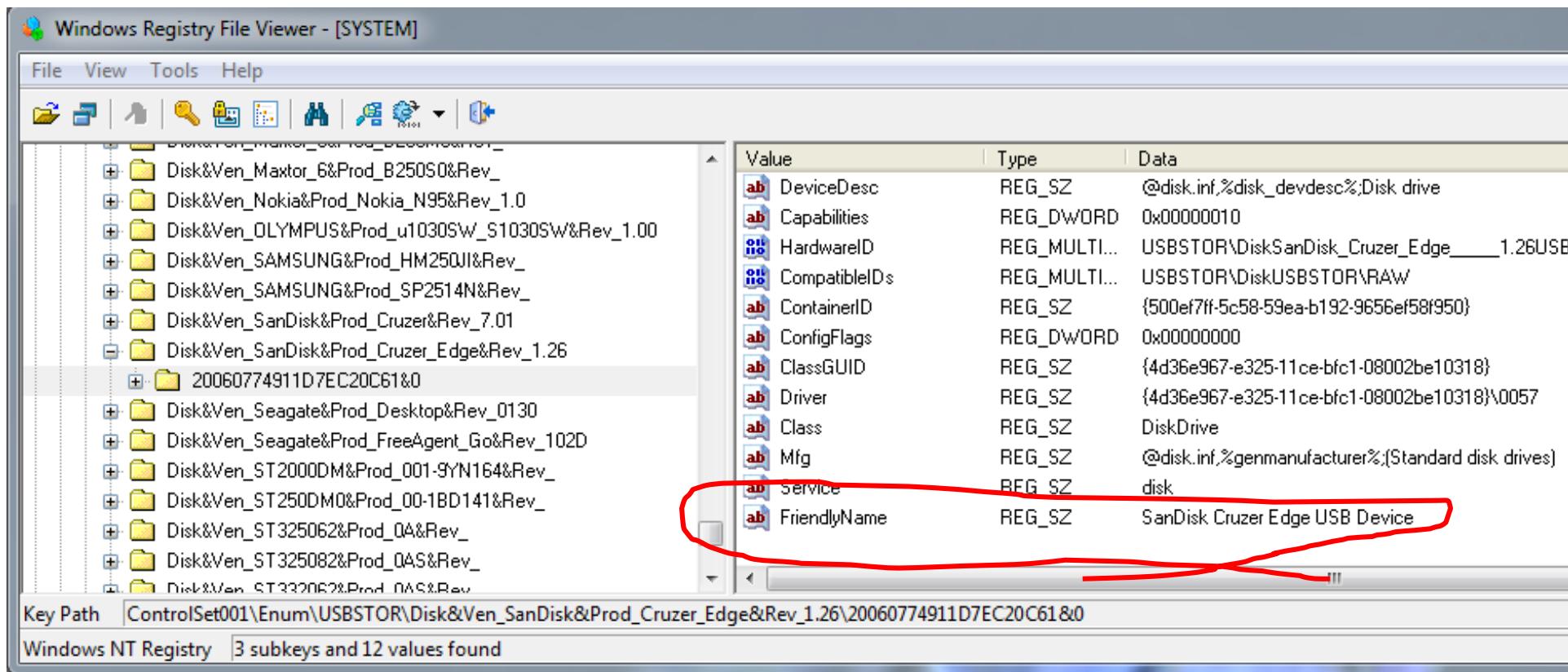
# Device Metadata



The symbolic name combines the VID and SID and the serial number. If the device is available this may be obtained via device properties.



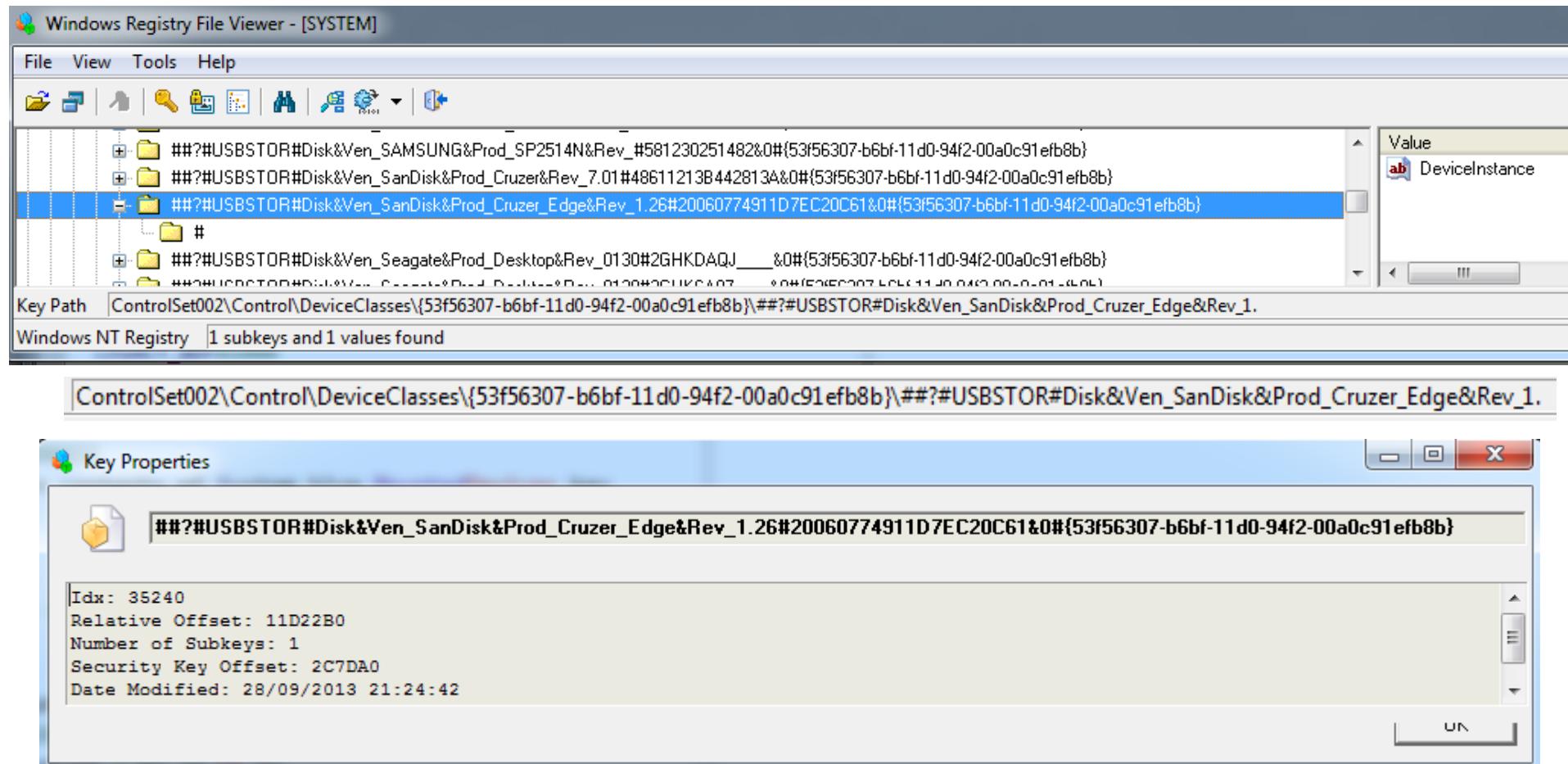
# Friendly Name



Searching for the serial number allows a second record to be found  
under Enum\USBSTOR.

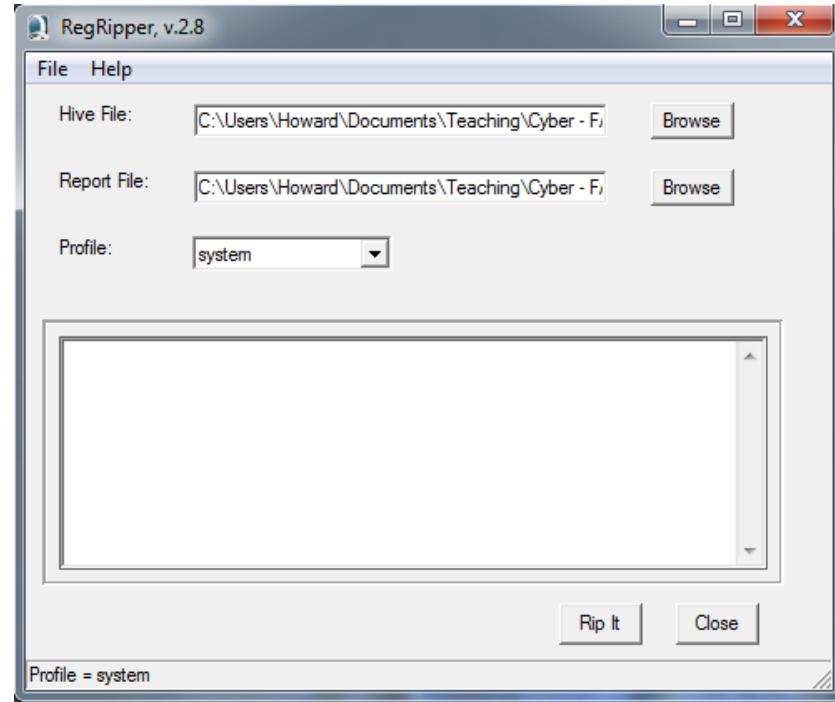
If the device is available this may be obtained via device properties.

# Devices



A further record is found under DeviceClasses: this is updated when the actual USB is inserted, so we have a time of use.

# Scripted Registry Extraction



There are several report generators that extract useful registry information. They are not a substitute for understanding how the applications work since their output depends on the installed plugins, which may not extract the specific evidence you need.

# Regripper Extract

## DevClasses - Disks

ControlSet001\Control\DeviceClasses\{53f56307-b6bf-11d0-94f2-00a0c91efb8b}

...

Sat Sep 28 21:24:42 2013 (UTC)

Disk&Ven\_SanDisk&Prod\_Cruzer\_Edge&Rev\_1.26,20060774911D7EC20C  
61&0

## usbdevices v.20120522

(System) Parses Enum\USB key for devices

VID\_0781&PID\_556B

LastWrite: Sun Jan 11 20:01:57 2015

SN : 20060774911D7EC20C61

LastWrite: Sun Jan 11 20:01:57 2015

# Further Information

- There are other important sources of information (there always are!) beyond the scope of this module:
  - Windows Event Logs: which may record USBs, when USBs are mounted.
  - `setupapi.dev.log`: which records the first time a device is installed.

# Conclusions

- The Windows Registry is a critical resource for forensics: not just configuration but also machine and user history.
- Often it is necessary to establish serial numbers, friendly names, GUIDs, SIDs, then find associated records in other parts of the registry.
- Be very cautious with date and time evidence:
  - Dates within binary keys will be application specific.
  - The granularity of a date is to a key, not a value.
  - Some events (anti-virus, device enumeration) may update huge numbers of sub-keys.

(PS: Many books will tell you that registry USB enumeration times give the last time the USB was plugged in. This was so on some systems, it seems that it is not always so now!)

# Additional Information

- Known Folder Ids – standard GUID references. <http://msdn.microsoft.com/en-us/library/windows/desktop/dd378457%28v=vs.85%29.aspx>
- IronGeek, Forensically interesting spots in the Windows 7, Vista and XP file system and registry, <http://www.irongeek.com/i.php?page=security/windows-forensics-registry-and-file-system-spots&mode=print>



# This week's lab

- You're going to examine some Windows areas to try to determine what a user has been doing...after you've spent some time working on data representation and interpretation.

# Forensic Analysis

Part 4

# Event Logs

# Why?

Windows has a rich logging structure which includes records of system events – some such as login/out and time updates are often important..

# Content

- Event log overview.
- Interpreting Logon and other events in networks.

# Windows Event Logs

- A framework for recording and auditing ‘events’:
  - Logon/logoff
  - Time synchronization
  - Application actions ....
- A potentially rich source of forensic evidence, but:
  - Default retention time is only 7 days for networked system.
  - In Windows 7 local accounts, events are storage rather than time limited.
- Logs may be centralised.
  - More common with UNIX systems (SYSLOG) than Windows.
- Logs are similarly available for servers.

# Location

/Windows/System32/winevt/Logs/

**Application.evtx**

**Security.evtx** includes logon/off events

**System.evtx** includes time synchronisation

**Setup.evtx**

...

+ Many application specific event files

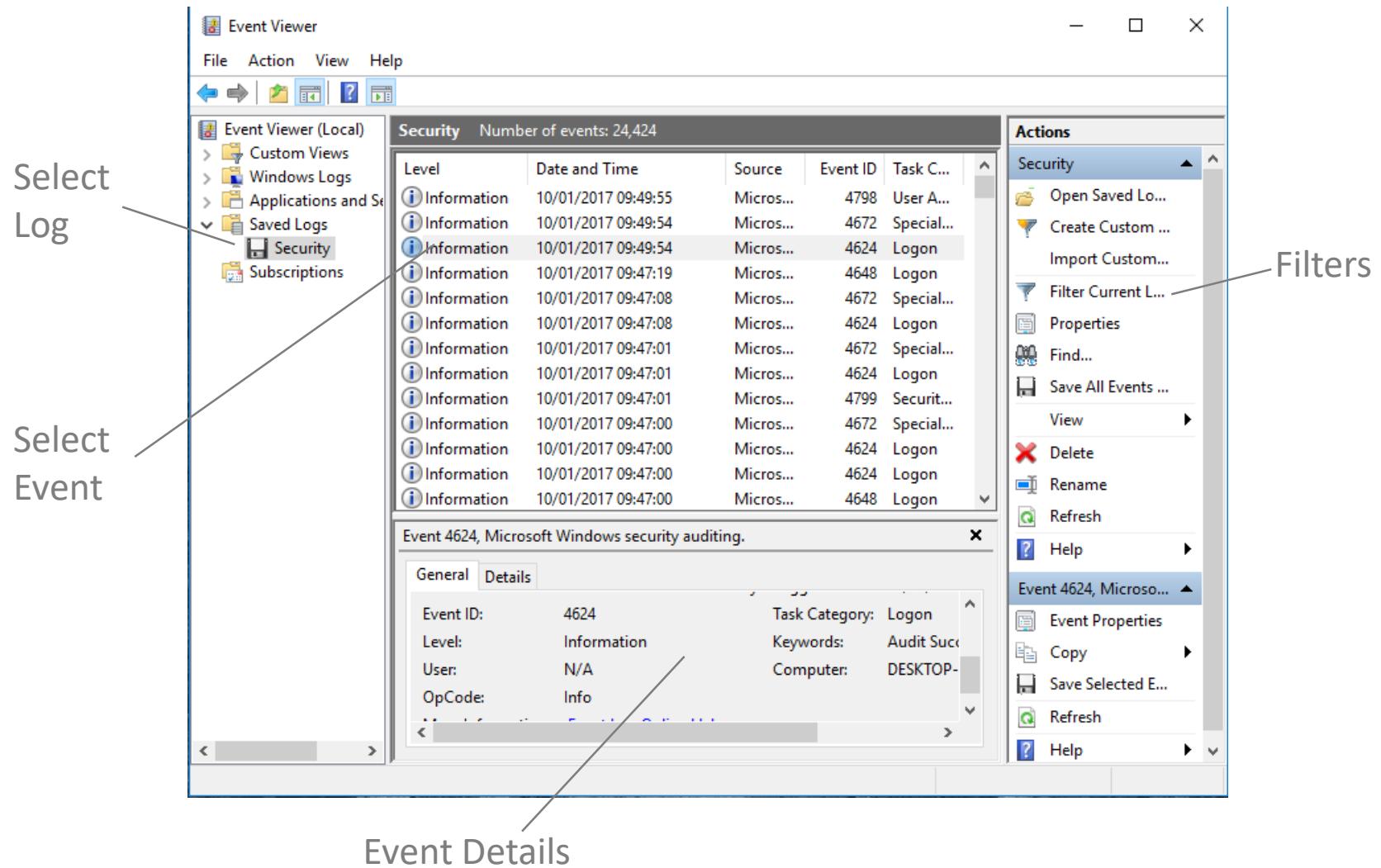
# Channels

- Individual logs (*security*, etc) are called ‘channels’.
  - Strictly, channels are members of ‘channel groups’: *Windows Logs or Application & Service Logs*.
- Channels may be of two types:
  - Serviced:
    - contain *Admin* or *Operational* events,
    - can be forwarded or collected remotely,
    - Include Security.evtx, System.evtx .
  - Direct
    - local logs, may not be centralised,
    - Include *Analytic* and *Debug* events.

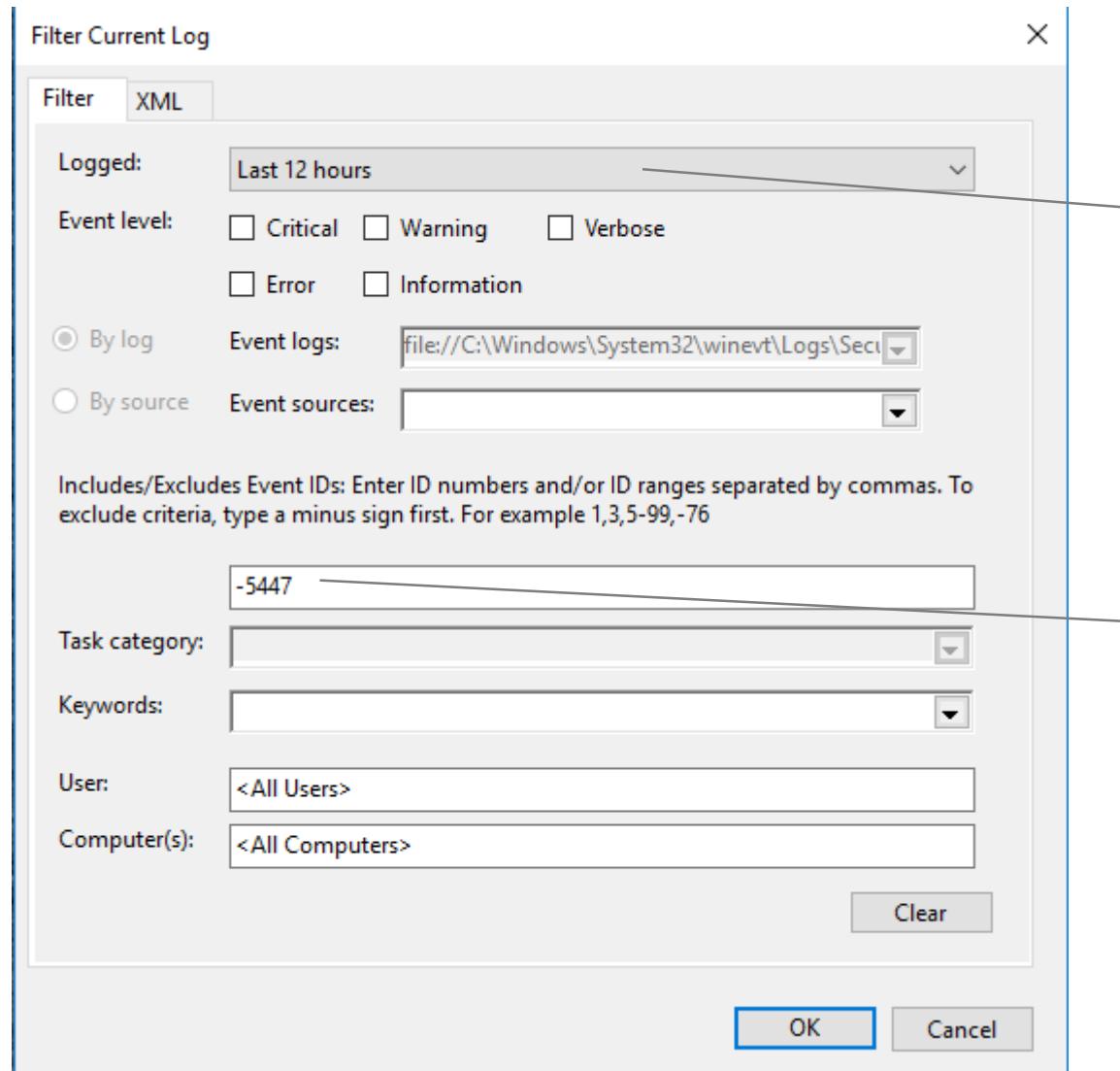
# Event Logs Include

- Source (Software, System Component).
- Event ID.
- Level (e.g. Information .... Critical ...).
- User.
  - Software often works for individuals, in which case it may use the ID of the individual, this is known as ‘impersonation’.
  - Events may include both ‘client’ and ‘impersonation’ Ids.
- Task information.
- Computer information (including datetimes).
- Ids.
  - Including a ‘correlation ID that links related events.

# Windows Event Viewer



# Filtering Events



Time Period

Note that '-'  
excludes events of  
this type

# Custom Views

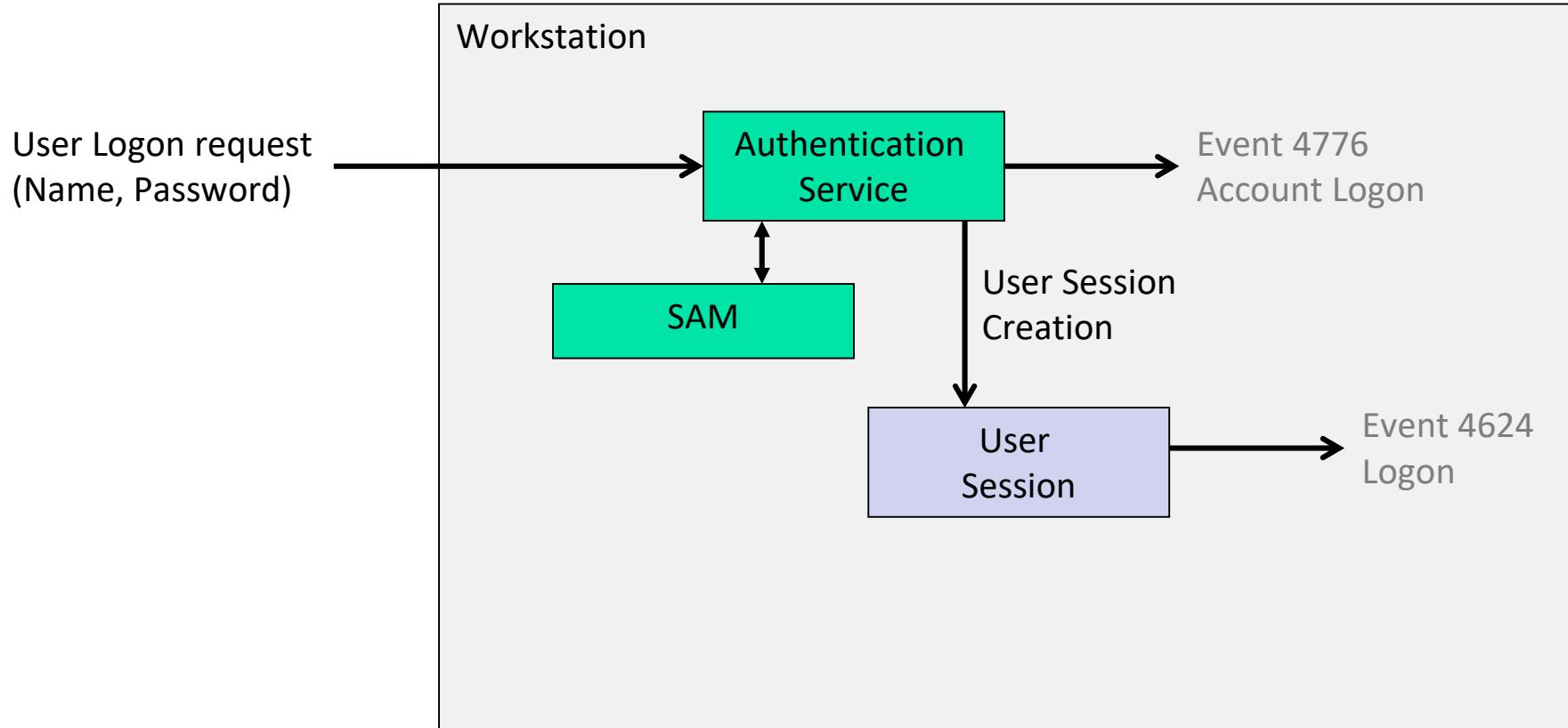
- For many investigations simple filtering of a single log provides the solution.
- Custom views provide the same functionality, plus:
  - The ability to incorporate several event logs (e.g. from different sources) into a single list.
  - Store and recall settings for repeated use.

# LOGON

Reminder: Logon events are in SECURITY.evtx

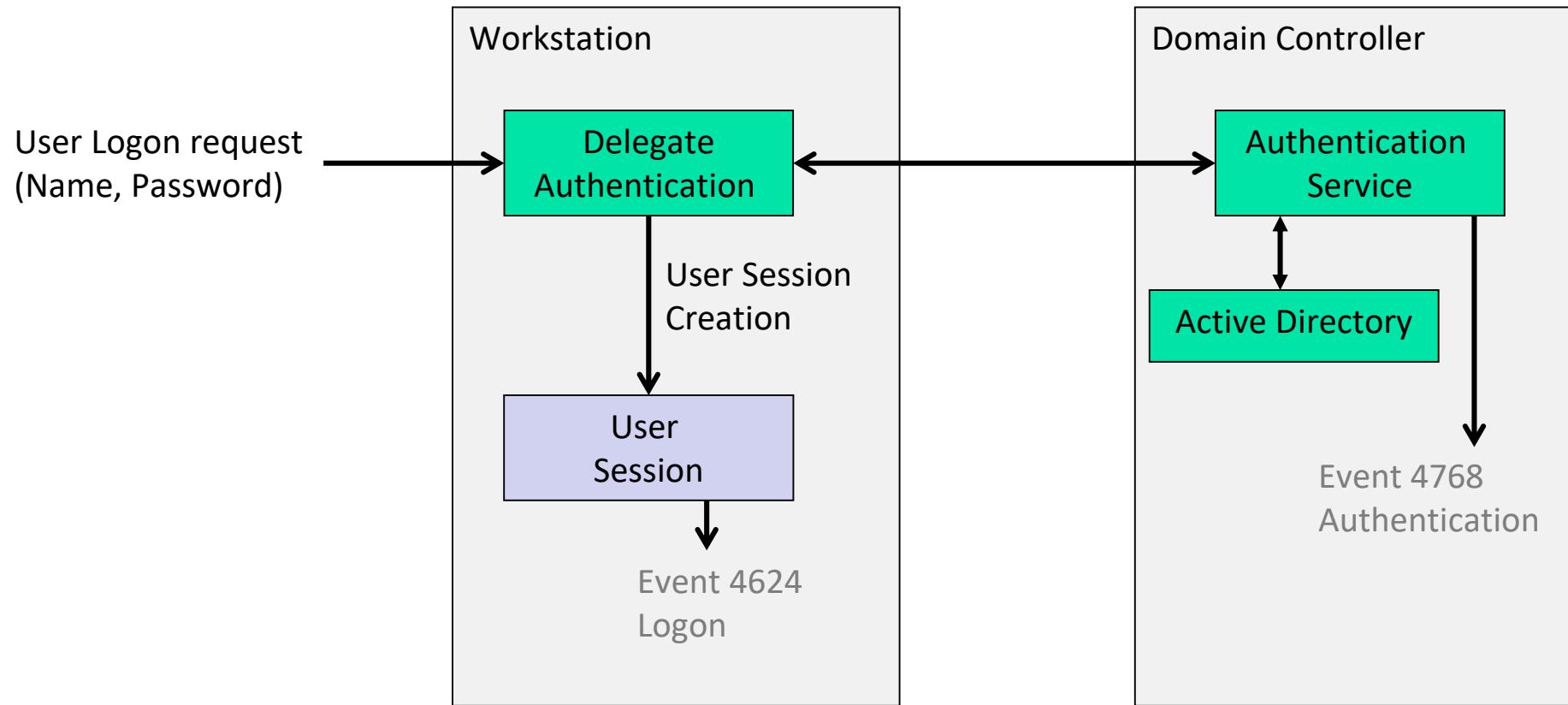
See: Security Audit Events for Windows 7 and Windows Server 2008R  
<http://www.microsoft.com/download/en/confirmation.aspx?id=21561>

# Workstation (LOCAL) Logon



Authentication and service creation are two separate actions, and (if enabled) generate two separate 'Logon' events

# Domain Authenticated Logon



# Logon to Network Services

- e.g. File Server, for Shares or Folder Redirect.
- Usually carried out by the system rather than the user.
  - Sometimes with user impersonation, depending on the service.
- Network logon is required to access services.
  - Results in logging event 4624 on the workstation
- Usually timeout quickly to optimise resource usage.
  - So expect to see several in a session lifetime.
  - Group policies refreshed every 90 minutes.
  - File servers release sessions when files are closed.

# Logon Types

Event 4624 carries a type code; common types are:

| Type | Meaning  |
|------|--|
| 2    | An Interactive logon by a local user.  |
| 3    | Network logon: e.g. Connection to a service or folder on this computer from another networked machine. |
| 4    | Batch logon. i.e. a scheduled task.  |
| 5    | Service logon/start.   |
| 7    | Unlock system (i.e. Password protected screen saver)   |
| 8    | NetworkClearText (i.e. Login with the username/password unencrypted)                                   |
| 10   | Remote Interactive logon. (Remote Desktop, Remote Assistance or Terminal Services)                     |
| 11   | CachedInteractive (login with cached domain credentials while offline from network)                    |

# Some Issues

- Event 4648
  - User ‘login using alternate credentials’.
  - A program is run that requires admin authority via User Account Control.
  - However, it often appears as part of other logon sequences and may be misleading.
- Caching
  - Event recording is highly optimised, a log may not show recent events due to lazy writing.

# Other User Activity Events

| Event | Meaning                      |
|-------|------------------------------|
| 4647  | Logoff – Initiated           |
| 4634  | Logoff – User Session Closed |

| Event | Meaning               |
|-------|-----------------------|
| 4608  | Startup               |
| 4609  | Shutdown              |
| 4778  | Session Reconnected   |
| 4779  | Session Disconnected  |
| 4800  | Workstation Locked    |
| 4801  | Workstation Unlocked  |
| 4802  | Screensaver Invoked   |
| 4803  | Screensaver Dismissed |

# More Issues

- Remember that not all logging will be enabled.
  - Absence of evidence is not evidence of absence.
- System behaviour needs to be understood carefully:
  - E.g. Workstation isn't locked when screensaver is in operation until after screensaver release.
  - Can't distinguish automatic and manual locking.
- Event pairing needs care to avoid pairing the wrong logon-logoff events.

# Time-Related Evidence

SYSTEM.evtx

| Event | Meaning                       |
|-------|-------------------------------|
| 35    | Invoking network time service |
| 37    | Updating system time          |

SECURITY.evtx

| Event | Meaning             |
|-------|---------------------|
| 4616  | System Time Changed |

Network time service updates (35, 37) provide a way of determining if system time stamps are likely to be reliable.

*System Time Changed* events are recorded when a user manually changes the machine time (as well as automatic minor changes)

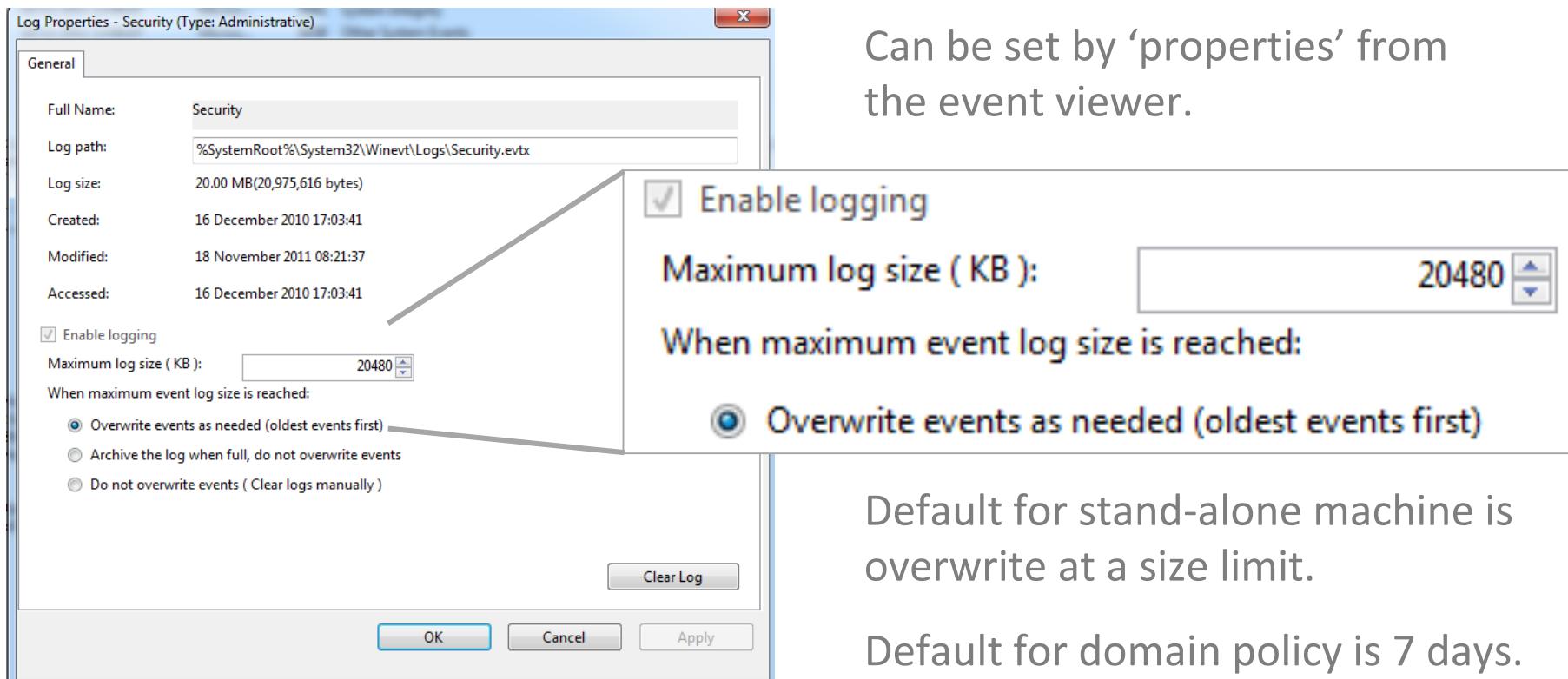
In Windows 7 these are enabled by default.

# Event Log Retention

Policy is at: **SYSTEM**

***ControlSet001\services\eventlog\Security***

***MaxSize, Retention***



# Exercise 14

(still in Windows for this one)

- Use file:

Security\_log\_machine\_12.evtx

This is a log recovered from a machine which is part of a managed domain.

- On 17 November 2011:

- Which users logged on?
- When were the related user sessions?
- Were the sessions online or offline?

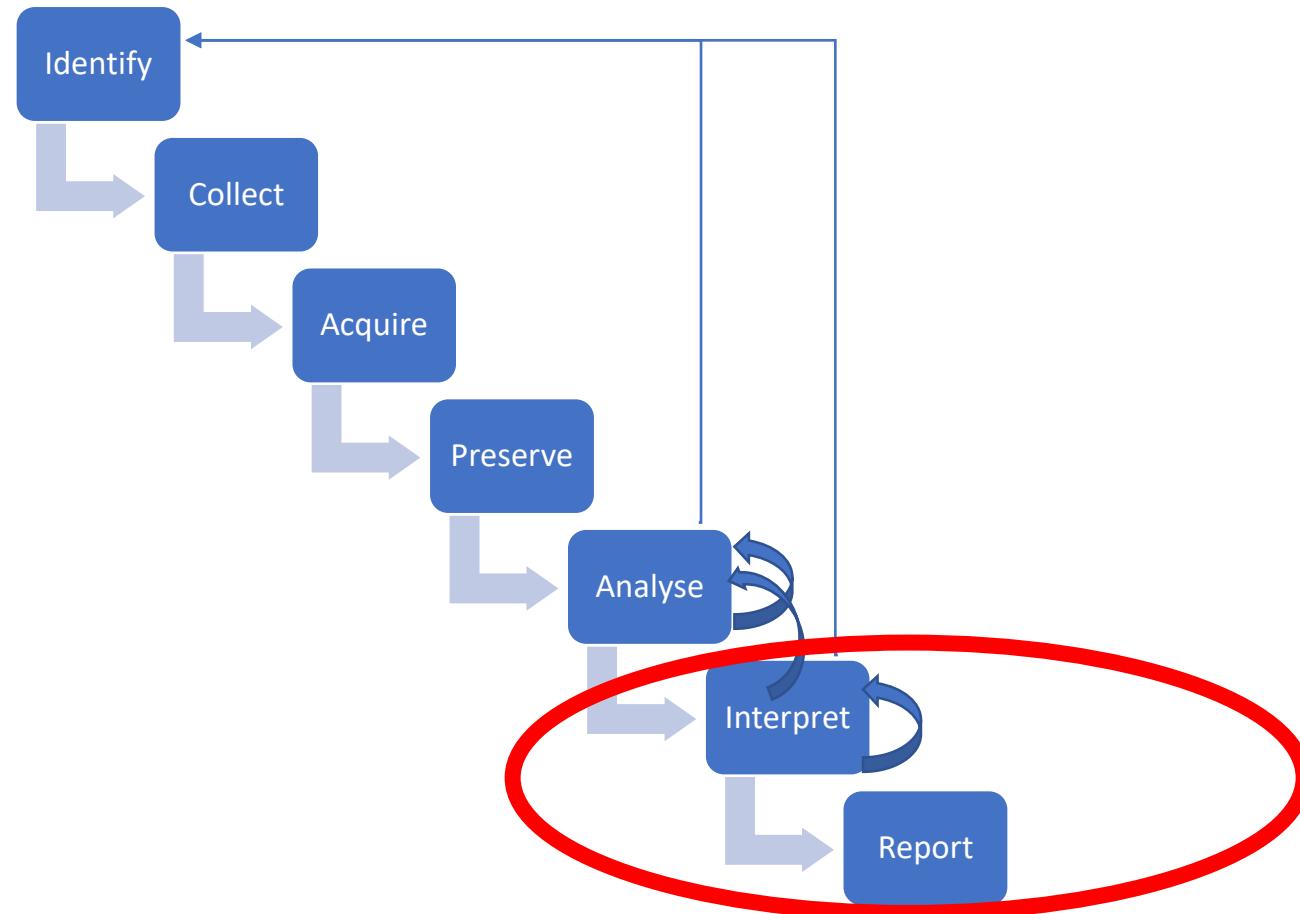


# Interpreting and Reporting Results

# Outputs of analysis

- A set of items of digital evidence that we *believe* are relevant to the investigation
  - Facts
    - Data
    - Gaps
- Need for more analysis
  - Indication of what should be done next
- Identification of (potential) missing PDE sources

# Investigation Phases



# Interpretation

- Based on the results of analysis, what does the DE mean?
- Which meaning is most likely?

# ABC

- Assume nothing
- Believe nothing
- Challenge everything
- Pay attention to detail & search for corroboration

# From earlier - 5WH

| What do we know at the moment? | What information do we need to find out?<br>(5WH) | How are we going to go about finding this information? |
|--------------------------------|---|--|
|                                |   |  |
|                                |   |  |
|                                |   |  |
|                                |   |  |
|                                |   |  |
|                                |   |  |
|                                |   |  |
|                                |   |  |

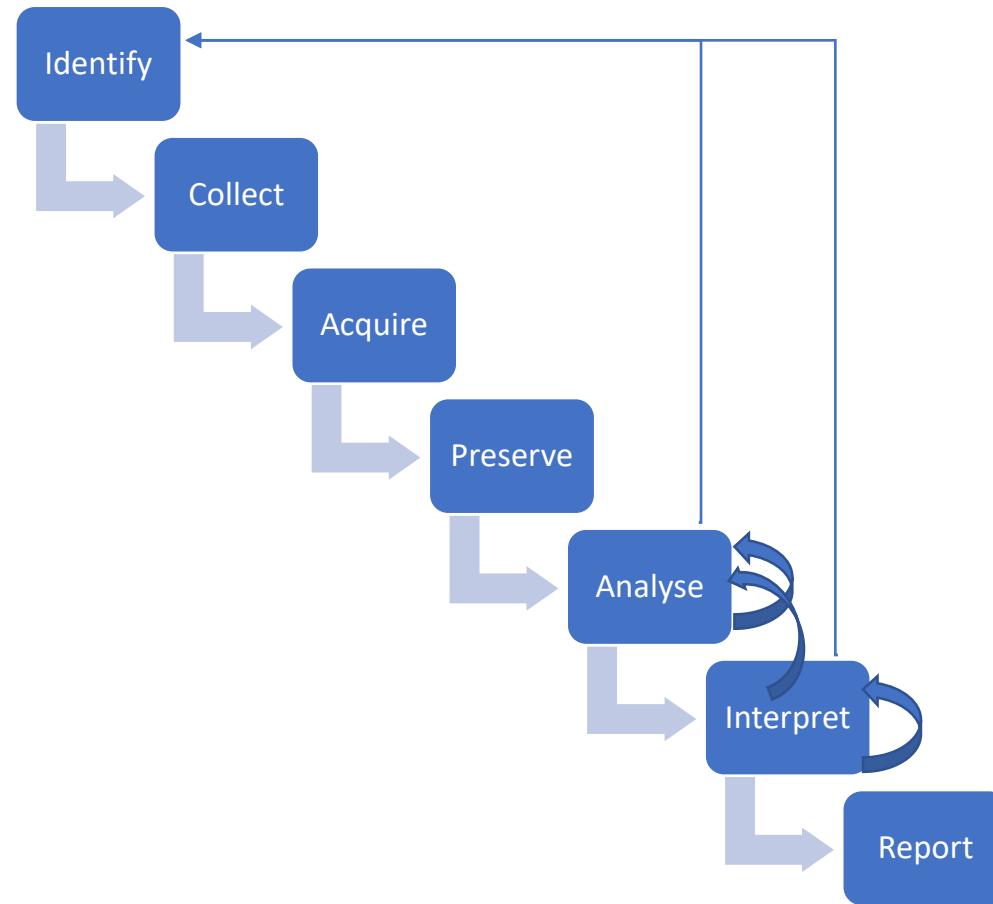
# 5WH

- By answering the 5WH questions, we should have found answers to some (most) of our key investigative questions :
  - What happened?
  - When did it happen?
  - How did it happen?
  - Why did it happen?
  - Who did it?
    - this is the hardest one – but you may have intelligence in your DE
- But ALWAYS approached with an open mind, as a scientist, NOT seeking to prove a hypothesis, but testing various options and following what the data tell you.

But to get there...

Planning

# Investigation Phases



# Phases of the model

1. Identify
2. Collect
3. Acquire
4. Preserve
5. Analyse
6. Interpret
7. Report

There's one missing...

# Phases of the model

## **0. PLAN AND PREPARE**

1. Identify
2. Collect
3. Acquire
4. Preserve
5. Analyse
6. Interpret
7. Report

# Prepare

- Ensuring that you have everything you need to carry out the plan
  - Equipment
  - Staff
  - Management support
  - Budget
- Not necessarily all in-house, but available on-call at short notice
- Depends on your plan

# Planning

“No plan survives contact with the enemy”

But, some plan is better than none

At the very least a plan gives a starting point

# Exercise

- 5-10 minutes
- Produce a plan for how you would start to investigate a possible network intrusion in a small business which has 2 servers and 5 workstations on the premises, permanently connected to the Internet, with the servers running 24x7.
- Be ready to come to the front of the room and present the first 2 or 3 steps of your plan.

# Options for planning

- Incident-based
  - Predict most likely incident types, and plan how to respond to each
    - Comes from internal risk assessment & risk management plans
  - Danger – confirmation bias. (“ when the only tool you have is a hammer, everything looks like a nail”) – tendency to categorise incidents according to the plans available
- Source-based
  - Work out how to ICAPA each potential source
    - Physical devices, systems, subsystems (hardware and software)
    - Devise SOPs for common operations

# SOP?

- Standard Operating Procedure – examples in ISO/IEC 27042
  - Also known as Work Instructions
- Simple, easy to follow, task-specific step by step instructions for key tasks.
  - Opinion about structure is divided, but 27042 suggests “atomicity” and non-branching.
- SWGDE do SOPs quite differently (<https://www.swgde.org/> check under Documents )

# If you have SOPs

- Plan = list of SOPs + guidance on decision about flow
- cf a Michelin-starred restaurant menu.
  - SOP = how to prepare each element on the menu, one course requires several SOPs (recipes!)
  - Diners are free to pick any dish from the menu, the kitchen cannot predict what they will choose, BUT
  - The kitchen can produce anything on the menu quickly, by implementing the recipes for each complete dish (e.g. Beef Wellington, pommes dauphinoise and sautéed green beans = 3 SOPs for 1 dish)
    - i.e. there is a plan for each dish, made up of the SOPs for that dish. Some SOPs are common to many dishes.

# The only thing worse than no plan

- Is an out of date plan
- All plans should be version-controlled and subject to periodic review, especially when new systems or devices are being commissioned
- Tied to asset register – every asset should be linked to one or more SOPs for investigation

# This week's lab.

- Finishing off any incomplete exercises – and then delving into event logs.

# Forensic Analysis (bonus)

User activity hints

# More Evidence...

Operating Systems help their users by providing prompts for previously used files and directories.

So user actions are recorded...

# Why

Most investigations are concerned with understanding what one or more computer users did, and when.

File systems provide valuable evidence, e.g.:

- Documents.

- Deleted Documents.

- Recent Documents.

Internet activity is also important (see later lecture).

The registry provides a further source of information about how the computer was used.

corroboration:  
mutually supportive evidence.

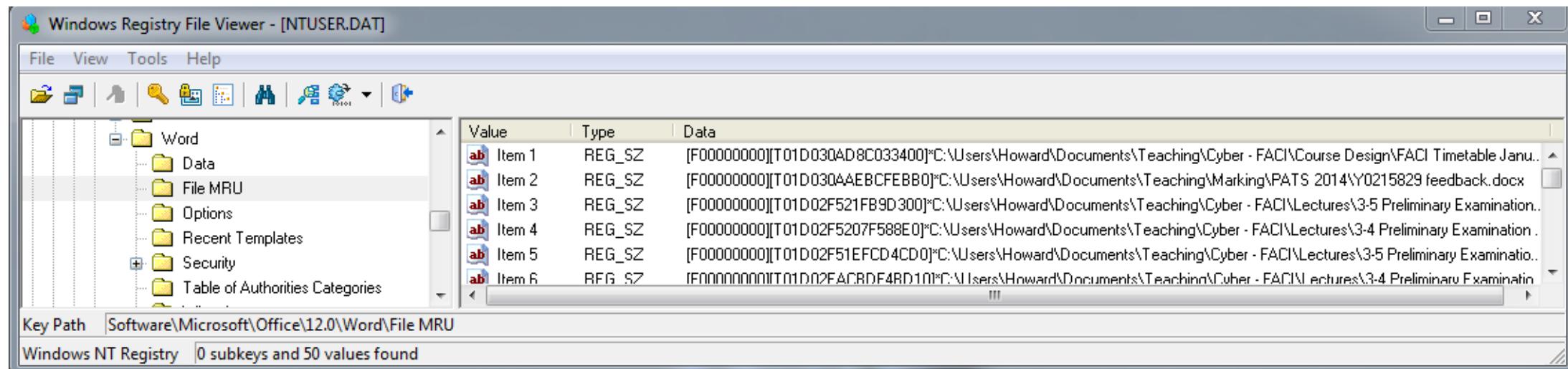
# Contents

MRUs (Most Recently Used) lists for files opened by:  
Applications.  
Dialogue boxes.  
Explorer ('Recent Documents').

Recently launched applications (User Assist Keys).

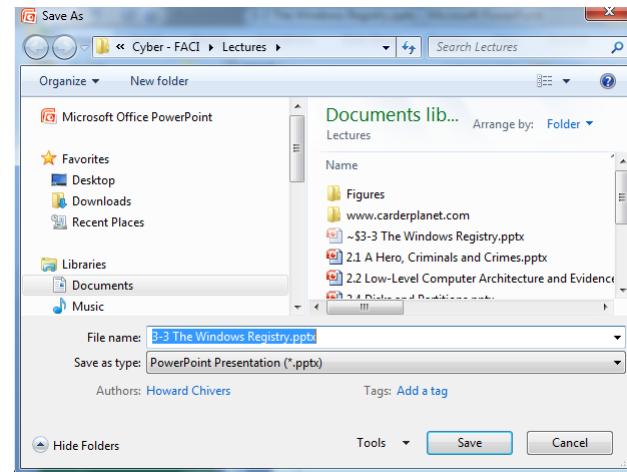
# Most Recently Used

- Allows applications to list recently accessed documents.
  - Located in the user hive (NTUSER.DAT) in an application sub-key under the Software Key, e.g.  
NTUSER.DAT\Software\Microsoft\Office\12.0\Word\File MRU



# Common Dialogue

- Common dialogues provide standard services to applications, such as ‘Save As’.



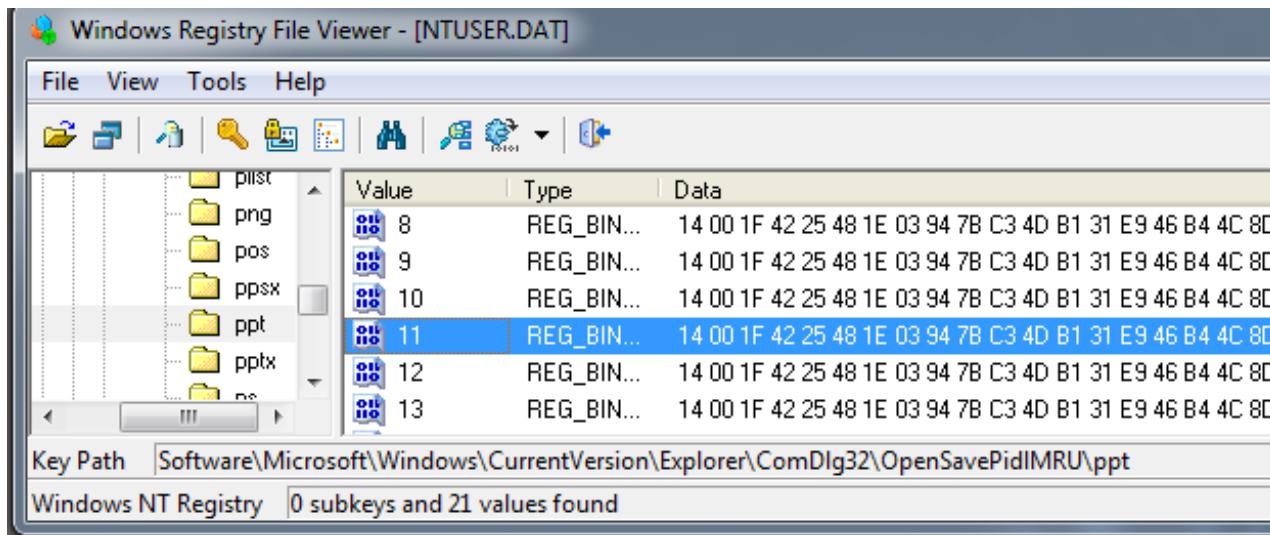
- The registry maintains separate MRU lists for these functions.

NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSavePidMRU

- Under this key there is a separate list for each file extension.

# Common Dialogue Binary Data

- Common Dialog Entries are in a complex binary format.
- However, the relevant file path is usually present in UNICODE.



The screenshot shows a list of registry values for the "ppt" key. The values are displayed as a series of green and blue squares, each representing a character in a UNICODE string. The string starts with "p" and ends with "t.....". The highlighted segments include "p", ".....", "Q.", ".C...\", "U.s.e.r.", "s.\", "H.o.w.a.r.d.", "\\", "D.o.c.u.m.e.n.", "t.s\", "\\", "T.e.a.c.h.", "i.n.g.\", "C.y.b.e.", "r. .-.\", "F.A.C.I.", "\\", "B.a.c.k.g.r.o.", "u.n.d\", "\\", "R.e.g.i.", "s.t.r.y.F.o.r.e.", "n.s.i.c.s...p.p.", "t.....").....", and "t.....".

(From Mitec Data View)

# Access Order

- ComDlg MRUs include a **MRUListEx** value which lists the order of use of the MRU values.

| Value     | Type       | Data  |
|-----------|------------|---|
| 0         | REG_BIN... | 14 00 1F 42   |
| MRUListEx | REG_BIN... | MRUListEx REG_BIN... 0B 00 00 00 09 00 00 00 08 00 00 00 07 00 00 00 0A 00 00 |
| 1         | REG_BIN... | 14 00 1F 42   |
| 2         | REG_BIN... | 14 00 1F 42   |
| 3         | REG_BIN... | 14 00 1F 42   |
| 4         | REG_BIN... | 14 00 1F 42   |
| 5         | REG_BIN... | 14 00 1F 42   |
| 6         | REG_BIN... | 14 00 1F 42   |
| 7         | REG_BIN... | 14 00 1F 42   |
| 8         | REG_BIN... | 14 00 1F 42   |
| 9         | REG_BIN... | 14 00 1F 42   |
| 10        | REG_BIN... | 14 00 1F 42   |
| 11        | REG_BIN... | 14 00 1F 42   |
| 12        | REG_BIN... | 14 00 1F 42   |
| 13        | REG_BIN... | 14 00 1F 42   |
| 14        | REG_BIN... | 14 00 1F 42   |
| 15        | REG_BIN... | 14 00 1F 42   |
| 16        | REG_BIN... | 14 00 1F 42   |
| 17        | REG_BIN... | 14 00 1F 42   |
| 18        | REG_BIN... | 14 00 1F 42   |
| 19        | REG_BIN... | 14 00 1F 42   |

- Here the order is 0x0000000B, etc (0xB, 0x9, 0x8...).
- The 0xB value (= record 11) is the most recent, 0x9 the next most recent, etc.
- If a MRUListEx or MRUList value is not present the order is as numbered with 0 the most recent.

# MRU list time and date

- Recall: dates and times are associated with registry **keys**. (Not with individual values)
- So there is a single time for all MRU values in a list.
- This provides the time of the most recent value.
  - Of course there is no information here about previous values, other than their historical order.

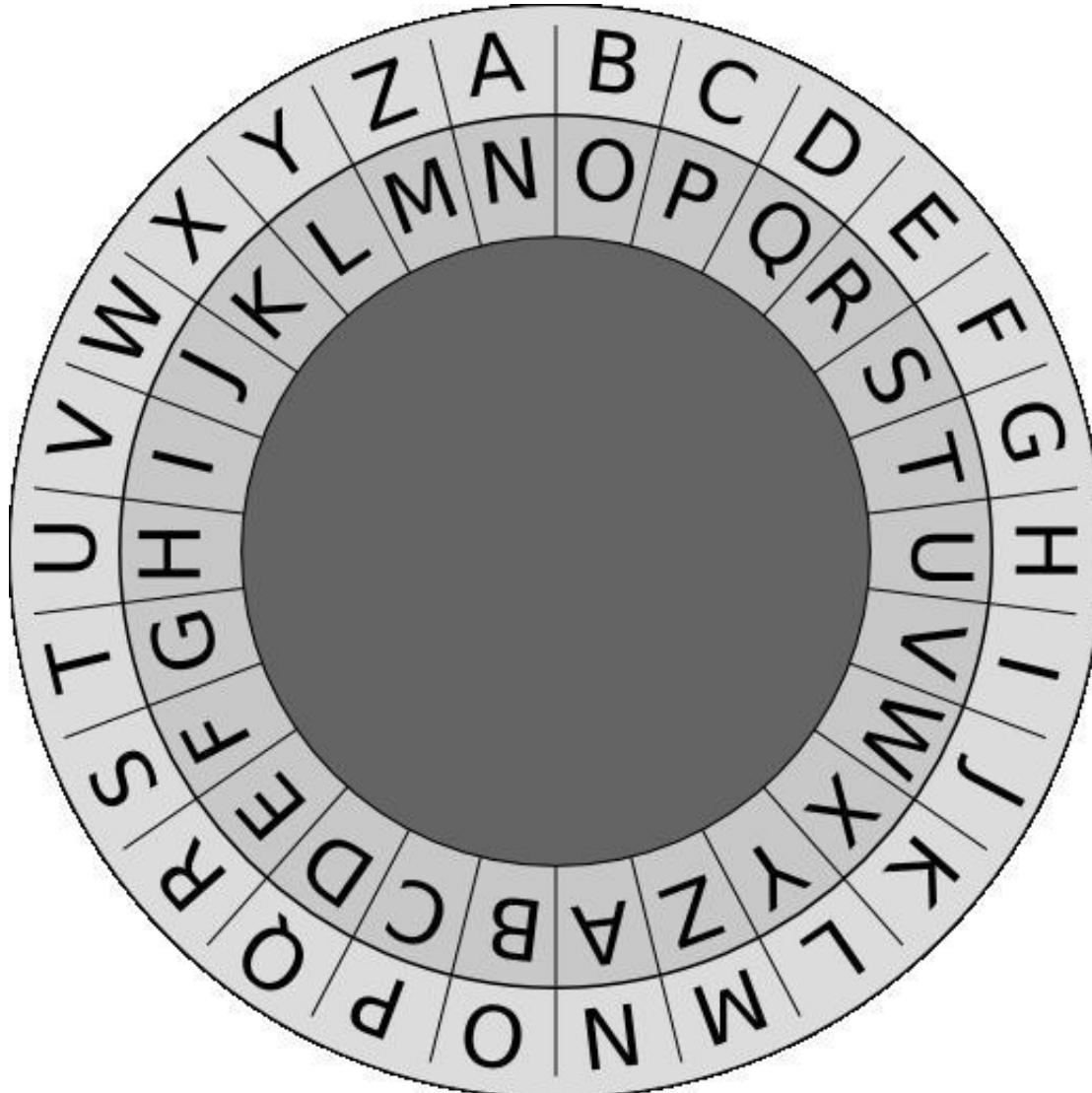
# Recent Documents

- Windows Explorer (the file browser) also records when documents are accessed.
- The format is similar to the Common Dialogue with subkeys corresponding to file extensions.
- The data are usually file paths in UNICODE.
- e.g.:  
NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer\Recent Docs\.ppt

# User Assist

- Stores information about software run using menus and windows explorer.
- This key often contains extensive information about a user's actions.
  - Binary format.
  - File Paths Obfuscated using ROT13.
  - Includes access count and time information with each value.

# ROT13



Obfuscate text by rotating the letters 13 places. (Caesar Cipher / ROT-13)

For unknown reasons popular with Geocache and Defcon puzzles.

Some Windows versions use a different form of encryption.

# User Assist

Registry Path:

NUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist\{\...}\Count

|   |            |                |
|---|------------|----------------|
| HHZH_PGlYPHNPbhag:pgbe                    | REG_BIN... | FF FF FF FF 01 |
| Zvpebfbsg.Jvaqbjf.PbagebyCnary            | REG_BIN... | 46 00 00 00 01 |
| (S380S404-1Q43-42S2-9305-67QR0028SP23)... | REG_BIN... | 46 00 00 00 31 |
| Zvpebfbsg.VagreargRkcybere.Qrsnhyg        | REG_BIN... | 46 00 00 00 01 |
| Zvpebfbsg.Jvaqbjf.Uryccnar                | REG_BIN... | 46 00 00 00 01 |

Under keys:

{CEBFF5CD-ACE2-4F4F-9178-9926F41749EA}  
{F4E57C4B-2036-45F0-A9AB-443BCFE33D9F}

ROT13 Encoded values

Data Includes count and timestamp

# Viewing/Decoding

The Mitec registry viewer is able to show the values using ROT13, but does not correctly decode counts or times (probably using the obsolete Windows XP binary encoding offsets).

One solution is to use RegRipper:

```
UserAssist
Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist|
LastWrite Time Thu Dec 16 17:13:25 2010 (UTC)

{CEBFF5CD-ACE2-4F4F-9178-9926F41749EA}
Thu Jan 15 11:13:31 2015 Z
E7CF176E110C211B (87)
Thu Jan 15 11:13:30 2015 Z
{7C5A40EF-A0FB-4BFC-874A-C0F2E0B9FA8E}\Microsoft Office\Office12\OUTLOOK.EXE (68)
Thu Jan 15 11:11:57 2015 Z
{7C5A40EF-A0FB-4BFC-874A-C0F2E0B9FA8E}\AccessData\FTK Imager\FTK Imager.exe (11)
Thu Jan 15 11:07:56 2015 Z
{7C5A40EF-A0FB-4BFC-874A-C0F2E0B9FA8E} ... ..
```

# Comment

- Reminder – you need several items of evidence to justify conclusions.
- Registry User Histories are an important additional resource.
- Real systems contain thousands of files: evidence of recent actions are often very helpful in providing the starting point for an investigation.

# Additional Information

- Carvey, H., *Windows Registry Forensics*, Syngress
- Stevens, D., *Windows 7 User Assist Registry Keys*, Into the Boxes, January 2010  
[http://intotheboxes.files.wordpress.com/2010/04/intotheboxes\\_2010\\_q1.pdf](http://intotheboxes.files.wordpress.com/2010/04/intotheboxes_2010_q1.pdf)

# Example Registry Keys

- Application MRU under software key, e.g.:

NTUSER.DAT\Software\Microsoft\Office\12.0\Word\File MRU

- Common Dialog MRU list for each function:

NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\<function>\<extension>  
  >     e.g.: \OpenSavePidMRU\ppt

- Recent Documents

NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs\<extension>  
  >  e.g.: \.ppt

- User Assist

NUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist\{\...}\Count  
  {\...}:     {CEBFF5CD-ACE2-4F4F-9178-9926F41749EA}  
              {F4E57C4B-2036-45F0-A9AB-443BCFE33D9F}

