Security Workshop

Dr. Carl Pulley
c.j.pulley@hud.ac.uk

Reverse Engineering

- All systems are built such that *implicit* and *explicit* **assumptions** are made as to how that system will be used
- Reverse Engineering attempts to identify (and so break!) these *developer* **assumptions** for our gain

Dynamic Analysis

- Aim of this analysis step is to enumerate the interaction surface for our software sample
  - detect changes to filesystem, registry, etc.
  - detect attempts to interact with networks
- Can then modify the environment so that everything looks *normal*!
  - API function hooking
  - honeynets and fake servers

Monday, 7 February 2011

Obviously, scapy could be used to sniff and interact with sample software. This approach allows one to log TCP/IP interactions but doesn't allow one to interact with the sample at the application layer.
honeyd, nepenthes and mwcollectd provide excellent low and high interaction honeynets. With these you can study network interactions and worm propagation behaviour (ie. application level behaviour!).
Metasploit can also be used to provide various fake services (but its probably more flexible to use a honeynet setup).
For the really keen, you can use the Python Twisted framework to build your own specialised servers.

Detecting Changes

- If we fingerprint our OS **prior** to running specimen software, can detect changes specimen makes
  - file system hashes (eg. *tripwire*) allow this to be done in production environments
  - *regshot* and *captureBAT* both do this on analysis machines

Monday, 7 February 2011

captureBAT can also log network interactions.
It's also worth mentioning HBGary's Flypaper - this attempts to stop the specimen software from exiting memory (thus, we can always dump from memory) and making any form of network or system changes.

Network Interactions

- As we learn more about a sample, we alter its executing environment by adding in required services
- Can utilise *honeynets* for this purpose
  - low interaction honeynets allow network interactions to be logged
  - high interaction honeynets allow scripted interactions with sample
  - can even fake server vulnerabilities!

Socket Hooking

- An alternative strategy is to simply hook all socket code that the sample uses
- Rationale here is that any network connection is likely to use the Windows socket API (ie. WinSock)
  - by analysing the import tables for the sample, we can determine what WinSock functions it uses

Monday, 7 February 2011

This is basically how captureBAT works.

Automated Analysis

- Sites such as Sunbelt's *CWSandbox* allow analysts to utilise automated sandbox environments
- Using such sites can quickly speed up the analysis of sample code
- Such sites can be fooled as well
  - uploaded sample may not be complete!