# Lecture 12 - 10/29/2019

# Malware

## Phone Worms

- First in 2009 on the Symbian OS
- Would replicate through bluetooth
- Often it will copy itself to the sim card
- Would call premium numbers owned by the attacker

## Phone Trojans

- First showed up around 2004
    - E.g. DroidDream
    - Gave the trojan writer full access to your phone
    - Google cleaned their app store in 2011

## Defenses

- First step:
    - Use the most current versions of everything
    - Set access controls properly

Technologies

## Antivirus software

### Host-based scanners

- **First generation is simple scanners**: would look for signatures.
- **Second generation is Heuristic Scanners**: Looking for code fragments that are associated with malware
- **Third generation is activity traps**: Looking for particular actions like access to files or self replication
- **Fourth generation is full-featured protection**: Combination of the above

#### Generic decryption scanner

- Has CPU emulation, Signature scanners, Emulation control module
- The code will run on the CPU emulator and break periodically, will look for code fragments and signatures during the safe run
- Huge performance degradation

### Behaviour-blocking software

- Integrates with the os to monitor all program behaviour
    - Modification to specific files
    - Formatting disks
    - Secure delete
    - Modification of network components.

## Network based Scanners

- Found on mail servers or corporate firewalls
- Block a virus before it can get on a network or individual system
- A limitation is that it can only look at content, not behaviour.
- Will typically have both an ingress and egress monitor
- Can be useful in detecting botnet activity

## Distributed Intelligence Gathering Approach

- Hybrid between network and host based scanners
- Would have thousands of scanners
- It has a central admin maching

Zero day attack

- **Zero day** or a **day zero attack** is the term used to describe the threat of an unknown security vulnerability in computer software or application for which either the patch has not been released or the application developers were unaware of or did not have sufficient time to address.

# Rootkit detection

- One approach: rootkit revealer
    - Would look at calls to the system through the API and compare those with the results that it got itself
- GMER
- Kernel level rootkit: Reinstall OS

c) Checksum Mers

- Store original hash
- If the file has a different hash when running it has a high probability of being changed.

D) type ‘data’

- Treat all new software as data that can not be executed
- Once you have found it is not a virus, you turn it into an executable.

e) flow distance

- Every time something is created, new file, is defined to have a flow distance of 0
- Every time that is shared with someone, flow distance increases by 1

f) reduce user rights

- Also reduce the rights of the system itself.
- Sandboxes and VMs

g) proof-carrying code

- Any code that you download, carries with it, a proof that it will not do anything.

h) look for unusual language characteristics in program code.

- Evidence of more programmers that were supposed to write it
- If you have the source code, look at the source code. duh.

i) Isolation and separation

- Isolation in the sense that your mission critical stuff is on another machine.
- The princess is not in this castle.

j) education and training

- Dont be stupid, stupid.
-

# Denial of Service Attacks (DoS)

- Famous as fuck
- Different flavours

## Denial of Network Services

- Higher capacity network link sending traffic to something with lower capacity

## Denial of System Services

- Specific types of packets to consume limited system resources.
- I.e. overfilling a buffer

## Denial of Application Services

- Sending packets that send packets that exploit a specific flaw in an application
- Valid requests that are resource intensive.

Examples:

- Do(network)S : Send large volume of ping commands
    - Problem is that all pings will return to you.
    - What we have to do is called Source address spoofing so responses dont return to you.
- Do(system)S : TCP SYN flood
    - Send a whole bunch of TCP SYN requests and receive ACKs but don't send the SYN-ACKS. This will fill the whole table.
    - Prevention:
        - Dont store the table
        - Make the client store the data.
        - https://en.wikipedia.org/wiki/SYN_cookies
- Do(application)S: SIP Flood or HTTP Flood

# DDoS

- Launch DoS Attacks from a large group of machines
- A control hierarchy is often used.
    - Attackers will have handler zombies
    - Handler zombies will have agent zombies
    - Agent zombies do the actual attack
    - Hard to trace back to actual attackers.

## Reflector attacks

- Attacker sends packets to a known service, service responds to spoofed service address UDP, DNS, SNPIP, TCP SYN, e.t.c have all been used

## Amplifier Attack

- Similar but multiple responses are generated for each packet sent. E.g. Broadcast.