Overview of Computer Security

CSC 348·648

WAKE FOREST
UNIVERSITY
Department of Computer Science

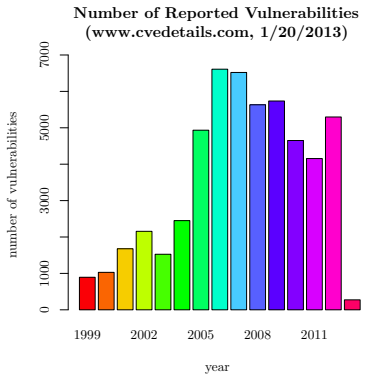
Spring 2013

Computer Security

- Computer systems have become indispensable
  - Store, process, and retrieve information quickly
  - Networks improve communication, collaboration, business
- This revolution has also introduced new risks
  - Businesses are more reliant on computer systems
  - Introduce new ways of obtaining sensitive information (*don't need to be on the inside...*)
  - Disrupt business by disrupting computer activities

Vulnerabilities

- System vulnerabilities are constantly found (*and sometimes reported*)



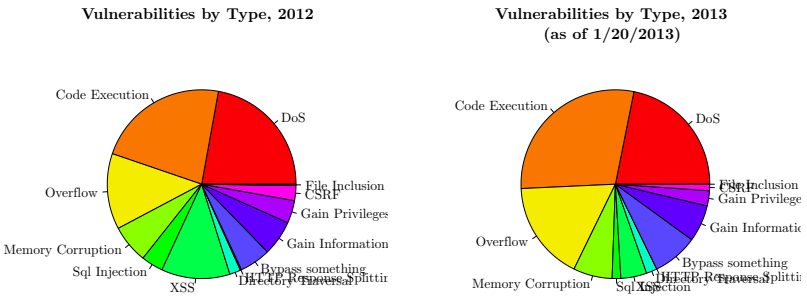
- A new exploit is quickly tried on a variety of systems

  Are security problems only due to poorly designed programs?

- Exploits are **not** always the result of programming errors
- Often problems are due to operating the system **insecurely**
  - Weak or no passwords used
  - Encryption not used for sensitive data
  - Poorly configured software
  - Software not updated according to security risks

  New exploits are complicated, only experts can use them, right?

Vulnerability Types

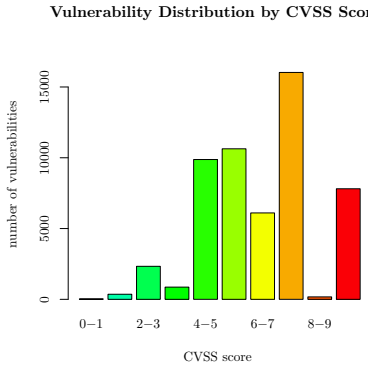


- 5297 vulnerabilities reported in 2012, and so far 274 in 2013

DoS	Denial of Service
Code Execution	Allows execution of arbitrary code
Overflow	Buffer overflow exploit
Memory Corruption	Change values in memory
SQL Injection	Issue SQL unintended queries
XSS	Cross site scripting
Directory Traversal	Access to arbitrary information
Bypass something	as advertised (*for example, permissions*)
HTTP response splitting	Failure to sanitize input values (*helpful for XSS*)
Gain information	as advertised
CSFR	Cross-Site Request Forgery, hijack authentication of user
File Inclusion	Arbitrary execution of "included" file (*alter file path*)

Severity of the Vulnerabilities

- Common Vulnerability Scoring System (CVSS) attempts to score



  - Score of 0 is "no problem" and 10 is "crap your pants"

- Average over all reported vulnerabilities is 6.7

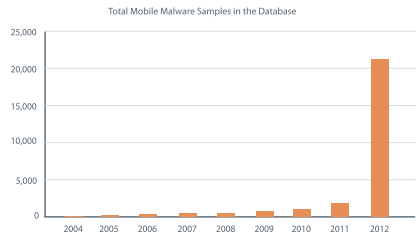
  How do you assign a score for a vulnerability?

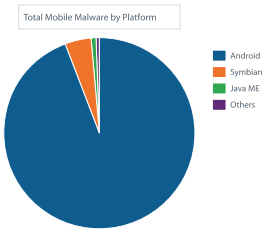
Interesting Stories for 2012

- Mobile malware
- Ransomware
- Rootkits
- Nation-state cyber-espionage
- Java (CVE-2013-0422)
- virut

Mobile Malware

- Android was the largest target for malware (McAfee)





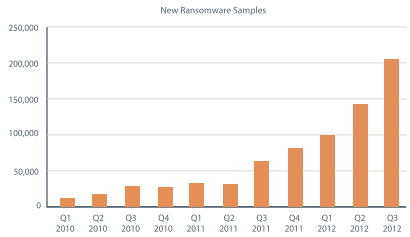
- Majority of malware was either spyware or botnets

> *Automatically purchases apps from a third-party Android market. It intercepts and resends the confirmation codes sent by the market, silently buying the apps. MarketPay.A deletes any other billing messages to keep the user unaware. – McAfee*

Why is Android the preferred target?

Ransomware

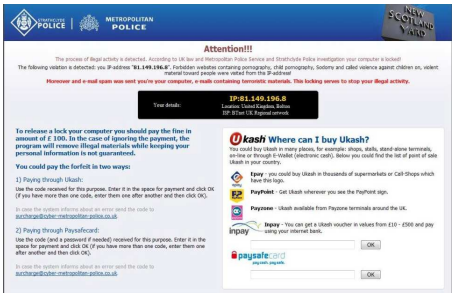
- Makes data inaccessible, or threatens to, unless money paid



- Majority of malware was either spyware or botnets

> *There are many ways in which the victims are infected. In addition to links in emails or messages in social networks, pay per install is a popular method. In this type of attack, computers that are already part of a botnet are further infected with additional malware... Recently drive-by downloads have also played a big role. – McAfee*

- Police ransomware is increasing



> *Much of the activity has been "police" ransomware, which claims to come from a law enforcement agency, accuses the user of visiting illegal websites, locks the computer, and then asks for payment of a fine to unlock the device. – McAfee*

  - Researcher claimed that in one day, hacker group infected 18,941 computers (95% success rate) with ransomware
  - Approximately 15% paid the ransom, for a total of $400k

A Few Ransomware Cases

- Hackers stole data from CreditPret (French financial company) and wanted €20,000 to prevent its disclosure
  - 8/25/2012 they reduced the amount and finally released the information due to non-payment
  - Similar threats against Elantis, and AmeriCash Advance
- On 9/4/2012 hacker(s) stole Mitt Romney's financial records from Pricewaterhouse Cooper
  - Demanded (via Pastebin) $1M converted into Bitcoins

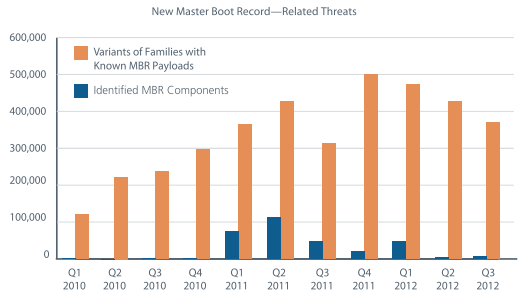
```
The deal is quite simple.  Convert $1,000,000 USD to
Bitcoinds (Google if if you need a lesson on what Bitcoin
it) using one of the various markets available out in the
world for buying...  Failure to do this before September
28, the entire world will be a lowed to fire the documents
with a publicly released key to unlock everything...
```


  - In November Secret Service searched a home in Tennessee
  - *The smoking gun?* Digital photos of cats
- On 9/16/2012 Hackers claimed to have customer info from 300 Webassur-designed databases
  - On 9/22/2012 a company paid $3k USD to the hackers

Is it illegal in the US to pay a ransom?

MBR Rootkits

- Rootkits infect portions of the OS/kernel to hide their existence



- Malware performs the following operations on boot so that it executes before OS starts, so the new boot sequence becomes
  1. Loads itself into memory
  2. Hooks the interrupt used for disk read and write
  3. Passes control to the old MBR

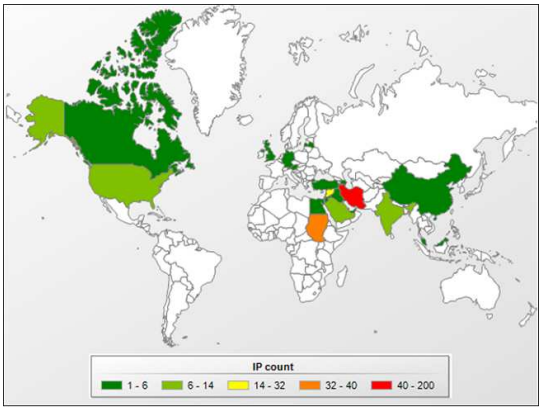
- Once infected, Windows starts as though the malware was not present
  - For example, malware able to intercept all disk reads and writes
  - Mebroot intercepts network packets and establishes its own private networking stack that it uses to open a back door

> *The only known defenses against bootkit attacks are the prevention of unauthorized physical access to the systema problem for portable computersor the use of a Trusted Platform Module configured to protect the boot path.*

Flame

- Flame (Flamer or sKyWIper) was discovered in May 2012
  - Believed to be a nation-state cyber-espionage
  - Designed to spy on various entities
- Can spread to other systems over a local network or USB stick
  - Consisted of more than 20 MB of modules, coded in the Lua scripting language with compiled C++ code linked in
  - Perform a wide array of functions such as audio interception, bluetooth device scanning, document theft, screenshots from the infected machine, and messin' wit yo GF or BF...
  - Used a fake Microsoft certificate to perform a man-in-the-middle attack against Windows Updates (can infect patched Windows 7)
  - Supports a `kill` command which removes the malware (and evidence) from the computer

- Infections in Iran, Israel, Sudan, Syria, Lebanon, Saudi Arabia, and Egypt, with a "majority of targets" within Iran

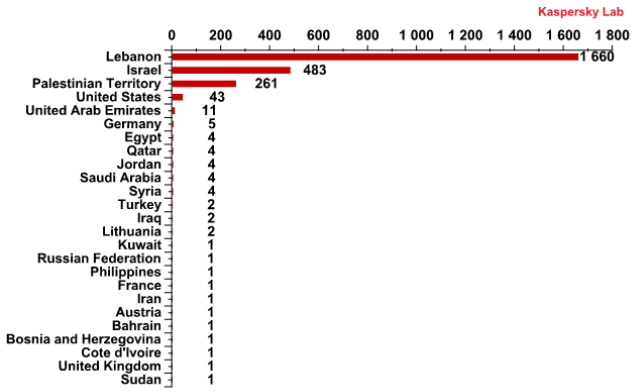


Kaspersky researchers believe Flame developers worked together with Stuxnet developers... "is certainly the most sophisticated malware we encountered during our practice; arguably, it is the most complex malware ever found."

Gauss

- Gauss is another type of espionage malware discovered in August 2012
  - Nation-state cyber-surveillance and an online banking Trojan
- Spreads to other systems via USB stick
  - Uses LNK vulnerability also used in Stuxnet and Flame
- Gauss module is only about 200 KB (one-third of main Flame module)
  - Steal credentials for online banking systems and payment methods (PayPal, Citibank, MasterCard, American Express, Visa, eBay, Gmail, Hotmail, Yahoo, Facebook, Amazon and some other Middle Eastern banks)... *why?*
  - Also steals info about specifics of network interfaces, computers drives, BIOS, and the 411 on yo GF or BF...
  - Includes an unknown, encrypted payload which is activated on certain specific system configurations.
  - Will limit the number of times it is copied/USB-stick

- Infected more than 2,500 systems in 25 countries with the majority, 1,660 infected machines, being located in Lebanon



"Gauss is a nation state sponsored banking Trojan which carries a warhead of unknown designation," Kaspersky wrote. "The payload is run by infected USB sticks and is designed to surgically target a certain system (or systems) which have a specific program installed. One can only speculate on the purpose of this mysterious payload."

Flame vs Gauss

Feature	Flame	Gauss
Modular architecture	Yes	Yes
Using kernel drivers	No	No
.OCX files extensions	Yes	Yes
Configuration settings	Predefined in main body	Stored in registry
DLL injections	Yes	Yes
Visual C++	Yes	Yes
Encryption methods	XOR	XOR
Using USB as storage	Yes (hub001.dat)	Yes (.thumbs.db)
Embedded LUA scripting	Yes	No
Browser history/cookies stealer	Yes (soapr32/nteps32)	Yes (winshell)
CVE2010-2568 (.LNK exploit)	Yes (target.lnk)	Yes (target.lnk)
C&C communication	https	https
Log files/stolen data stored in `%temp%`	Yes	Yes
Zlib compression of collected data	Yes	Yes
Messin' wit GF/BF	Yes	Heck-Yes

The 411 on CVE-2010-2568

- Originally reported 07/22/2010
  - Reported for malware that leverages CVE-2010-2772 in Siemens WinCC SCADA systems
- Potentially vulnerable systems include, Windows Shell in Microsoft Windows XP SP3, Server 2003 SP2, Vista SP1 and SP2, Server 2008 SP2 and R2, and Windows 7
  - Allows local users or remote attackers to execute arbitrary code via a crafted `.LNK` or `.PIF` shortcut file, which is not properly handled during icon display in Windows Explorer
- CVSS base score is 9.3

Java Fail

- Multiple vulnerabilities reported for Java 7
  - Fails to restrict access to privileged code that can allow a remote, unauthenticated attacker to execute arbitrary code
- Java JRE plug-in provides its own Security Manager
  - Typically, a web applet runs with a security manager provided by the browser or Java Web Start plugin

> *"If there is a security manager already installed, this method first calls the security manager's checkPermission method with a `RuntimePermission("setSecurityManager")` permission to ensure it's safe to replace the existing security manager. This may result in throwing a SecurityException." – Oracle*

- Exploit uses a vulnerability in the Java Management Extensions (JMX) MBean components, to access restricted classes
  - Then use a second vulnerability involving recursive use of the Reflection API via the `invokeWithArguments` method of the `MethodHandle` class
  - Now an untrusted Java applet can escalate its privileges by calling the the `setSecurityManager()` function to allow full privileges, without requiring code signing
- Oracle Security Alert CVE-2013-0422 states that Java 7 Update 11 addresses this (CVE-2013-0422) and an equally severe, but distinct vulnerability (CVE-2012-3174)
  - It has been noted that only the reflection vulnerability has been fixed and that the JMX MBean vulnerability remains...

So just how bad is this? What is the CVSS score?

CVE-2013-0422 Score

- A CVSS score has been assigned to the Java exploit
  - The base score is 10 (*yes, crap your pants*)
  - Impact score is 10 (*yes, face punch*)
  - Exploitability score is 10 (*yes, anyone can punch your face*)
- The base score is determined using the following characteristics

Access Vector : Network	Confidentiality : Complete
Access Complexity : Low	Integrity : Complete
Authentification : None Required	Availability : Complete

> *"[It has been reported that] a fully "weaponized" executable that exploits the bug was being advertised for $5,000 in an underground Internet forum. " – Ars Technica*

virut

- CERT Polska started to dismantle the "virut" botnet 1/17/2013
  - Working with NASK, CERT took control of 23 `.pl` domains
  - Consisted of 300k machines, infected 890k machines, and accounted (involved with) for 5.5 of malware infections
  - virut was among the largest Pay-Per-Install (PPI) services available

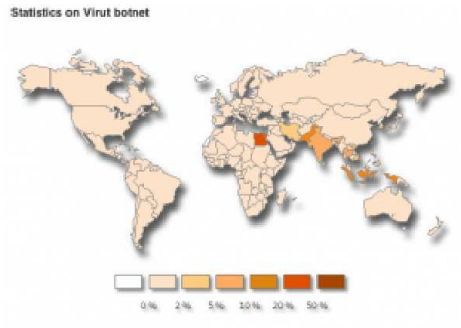


Figure 2. Virut global detections, based on sinkhole data

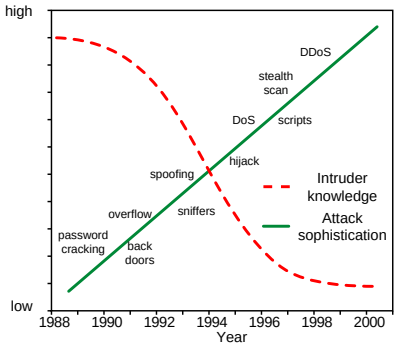
Pay-Per-Install Service

- Historically worms were used to self propagate malware
  - Server-side exploits allowed propagation with little (if any) user (administrator) interaction
  - Only required a network connection
  - While these exploits exist, they are difficult to find
- Focus is now on the client-side and social engineering
  - Most client-side exploits browser/plug-in oriented
  - Exploits also require victim interaction (visit, click, and download)
- Therefore clint-side exploits requires a victim to visit an infected website, click on a malicious link, or open an email attachment...
  - That's a lot of work for a hacker like Chukar...
  - Many malware authors do not have the resources to distribute their malware, *so what is Chukar to do?*

- Pay-Per-Install (PPI) networks enable large-scale malware distribution
  - PPI is a profit sharing model where "network affiliates" distribute the malware and get paid a commission
- Assume "Kingpin" (entrepreneurial hacker) creates a PPI for hackers like Chukar (malware writer, but no way of distributing)
  1. Kingpin creates a PPI website
  2. Kingpin recruits "affiliates" who will receive malware to install
  3. Kingpin then charges malware-writers (like Chukar), to distribute their malware via the affiliates
- So the PPI connects malware-writers/owners (or hackers who just want to spread malware, develop a botnet, etc...) with those that are good at installing (finding vulnerable computers)

Powerful Tools in the Hands of Dumb People

- Early exploits required advanced (inside) knowledge
  - *Programmers gone bad...*



- New exploits can extremely complicated and damaging
  - Scripts written to perform the exploit
  - Packaged so anyone can use (*script kiddie*)

Basic Security Components

Computer security consists of the following components

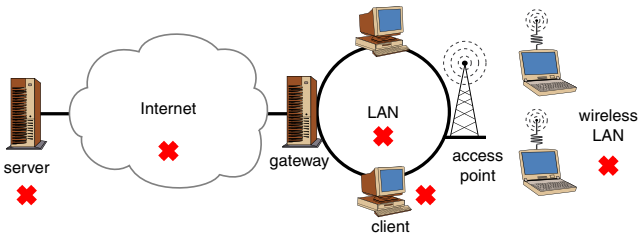
- **Confidentiality**
  - Concealment of information or resources
  - Also applies to the existence of data or resources
  - Mechanisms include encryption and access control
- **Integrity**
  - Trustworthiness of information or resources
  - Preventing and detecting improper or unauthorized change
  - Includes data integrity and origin integrity (authentication)
- **Availability**
  - Ability to use the information or resource as desired
  - Denial of Service (DoS) is an attack on availability

Threats and Attacks

- A threat is a potential violation of security
  - The action is called an **attack**

  Threats are always associated with networks, right?

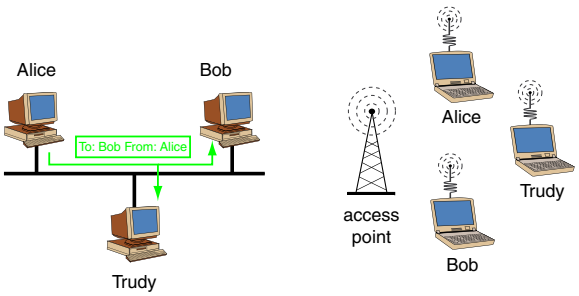
- Possible points of attack



- These are possible points of attack, *what attacks are possible?*
  - Consider the following categories

Snooping

- Unauthorized interception of information
  - Someone listening to (or reading) communications
  - Considered a *passive* attack on confidentiality
- Network sniffers (simple program) are one example
  - Sniffer is able to listen to LAN traffic



  - Effective for shared medium networks

Encrypting the data solves snooping, right?

- Traffic analysis can provide information
  - Cannot read information, but notice conversations...
- Microsoft shared folders
  - Referred to as a *NetBIOS* attack
  - Connect to a computer and view any shared information

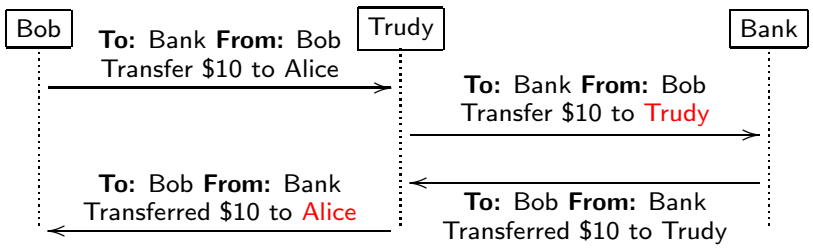
Terminal

```
C:/windows> nbtstat -a 127.0.0.1
C:/windows> net view \\127.0.0.1
C:/windows> net use K: \\127.0.0.0\CDISK
C:/windows> K:
```

- Shoulder surfing... (*physical security is difficult*)

Modification or Alteration

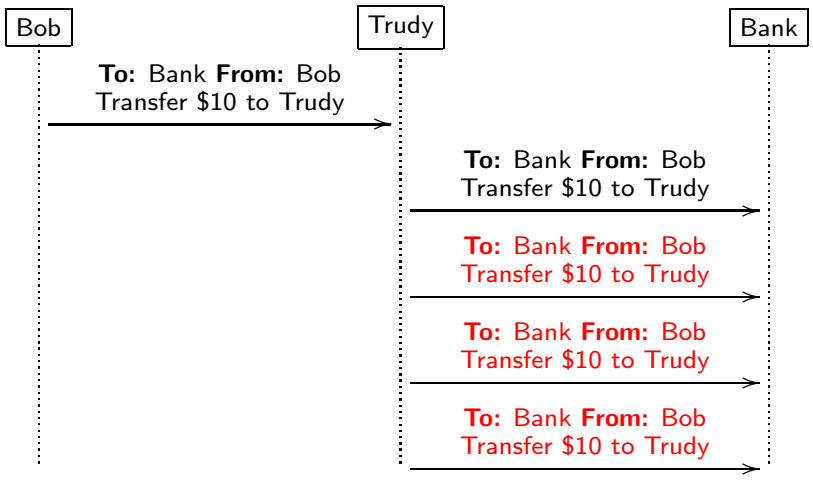
- Unauthorized change of information
  - Considered an *active* attack
  - Attacker modifies data transmitted between two computers
- (Wo)Man in the Middle (MiM) Attack



- Integrity services counter this threat

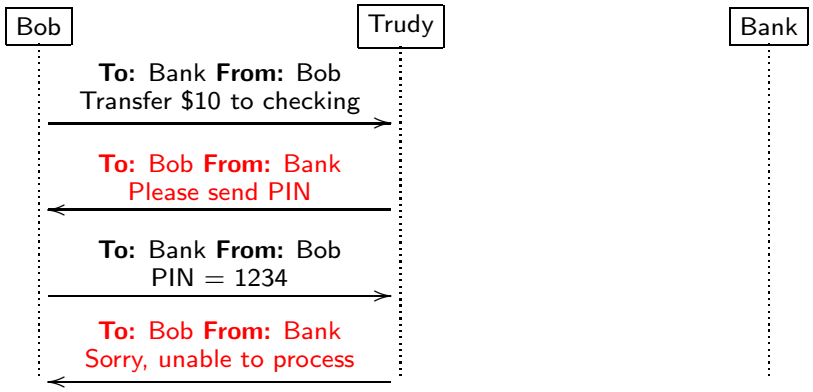
Replay

- Passive capture of legitimate data
  - Subsequent retransmission to produce unauthorized effect



Masquerading or Spoofing

- Impersonation of one entity by another
  - A form of deception and usurpation
- Victim connects to a machine that is an attacker



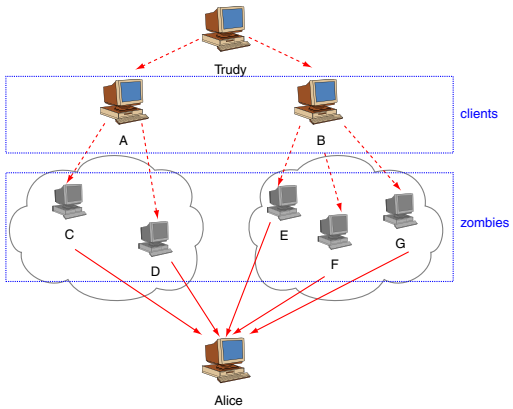
- This includes email...

```
Date:  Fri, 9 Oct 2009 00:49:18 +0200 (CEST)
Subject:  Mail.  Response Needed
From:  "Wfu Team" <verification@dishmail.net>
Reply-To:  verification@dishmail.net

Account Subscriber,
Due to excess abadoned webmail accounts, We are currently performing
maintenance on our Digital webmail Server to improve the spam filter
services in our webmail systems for better online services to avoid
virus and spam mails.  In order to ensure you do not experience
service interruption, respond to this email immediately and enter
your UserID here (********) password here (********) and future
password here (********).

Wfu Team
```

- Again, a network is not required for these attacks
  - Loading software from a CD that contains a backdoor
  - Trojan horse programs
  - Viruses

Denial of Service

- Long term inhibition of service
  - Attacker causes a resource overload at the victim
  - Requires multiple *zombie* computers



- Very difficult to find the attacker

Anatomy of an Actual Attack

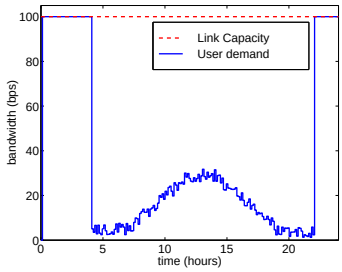
Typically multiple attacks performed for an objective, consider the following **script-kiddie** example *(names removed to protect the innocent)*

1. Attacker scanned to identify computers with a new vulnerability
2. Buffer-overflow used to compromise these machines
3. Attacker loaded password sniffer on compromised machines
   - Looking for passwords for other machines/accounts
4. Machines emailed passwords back to attacker
5. Newly compromised machines loaded with DoS attack (zombies)
6. All the compromised machines DoS attack on an Austrian ISP

How was the compromise finally discovered?

Actual Attack Detection

- After receiving an email from Austria and looking at the traffic...



- Aftermath of the attack was worse, had to determine the following
  - *How did the attack occur?*
  - *Which machines were effected?*
  - *What information was lost?*
- This is an example of how **not** to provide security
  - A security policy had **not** been defined, implemented, ...

Determine Your Risks

Before defining a policy, determine the risks

- Identify and assign values to assets
  - What information and resources are critical for business
  - What would be the cost if information was damaged, etc...
- Prioritize assets
- Determine vulnerability to threats and possible damage
  - *How are the information and resources accessed?*
  - *Who should have access to the information and resources?*
- Select cost-effect safeguards

With this information you can start a **security policy**

Policy and Mechanism

- Policy - Statement of what is and what is not allowed
  - Specify actions that are allowed and disallowed
  - Provides guidance when an attack occurs
- For example, a simple policy could state

> *"Only HR employees are allowed to access payroll information. Payroll information is never to be accessed from outside the company (regardless of medium) without written permission of the Director of HR."*

Policy and Mechanism

- Given the policy statement, determine the components
  - Specific actions that are allowed and disallowed
- Mechanism is attached to each component
  - A method, tool, or procedure for enforcing the policy

Policy Component	Mechanism
Payroll accessed only by HR	Password protection
Never transmit sensitive data in plaintext	Encryption
Do not allow logins from outside	Disable `telnet`
Patch buffer-overflows	Upgrade software
Finger print certain files	MD5 hash
Watch for scanning	Sniffer
Back-up files, keep off-line	CD-ROM weekly

Goals of Security

Given a policy's specification, it can perform the following

- **Prevention**
  - A mechanism to prevent a compromise (e.g., passwords)
  - Often can be cumbersome and interfere with system use
- **Detection**
  - Useful when an attack cannot be prevented
  - Goal to determine when an attack is underway
  - For example, detecting three consecutive wrong passwords
- **Recovery**
  - Assess and repair damage of an attack (disaster recovery plan)
  - Requires logs, audits, and information back-up

The more you invest in prevention the more you must invest in detection

Private Property Example

- Prevention - lock doors, bar window, walls around property
- Detection - Inventory items, burglar alarms
- Reaction - Call police, make insurance claim

Policy and Goals

- A good policy should cover each type of goal

Policy Component	Mechanism	Goal
Payroll accessed only by HR	Password protection	Preventive
Never transmit sensitive data in plaintext	Encryption	Prevention
Do not allow logins from outside	Disable `telnet`	Prevention
Patch buffer-overflows	Upgrade software	Prevention
Finger print certain files	MD5 hash	Detection
Watch for scanning	Sniffer	Detection
Back-up files, keep off-line	CD-ROM weekly	Recovery

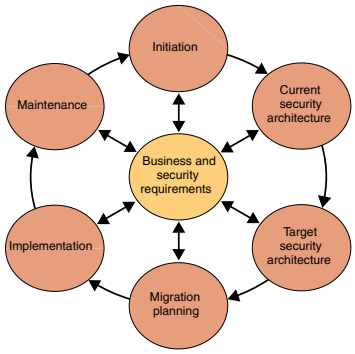
- Multiple types goals are present in the above policy
  - Multilayer defenses are better

Are the Policy and Mechanisms Sufficient?

- Must answer the following questions
  - *Is the policy correct? Does it describe a* **secure** *system?*
  - *Can the policy be enforced by the mechanisms chosen?*
- Trusting the mechanism requires several assumptions
  1. Each mechanism is designed to implement a part of the policy
  2. Union of the mechanism implements all policy aspects
  3. Mechanism implemented, installed, and administered correctly
- The security policy is **never** complete
  - New technology and threats are continually introduced

Putting It All Together

- The preceding topics help form a secure system
  - They appear to be a linear sequence of events
- Consider threats ⇒ Analyze risk ⇒ Create policy ⇒ Determine mechanisms ⇒ Implementation ⇒ Maintenance



- Actually it is a **cycle** that never ends

Fundamental Design Decisions

1. *Where to focus security controls?*
   - Focus may be on, data, operations, users, ...
2. *Where to place security controls?*
   - Applications, services, OS, hardware, ...
3. *Complexity or assurance?*
   - Simple and highly assured, or feature-rich environment

   What? Is complex and assured not an option?
4. *Location of control?*
   - Centralized or decentralized

   Advantages and disadvantages of each?
5. *Block access to layer below*

Network Threats

- Wireless networks (802.11x is the most common)
  - Shared medium, therefore sniffing is easy
  - Encryption would help, typically **not** enabled
  - Wireless Encryption Protocol (WEP) for 802.11x is **not** secure

  Should a business deploy a wireless network?
- Peer-to-peer networking
  - A new paradigm for sharing data (*as opposed to the client server model*)
  - Computers transfer data directly to one another
  - **Trust** is the issue: *Is what I am downloading safe? Is the other computer who they claim to be?*

In Code We Trust

- What code can we trust?
  - Consider `login` or `su` in Unix
  - Is RedHat binary reliable?
  - Does it send your password to someone?
- Can't trust binary so check source, recompile
  - Read source code or write your own

  Does this solve the problem?

Compiler Backdoor

- This is the basis for Ken Thompson's attack
  - Compiler looks for the `login` program source code
  - If found, insert login backdoor (allow special user to log in)
- How do we solve this? Inspect the compiler source?

C Compiler is Written in C

- Change compiler source

```
1  compiler(S) {
2        if (match(S, "login-pattern")) {
3              compile (login-backdoor)
4              return
5        }
6        if (match(S, "compiler-pattern")) {
7              compile (compiler-backdoor)
8              return
9        }
10       .... /* compile as usual */
11   }
```


Is Detection Possible

- Compile this compiler and delete backdoor tests from source
  - Someone can compile login and get login with backdoor
- Simplest approach to detection?
  - Compiling the compiler twice (once with suspected compiler and with a trusted compiler) might identify the backdoor
  - Make code for compiler backdoor output itself

> *The moral is obvious. You can't trust code that you did not totally create yourself. (Especially code from companies that employ people like me.) No amount of source-level verification or scrutiny will protect you from using untrusted code. In demonstrating the possibility of this kind of attack, I picked on the C compiler. I could have picked on any program-handling program such as an assembler, a loader, or even hardware microcode. As the level of program gets lower, these bugs will be harder and harder to detect. A well installed microcode bug will be almost impossible to detect. – Ken Thompson, 1984*

7 Types of Hackers

Understanding motivation is helpful, list by Roger Grimes of InfoWorld

1. **Cyber criminals** Professional criminals comprise the biggest group of malicious hackers, using malware and exploits to steal money. It doesn't matter how they do it, whether they're manipulating your bank account, using your credit card numbers, faking antivirus programs, or stealing your identity or passwords. Their motivation is fast, big financial gain.
2. **Spammers and adware spreaders** Purveyors of spam and adware make their money through illegal advertising, either getting paid by a legitimate company for pushing business their way or by selling their own products. Cheap Viagra, anyone? Members of this group believe they are just "aggressive marketers." It helps them sleep at night.
3. **Advanced persistent threat (APT) agents** Intruders engaging in APT-style attacks represent well-organized, well-funded groups – often

located in a "safe harbor" country – and they're out to steal a company's intellectual property. They aren't out for quick financial gain like cyber criminals; they're in it for the long haul. Their dream assignment is to essentially duplicate their victim's best ideas and products in their own homeland, or to sell the information they've purloined to the highest bidder.

4. **Corporate spies** Corporate spying is not new; it's just significantly easier to do, thanks to today's pervasive Internet connectivity. Corporate spies are usually interested in a particular piece of intellectual property or competitive information. They differ from APT agents in that they don't have to be located in a safe-harbor country. Corporate espionage groups aren't usually as organized as APT groups, and they are more focused on short- to midterm financial gains.
5. **Hacktivists** Lots of hackers are motivated by political, religious, environmental, or other personal beliefs. They are usually content with embarrassing their opponents or defacing their websites, although they

can slip into corporate-espionage mode if it means they can weaken the opponent. Think WikiLeaks.

6. **Cyber warriors** Cyber warfare is a city-state against city-state exploitation with an endgame objective of disabling an opponent's military capability. Participants may operate as APT or corporate spies at times, but everything they learn is geared toward a specific military objective. The Stuxnet worm is a great example of this attack method.
7. **Rogue hackers** There are hundreds of thousands of hackers who simply want to prove their skills, brag to friends, and are thrilled to engage in unauthorized activities. They may participate in other types of hacking (crimeware), but it isn't their only objective and motivation. These are the traditional stereotyped figures popularized by the 1983 film "War Games," hacking late at night, while drinking Mountain Dew and eating Doritos. These are the petty criminals of the cyber world. They're a nuisance, but they aren't about to disrupt the Internet and business as we know it