NCS Secure and Defensive Coding Course
13th Mar 2015

GROUP ENTERPRISE

Lee **Chee Yong**
Lee Xing Wang **Andors**

ncs making IT happen

Course Agenda

Day 1 AM

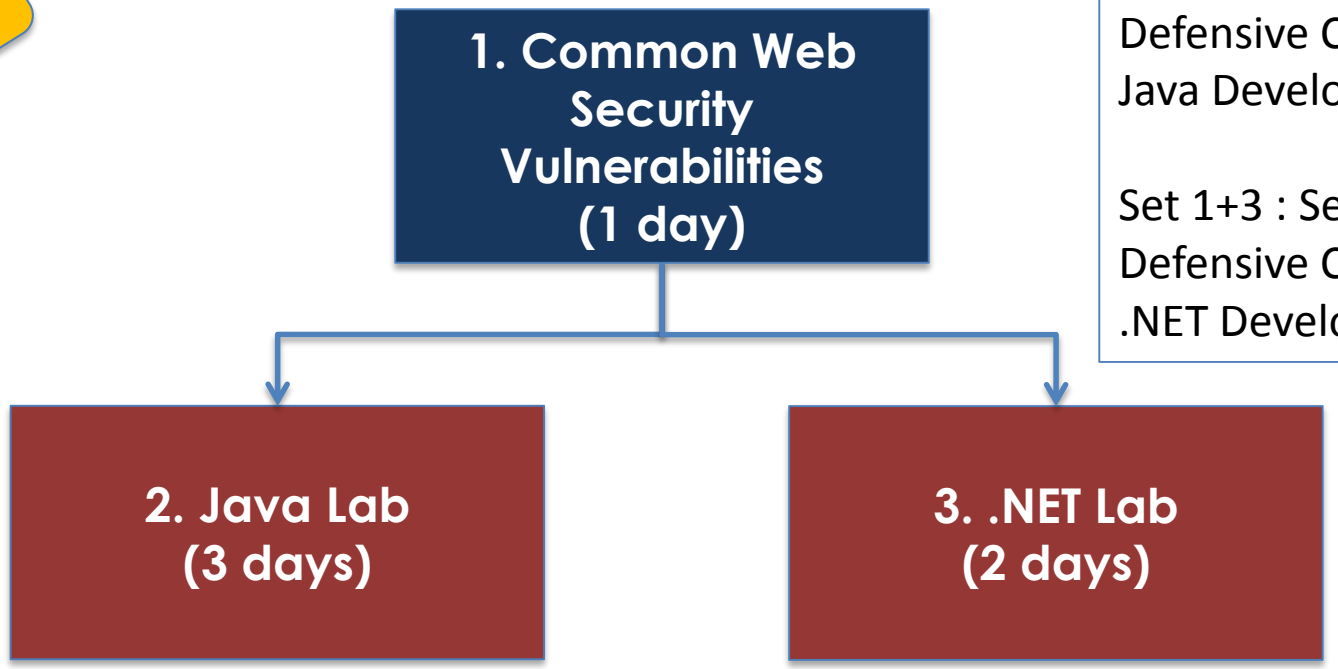
- Introduction
  - Headline Security Issues
  - IT Security Issues in Singapore
  - Security Tender Clauses
  - Common Attacks
- Techniques to Identify Sources of Attack
- Classes of Hackers
- Secure Development Lifecycle
- Secure Development Principles
- Password Policies

Day 1 PM

- Cryptography
  - Symmetric and Asymmetric Encryption
  - Digital Signature
  - One Way Hash
  - Transport Level Encryption
  - Stenography & Watermarking
- Wifi Security
- Regular Expression
- Validation Strategies
- Static Code Analyzers
- NCS Code Quality Gates and Scan Statistics

Defensive Programming Course Design

Mandatory Training



Set 1 + 2 : Secure and Defensive Coding for Java Developers Course.

Set 1+3 : Secure and Defensive Coding for .NET Developers Course

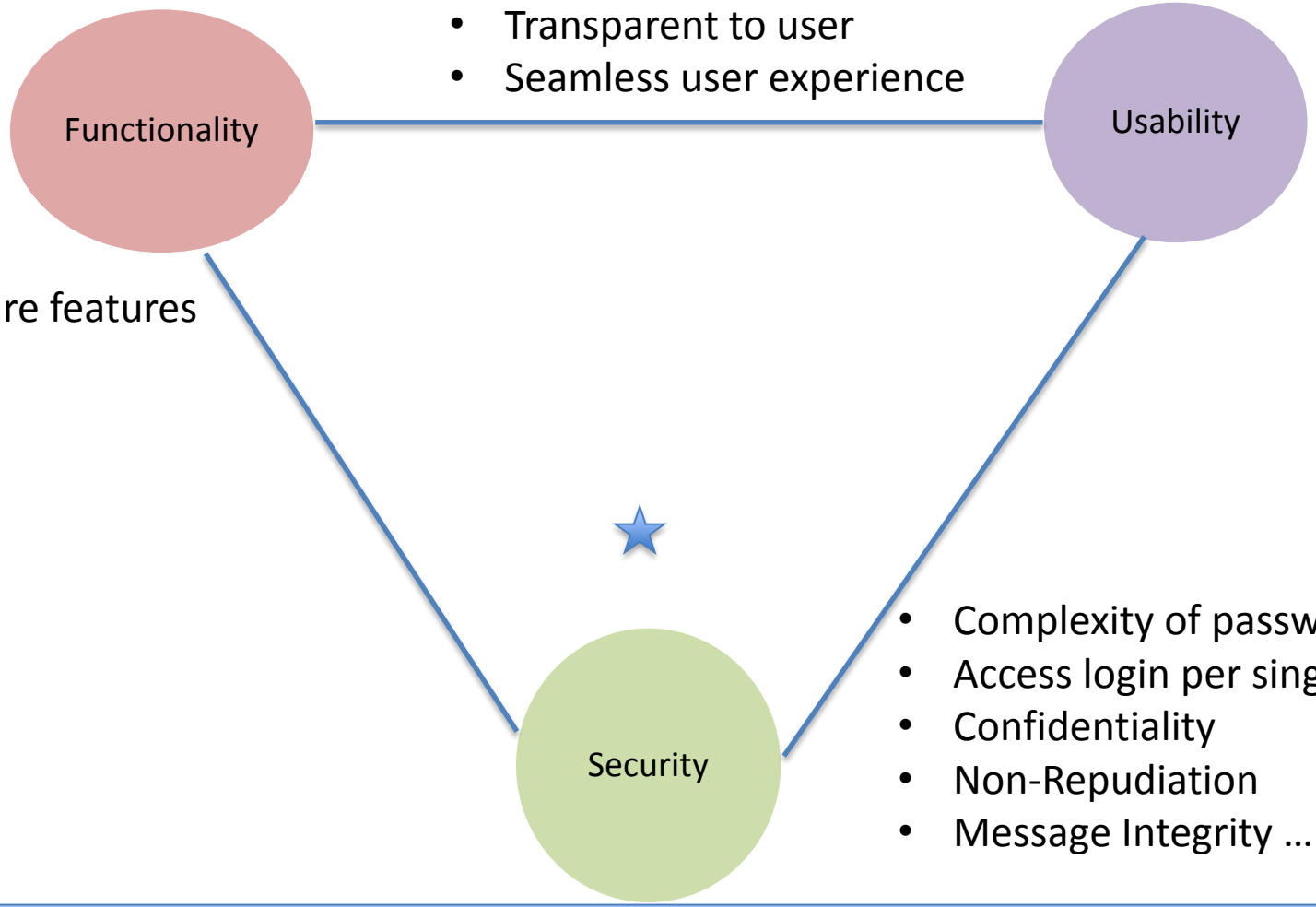
Course Administrative Notes

- Course Timing:
  - 9:00 am to 5:00 pm
- Lunch-break
  - 1 pm to 13:15pm
  - Room will be locked during lunchtime
- Place name tags in front for easy Identification
- Check emails at designated break times
- Phones on Silent Mode. Step out of classroom to take urgent phone calls.
- Course sign-in are carried out at every half-day basis.


GROUP
ENTERPRISE
Introduction to Secure and Defensive Coding

Usability Triangle

- Easy to use
- Do not affect accessibility
- Transparent to user
- Seamless user experience



- Key software features

- Complexity of passwords
- Access login per single entry
- Confidentiality
- Non-Repudiation
- Message Integrity …..

Need to balance Psychological Acceptability with security requirements!

Attacker Versus Defender Viewpoints

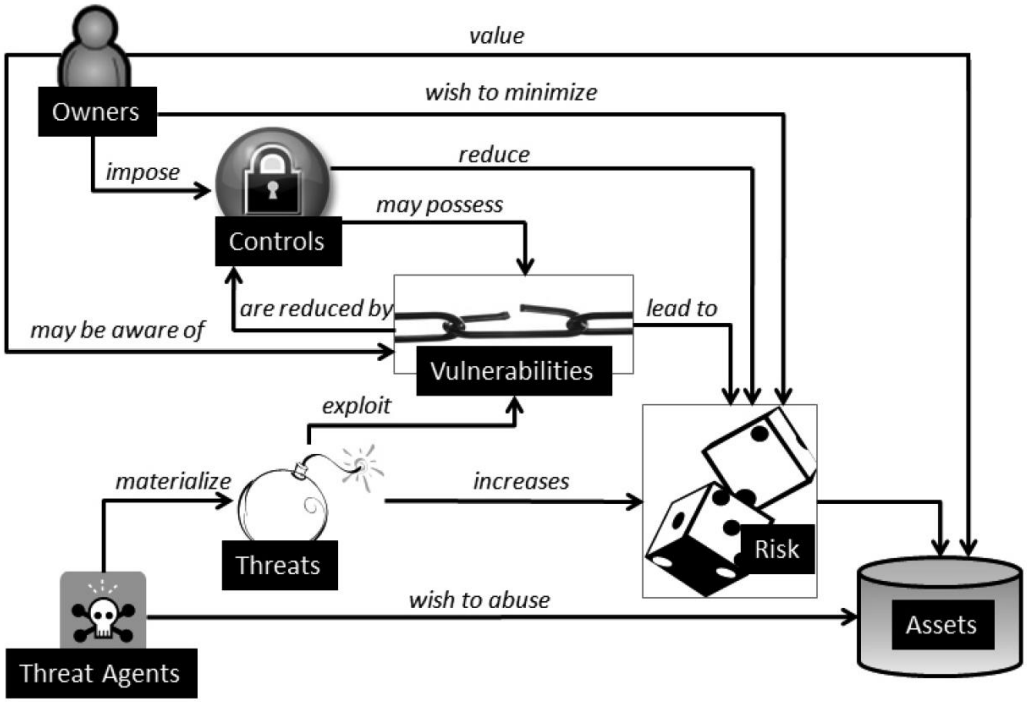
- To be able to design and develop secure code, understanding of the following three items are important.

Attackers' Perspective

Secure Coding Principles

Defenders' Perspective

Relationship of Key Security Terms



Asset	Threat	Vulnerability	Risk	Exploit
• An **Asset** is what we are trying to protect.	• A **Threat** is what we are trying to protect against	• A **Vulnerability** is a weakness or gap in our protection effort	• **Risk** is the potential loss, damage or destruction as a result of threat exploiting a vulnerability	• An **Exploit** is a means of taking advantage of the vulnerability.

GROUP
ENTERPRISE

Headline Security Issues

The Heartbleed Bug

CVE-2014-0160

The Heartbleed Bug is a serious vulnerability in the popular OpenSSL cryptographic software library. This weakness allows stealing the information protected, under normal conditions, by the SSL/TLS encryption used to secure the Internet. SSL/TLS provides communication security and privacy over the Internet for applications such as web, email, instant messaging (IM) and some virtual private networks (VPNs).

The Heartbleed bug allows anyone on the Internet to read the memory of the systems protected by the vulnerable versions of the OpenSSL software. This compromises the secret keys used to identify the service providers and to encrypt the traffic, the names and passwords of the users and the actual content. This allows attackers to eavesdrop on communications, steal data directly from the services and users and to impersonate services and users.

What leaks in practice?

We have tested some of our own services from attacker's perspective. We attacked ourselves from outside, without leaving a trace. Without using any privileged information or credentials we were able steal from ourselves the secret keys used for our X.509 certificates, user names and passwords, instant messages, emails and business critical documents and communication.

How to stop the leak?

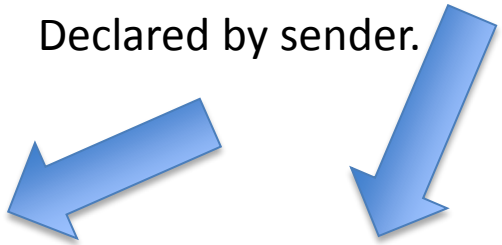
As long as the vulnerable version of OpenSSL is in use it can be abused. Fixed OpenSSL has been released and now it has to be deployed. Operating system vendors and distribution, appliance vendors, independent software vendors have to adopt the fix and notify their users. Service providers and users have to install the fix as it becomes available for the operating systems, networked appliances and software they use.

Heart Bleed Video

http://www.youtube.com/watch?v=8oI_laHhGjE

Since the Heartbleed bug has existed for two years, it raises obvious questions about whether the NSA or other spy agencies were exploiting it before its discovery.

Data Structure of a Heart Beat




```
struct {
    HeartbeatMessageType type;
    uint16 payload_length;
    opaque payload[HeartbeatMessage.payload_length];
    opaque padding[padding_length];

} HeartbeatMessage;
```

```
/* Read type and payload length first */
hbtype = *p++;
n2s(p, payload);
pl = p;
```

1. The first byte of the record is the heartbeat type.
2. The macro *n2s* takes two bytes from p, and puts them in payload. This is actually the *length* of the payload. Note that the actual length in the record is not checked.
3. The variable *pl* is then the resulting heartbeat data, supplied by the requester.

Buffer Overflow cause for Heartbleed Bug

```
/* Allocate memory for the response, size is 1 byte
 * message type, plus 2 bytes payload length, plus
 * payload, plus padding
 */
 buffer = OPENSSL_malloc(1 + 2 + payload + padding);
 bp = buffer;

 /* Enter response type, length and copy payload */
*bp++ = TLS1_HB_RESPONSE;
 s2n(payload, bp);
 memcpy(bp, pl, payload);
 bp += payload;
 /* Random padding */
 RAND_pseudo_bytes(bp, padding);

 r = dtls1_write_bytes(s, TLS1_RT_HEARTBEAT, buffer, 3 + payload + padding);
```

1. Heartbeat replies contain payload data, as a way to verify encrypted circuit is still working both ways.
2. Can send a small heartbeat request, but sneakily set your payload length field to 0xFFFF (65535 bytes).
3. Then, OpenSSL will still copy 65535 bytes, even though sender do not send across that many bytes.
4. That means OpenSSL runs off the end of your data and scoops up whatever else is next to it in memory at the other end of the connection, 64KB each time you send a malformed heartbeat request.
5. According to the Finnish National Cyber Security Centre, the sort of data that "bleeds" when the bug is triggered varies enormously.

The fix

```
/* Read type and payload length first */
hbtype = *p++;
n2s(p, payload);
pl = p;
```

```
/* Read type and payload length first */
if (1 + 2 + 16 > s->s3->rrec.length)
    return 0; /* silently discard */
hbtype = *p++;
n2s(p, payload);
if (1 + 2 + payload + 16 > s->s3->rrec.length)
    return 0; /* silently discard per RFC 6520 sec. 4 */
pl = p;
```

Simple boundary check. Morale of the story => Do not trust your client.

Some Affected Heart Bleed Products

Oracle

Patch Availability Matrix	
Affected Products	**Patch Availability**
MySQL Connector/C 6.1.0-6.1.3 [Product ID 8576/CONC]	MOS Note 1663909.1
MySQL Connector/ODBC 5.1.13, 5.2.5-5.2.6, 5.3.2 [Product ID 8576/CONODBC]	MOS Note 1663909.1
MySQL Enterprise Backup 3.10.0 [Product ID 4629]	MOS Note 1663909.1
MySQL Enterprise Monitor 2.3.13-2.3.15, 3.0.0-3.0.8 [Product ID 8480]	MOS Note 1663909.1
MySQL Enterprise Server 5.6.11-5.6.17 [Product ID 8476]	MOS Note 1663909.1
MySQL Workbench 6.1.4 and earlier [Product ID 4627]	MOS Note 1663909.1
Oracle Big Data Appliance (includes Oracle Linux 6) [Product ID 9734]	MOS Note 1662966.1
Oracle Communications Internet Name and Address Management [Product ID 2262]	MOS Note 1665972.1
Oracle Communications Application Session Controller 3.7.0.m1p0, 3.7.0.m2p0 [Product ID 10769]	MOS Note 1664964.1
Oracle Communications Interactive Session Recorder 4.0.0 and later [Product ID 10765]	MOS Note 1664216.1
Oracle Communications Network Charging and Control 5.0.1 [Product ID 4623]	MOS Note 1664010.1
Oracle Communications Session Monitor Suite 3.3.40, 3.3.50 [Product ID 10761]	MOS Note 1664883.1
Oracle Communications WebRTC Session Controller 7.0.1 [Product ID 10811]	MOS Note 1664964.1
Oracle Endeca Information Discovery Studio (using Tomcat on Windows) [Product ID 9634]	Only customers who use Tomcat and have enabled the APR/Native interface may be vulnerable. MOS Note 1666812.1
Oracle Explorer [Product ID 1330/EXPLORER]	MOS Note 1664793.1
Oracle Linux 6 [Product ID 1309]	MOS Note 1663998.1
Oracle Mobile Security Suite [Product ID 10913]	MOS Note 1664164.1
Oracle Virtual Compute Appliance Software [Product ID 10635]	MOS Note 1664138.1
Primavera P6 Professional Project Management (includes Primavera P6 Enterprise Project Portfolio Management) [Product ID 5579, 5580]	MOS Note 1664871.1 (P6 PPM) and MOS Note 1662799.1 (P6 EPPM) and MOS Note 1665370.1

IBM

- IBM Tivoli Composite Application Manager for Transactions
- IBM Tivoli Endpoint Manager for Remote Control
- IBM Tivoli Endpoint Manager for Remote Control 9.1.0, 9.0.1, 9.0.0
- IBM Tivoli Management Framework 4.1.1 (linux-ix86 and linux-s390)
- IBM Tivoli Netcool/Portal
- IBM Tivoli Netcool/Reporter
- IBM Tivoli Netcool SSM 4.0.0 FP1 - P14 and Interim Fix 14-01
- IBM Tivoli Netcool SSM 4.0.1 FP1 and Earlier
- IBM Tivoli Remote Control version 5.1.2
- IBM Tivoli Storage Productivity Center 5.2.0-5.2.1.0, 5.1.0-5.1.1.3, 4.2.2.143 (FP3)-4.2.2.177
- IBM ToolsCenter
- IBM Upward Integration Modules (UIM)
- IBM WebSphere Cast Iron
- IBM WebSphere Hypervisor Edition v8.5.5.1 on zLinux for SmartCloud Orchestrator
- IBM WebSphere MQ (Paho MQTT and HP-NSS clients)
- IBM Worklight 6.1.0.0 and 6.1.0.1
- IBM Worklight Consumer Edition 6.1.0.0 and 6.1.0 Fix Pack 1
- IBM Worklight Enterprise Edition 6.1.0.0 and 6.1.0 Fix Pack 1

....

....

....

Affected Operating Systems

- Debian Wheezy (stable), OpenSSL 1.0.1e-2+deb7u4
- Ubuntu 12.04.4 LTS, OpenSSL 1.0.1-4ubuntu5.11
- CentOS 6.5, OpenSSL 1.0.1e-15
- Fedora 18, OpenSSL 1.0.1e-4
- OpenBSD 5.3 (OpenSSL 1.0.1c 10 May 2012) and 5.4 (OpenSSL 1.0.1c 10 May 2012)
- FreeBSD 10.0 - OpenSSL 1.0.1e 11 Feb 2013
- NetBSD 5.0.2 (OpenSSL 1.0.1e)
- OpenSUSE 12.2 (OpenSSL 1.0.1c)

How to Patch OpenSLL Vulnerability Issues?

1. Determine whether your machines are impacted.
   - Go to http://www.cvedetails.com/ to check on whether the flaw is present in your machines OSes and application servers.
2. Install Patched Version of OpenSSL. SSL/TLS private keys might have been compromised, which allows for hackers to listen in to the encrypted SSL traffic.
3. Obtain new SSL certificates and install on web servers.

High Publicity Sony Pictures Entertainment Hack

- Confidential Data belonging to Sony Pictures are revealed on Nov 24th 2014.
  - Personal Information about Sony Employees, Families.
    - Social Security Number
  - Emails between employees
    - Emails of Pascal and Scott Rudin
    - Insider News of Movie Stars Acting Capabilities
      - Angelina Jolie as a “minimally talented spoiled brat”.
- Demanded cancellation of file “The Interview”.
  - Threaten terrorist actions against New York City Sunshine Cinema

  Comedy of plot to assassinate North Korean Leader Kim Jong-un.
- Hacked by Guardians of Peace (GOP).
  - Any association with North Korean?

Source:
http://deadline.com/2014/12/sony-hack-timeline-any-pascal-the-interview-north-korea-1201325501/

You can be Hacked by Government

- Many of the Wiki Leaks revealed by Edward Snowden are suggesting that US government and many other governments are listening in to private citizen information, requesting protected information from technology companies and creating back-doors for real-time access.

GROUP
ENTERPRISE

IT Security Issues in Singapore

2012: the year of hacktivism

Arab Spring
Political freedom

Foxcon
Working conditions

Justice Department
Anti-corruption

Vatican
Unhealthy transmitters

UN ITU
Internet deep packet inspection

Knowledge Is Free.
We Are Anonymous.
We Are Legion.

We Do Not Forgive.
We Do Not Forget.
Expect Us.

Trigger Points of the Messiah Attack

On 1st June 2013, new set of censorship regulations by MDA
websites with at least 50,000 unique visitors from Singapore every month that publish at least one local news article per week over a period of two months ... will have to remove 'prohibited content' such as articles that undermine 'racial or religious harmony' within 24 hours of being notified by Singapore's media regulator.

Seen **as impinging on Internet Freedom** by the Anonymous Group.
Declared war on Singapore PAP government.

http://www.youtube.com/watch?v=CwEyB42swMU&feature=youtube_gdata_player

Recent Incidents on Straits Times Website

From Around The World
Life In Review
STs Sports Arena

Dear ST: You just got hacked for misleading the people!

"ST is tempting fate" - newstation.sg Greetings Irene Tham & Straitstimes.com, I am The Messiah from the Anonymous Collective. We are a decentralized non-violent resistance movement, which seeks to restore the rule of law and fight back against the organized criminal class. We oppose any form of internet censorship among other things. Allow [...]

Email
Print
1
Tweet

By Irene Tham
Correspondent
itham@sph.com.sg

+1 0

Published on November 1st, 2013

"ST is tempting fate" - newstation.sg

« ST Blogs
« Most Commented
« ST's Home Ground
« From Around The World
« On The Money
« Life in Review
« STs Sports Arena
« Through The Lens

ST Blogs

ALSO BY IRENE THAM

The power of group buying
Dimwits on Facebook?
Buying music at a kiosk?
More legroom for lappie users
More than just technology

Popular Tags

Recent Security Attacks on PM's website

Singapore PM's website hacked by Anonymous

AFP News – Fri, Nov 8, 2013

Recommended for you

Activist hacker group Anonymous attacked the government website of the Singapore …

Singapore Prime Minister Lee Hsien Loong's official website was hacked Thursday by apparent members of activist group Anonymous after he vowed to hunt down anyone who attacks the city-state's technological network.

"It's great to be Singaporean today," read a mocking headline in a section of www.pmo.gov.sg next to the group's trademark Guy Fawkes mask, a symbol of anti-establishment defiance worldwide.

Next to it was another image saying: "PM Lee warns hackers: We will track you down -- even if you think you're 'anonymous'".

Another message read: "ANONYMOUS SG WAS HERE BIATCH", using a pejorative in online youth slang.

The defaced section was quickly taken offline after the hacking incident surfaced in a posting on Facebook. The rest of the site was working normally.

In a statement issued early Friday, the government's Infocomm Development Authority said it was investigating the incident.

"The PMO (Prime Minister's Office) main website is still working, and we are working to restore the page that has been compromised," it said.

Behind the Scene for PMO Attack

Nov 12, 2014

Published on Nov 12, 2014

Man admits hacking PMO website

IAN POH

A JOBLESS man has admitted to entering unauthorised computer code into the Prime Minister's Office (PMO) website server on Nov 7 last year, among other offences.

A district court heard yesterday that Mohammad Azhar Tahir, 27, used a Google search bar embedded on the page to create a modified version that referenced international hacktivist group Anonymous.

He did this by entering HTML code instead of proper search terms.

Internet users who clicked a link containing the script, which he posted on various social media websites, would see a Guy Fawkes mask and two messages instead of what the search-results page would normally display.

The messages were: "It's great to be Singaporean today" and "ANONYMOUS SG WAS HERE B*****".

Mohammad Azhar pleaded guilty yesterday to seven of 59 charges under the Computer Misuse and Cybersecurity Act.

These included accessing a neighbour's wireless Internet service and modifying social media and e-mail accounts that belonged to Ah Boys To Men actor Muhammad Ridhwan Azman, 21.

Deputy Public Prosecutor April Phang told the court that there had been widespread concern because it seemed as if the PMO website had been defaced by hackers from Anonymous, although data on its server had not actually been altered.

Cross Site Scripting

Tapping on Others Network

Mohammad Azhar had effectively masked his identity by using his neighbour's Internet connection, said DPP Phang.

In two instances, he had used the actor's accounts to publicise the link containing the offending script.

The court heard that Mohammad Azhar had committed the PMO website intrusions with a technique known as cross-site scripting.

He decided to test his skills on the page's search function after hearing of Prime Minister Lee Hsien Loong's warning that hackers who attack the country's computer networks would be brought to justice.

Mr Lee had issued the warning about a week after a person claiming to be from Anonymous issued a threat via a YouTube video to attack Singapore's infrastructure in protest against new licensing rules for news websites.

Similar charges had also been brought against Mohammad Azhar's 22-year-old brother, full-time national serviceman Mohammad Asyiq Tahir.

Mohammad Asyiq pleaded guilty yesterday to five of the 11 counts, which largely involved tampering with the actor's e-mail and social media accounts. None of the offences concerned the PMO website server.

Both brothers are expected to be sentenced on Dec 8.

Mohammad Azhar faces up to three years in jail and up to $10,000 in fines for each of two counts of securing unauthorised access to the website server.

Source: http://mypaper.sg/top-stories/man-admits-hacking-pmo-website-20141112

Recent Security Attacks

Script Kiddie Scare???

The man alleged to have hacked Singapore sites under the moniker "The Messiah" has been arrested by police, according to newsdaily Straits Times.

Singapore police arrested James Raj Arokiasamy, 35, who was charged in court on Tuesday for hacking the site of Ang Mo Kio Town Council.

Raj is accused of hacking the town council's site from a unit in Dorchester Apartment at Jalan Sri Hartamas in Kuala Lumpur, Malaysia, on October 28 at about 1.35pm, according to Channel NewsAsia. He had allegedly defaced the site with an image of a Guy Fawkes mask and a message signing off as "The Messiah".

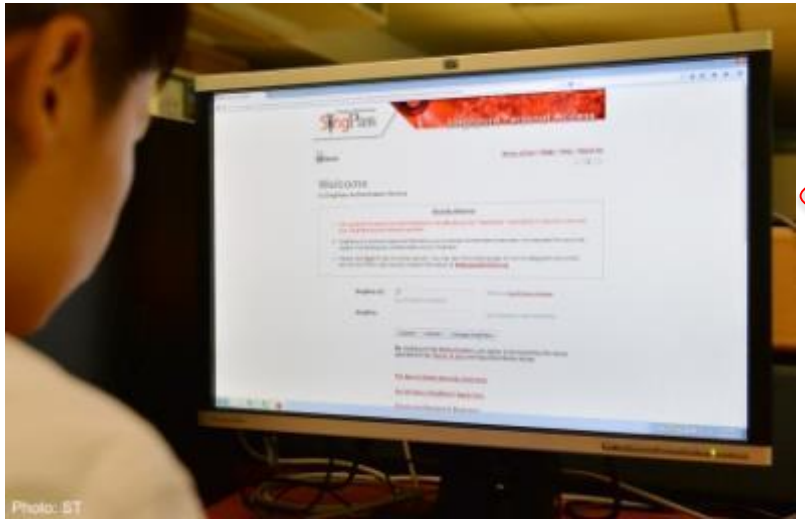
He is also suspected of hacking the sites belonging to The Straits Times, People's Action Party Community Foundation and City Harvest Church's co-founder and musician Sun Ho, according to a police statement.

He faces up to three years in jail and a S$10,000 (US$8,011) fine for one charge under the Computer Misuse and Cybersecurity Act. Separately, Raj also has been accused of three prior drug charges, which could add up to 10 years in jail and a S$20,000 (US$16,022) fine on each count.

How PMO site was "hacked"

According to Trend Micro, the attack was a result of typical Cross Site Scripting (XSS) where the cybercriminal exploited the "search" function on the site, and injected

SingPass Accounts Accessed Illegitimately



Source: http://news.asiaone.com/news/digital1/more-1500-singpass-accounts-accessed-illegitimately-ida

Why is installing anti-virus software important?

Here is the statement from IDA:

On Monday, June 2, 2014, IDA was notified by the SingPass operator that a number of SingPass users had received a SingPass Password Reset Notification Letter even though they did not request for any password reset.

IDA's preliminary investigations revealed that 1,560 users' IDs and passwords had potentially been accessed without the users' permission.

An anomaly was detected between the number of mobile numbers used for Immediate Reset One-Time Passwords and the number of SingPass accounts that they were tied to.

Of these 1,560 users, 419 passwords were also reset triggering the SingPass Password Reset Notification Letters to be sent to the registered address of the actual account holder.

A police report was lodged on June 3, 2014 and the matter is currently under investigation.

Based on IDA's checks, there is no evidence to suggest that the SingPass system has been compromised.

The passwords of all affected users have been reset and we are in the process of notifying them of this incident.

"For every individual, this incident underlines the importance of taking personal responsibility for cyber security. The Government strongly urges all SingPass users to take the necessary precautions to enhance their cyber security. They should ensure that they use strong passwords to access not only SingPass but all the other e-Services they subscribe to. Strong passwords contain a combination of numerical figures, capital letters and are at least eight characters long. Users should also install anti-virus software and update all their software regularly," said Ms Jacqueline Poh, Managing Director of the Infocomm Development Authority of Singapore.

The Singapore Government takes cyber security very seriously. The protection of personal data and the delivery of secure e-Services are critical. We will continue to strengthen all Government e-Services as part of on-going efforts to enhance security. Users can visit the GoSafe Online website at www.gosafeonline.sg to learn more about how to protect themselves against cyber threats or seek assistance.

Hacked SingPass Accounts Used to Apply Workpass

Source:
http://therealsingapore.com/content/ida-hacked-singpass-accounts-were-used-apply-work-passes

Three breached SingPass accounts used for fraudulent work pass applications

Nurdianah Md Nur | July 7, 2014

ShareThis

Three of the 1,560 SingPass accounts compromised last month were used to apply for six work passes, according to a joint statement by the Ministry of Manpower (MOM) and the Infocomm Development Authority (IDA) on 4 July 2014.

Upon discovery of the fraudulent applications, MOM cancelled the work passes and reported the matter to the police. The ministry has also implemented "additional measures to strengthen and further safeguard its work pass transactions" but did not elaborate.

SingPass provides residents with access to e-government services including personal income tax filings and statements of their retirement savings, known as the Central Provident Fund (CPF).

Just last month, IDA announced that 1,560 SingPass accounts were compromised, of which 419 accounts had their passwords reset without the users' permission.

In response to that incident, IDA said in a statement that it is currently enhancing the SingPass system, which will be ready by the third quarter of 2015. It will also introduce further security measures such as enabling users to set their own usernames in the enhanced SingPass system and using second-factor authentication (2FA) for e-government transactions involving sensitive data.

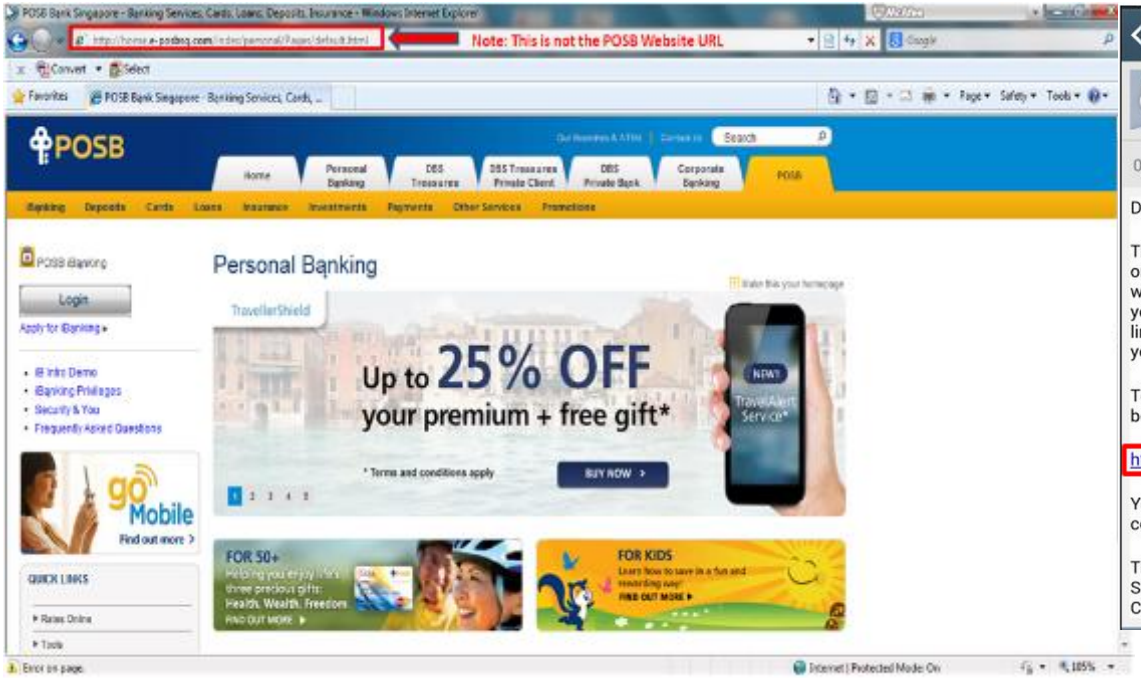
In the meantime, IDA and MOM encourage users to strengthen their SingPass passwords by ensuring that they are alphanumeric and contain at least eight characters.

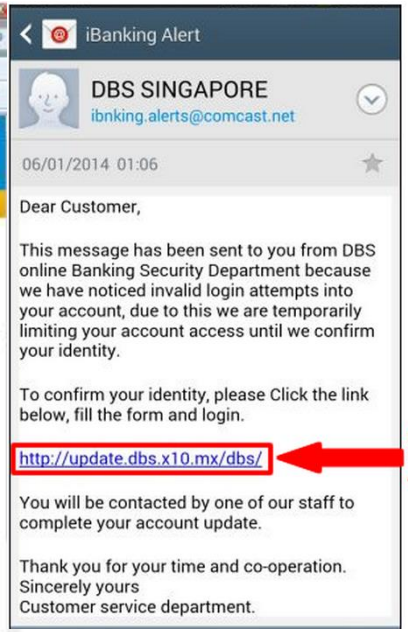
Introduce Two Factor Authentication to Singpass?

Have Your Application Design Catered for Easy Change to Two Factor Authentication?

POSB Phishing Site

https://income.posb-bank.com/personal/





Note: This is not the DBS Internet Banking URL.
Never click links from emails.
Always type the URL into the browser diretly.

https://inspirasidunia.com/DBS/

From: DBS BANK LTD [mailto:customerservice@dbs.com.sg]
Sent: Tuesday, 13 August, 2013 9:07 AM
To:
Subject: DBS INTERNET BANKING SECURITY UPDATE.

Dear valued customer,

DBS INTERNET BANKING SECURITY UPDATE.

Since March 2013, we introduced a new security system. This new system ensures that there can be no abuse on your DBS iBANKING. To ensure that your account is protected by our new security system, we recommend that you click on the link below and enter your information on check. Once you have done this, your account will be updated with the new security software.

http://www.salonworldknoxfield.com/osc/Update/index.php

Note: This is not the DBS Internet Banking URL.
Never click on links from emails.
Always type in the URL into the browser directly.

CAUTION: After filling out the information requested, you will be contacted/called by one of our staff to be able to complete the software installation.

DBS, The Safest Bank in Asia, your safety and protection is obliged. Thank you for your time and cooperation.

Sincerely yours
Customer service department
HEAD OFFICE ADDRESS:- (12 Marina Boulevard DBS Asia Central @ Marina Bay Financial Centre Tower 3
Singapore 018982)
DBS, Asia's Safest, Asia's Best
Safest Bank in Asia 2009-2013, Global Finance
Bank of the year Asia 2012, The Banker
Best Managed Bank in Asia Pacific 2013,
The Asian Banker.

Fake Manpower Ministry Website

SINGAPORE - The Ministry of Manpower (MOM) has lodged a police report over another fake website disguised as the ministry's, even carrying its logo.

The fake site - www.momgov-sg.com - is a near replica of Manpower Ministry's actual site, and it has since informed the police.

"Fake websites surface from time to time, and we'd like to remind members of the public to only use the official MOM website at www.mom.gov.sg for all informational and transactional needs concerning MOM matters," said the ministry in a Facebook post on Thursday afternoon.

Earlier this month on May 5, another fake MOM site - www.mom-govsg.com - was taken down following investigations. MOM assured then that access to the official MOM website remained unaffected and no data had been compromised.

Give up account/password information?

Important personal data being compromised?

Source: http://www.straitstimes.com/news/singapore/courts-crime/story/police-report-filed-over-fake-manpower-ministry-website-20140529

GROUP
ENTERPRISE

Code Security Tender Clauses

Common Security Clauses in Tender Specs (I)

- The Supplier shall ensure that the design and implementation of the System are conducted securely, and the System shall not be affected by the following list of common vulnerabilities.
  - (a) Parameters not validated causing SQL Injection, parameter manipulation
  - (b) Broken access control (malicious use of user IDs)
  - (c) Broken authentication and session management (use of account credentials and session cookies)
  - (d) Cross-Site Scripting (XSS) attacks
  - (e) Buffer Overflows
  - (g) Insecure storage
  - (h) Denial of Service
  - (i) Insecure configuration management

Need for a Secure Coding Practices, alignment to OWASP Top 10

Common Security Clauses in Tender Specs (II)

- Secure By Design – The system shall be analyzed, engineered and tested through the software development lifecycle rather than as an afterthought. Errors and exceptions shall be considered and handled gracefully by the system to maintain application and data integrity.
- The Supplier shall ensure that appropriate authentication process is built into the application to prevent access of sensitive content without proper authentication. The supplier shall propose the authentication process to the Government for review and approval prior to implementation. The proposed process shall minimally cover the following:
  - (i) Generic authentication responses for login errors
  - (ii) Use of multi-factor authentication in preventing Cross-Site Request Forgery (CSRF) attacks
- The supplier shall implement appropriate security measures to ensure that transport level security measures (for example SSLv2) are implemented.

Need for a Secure Design.

Common Security Clauses in Tender Specs (III)

- The supplier shall document how they would integrate essential IT security steps and activities for Secure System Development Process in the Project Management Plan.
- The contractor shall conduct software security tests, taking into consideration the overall security testing requirements imposed by the government. The contractor shall develop the software security test plan and submit to the authority for review as one of the deliverables in system analysis and design phase.
- The supplier shall perform a threat model analysis and assess the need to conduct software security testing to ensure that the software is implemented securely and is not subjected to any known vulnerabilities. Software security test shall cover all aspects of the software delivered, including custom code, components, products and system configuration.
- The contractor shall fix the security vulnerabilities discovered during the tests. If a solution cannot be delivered in the immediate rollout phases of the product, the contractor must provide documented mitigation procedures on handling the vulnerabilities, risk impact assessment and the expected timeline to make the correction.

Need for a Secure Software Development Lifecycle.

Common Security Clauses in Tender Specs (IV)

- All computers used shall be hardened in compliance with the requirements as specified by the Authority's security standards.
- The contractor shall ensure that the anti-malware solution is compatible with the proposed system, and there is no impact to the stability, functionality, safety and performance.
- The contractor shall put in DDOS mitigation measures such as clean pipe services or Content Delivery Network.

Need for System security and anti-virus measures.

Need for Network Security.

Liquidated Damages Clauses

- If the System's Confidentiality is compromised, for e.g. data is stolen or for any loss of data, 5% of the monthly maintenance contract price per incident, for the defaulting month.
- If the system's integrity is compromised, for e.g. unauthorized transaction made or web-site defaced, 5% of the monthly maintenance contract price, per incident, for the defaulting month.
- If the System's availability is compromised, for e.g. website unreachable due to denial of service attack, 5% of the monthly maintenance contract price, per incident, for the defaulting month.

Maintenance Fees Tied to Security Clauses.

Need infra protection against Denial of Service attacks.

Group Discussion – Part 1

- Form 5 groups, give your group a fun name.
- Elect a facilitator within the group to guide the discussion. Each group is expected to present on their views.

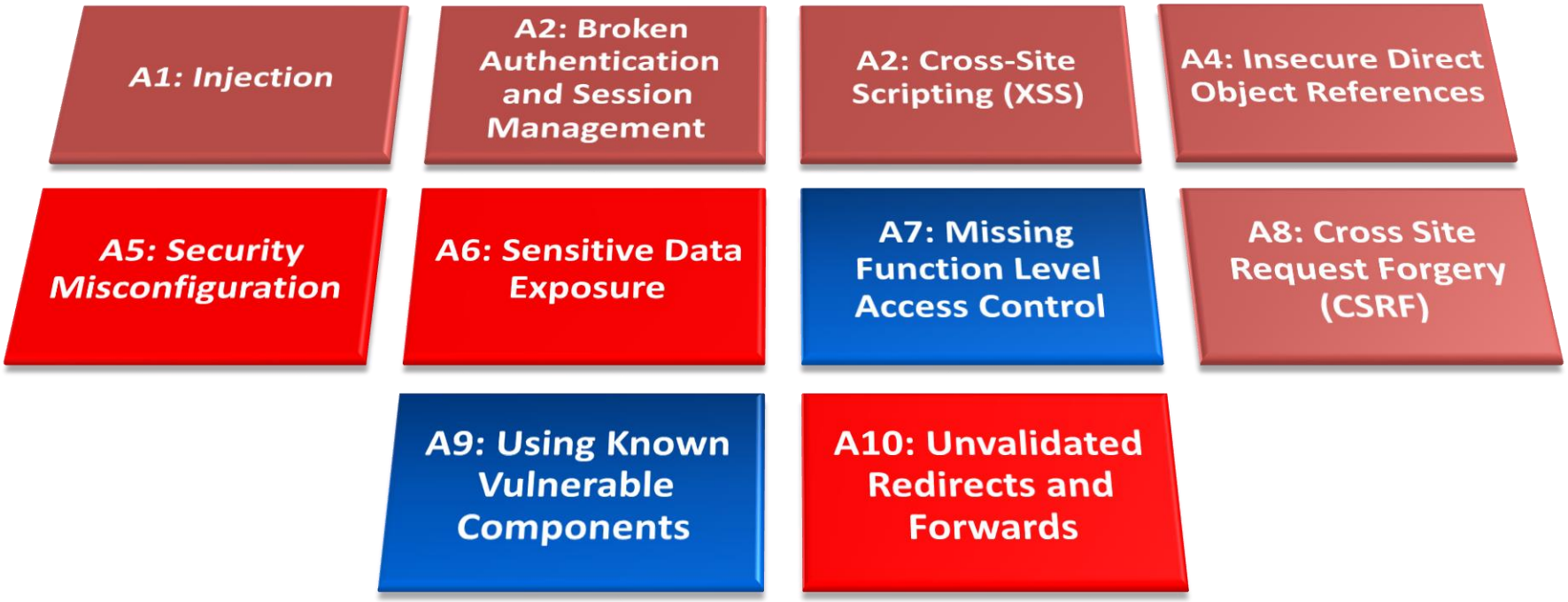
- Question 1:
  - (a) In view of these recent attacks, what are the steps NCS as a system integrator company should work on to ensure our developed systems do not become the next targeted victims?

- Question 2:
  - (a) What sort of sensitive information have your current or previous project handle?
  - (b) What are the type of security threats the current application you are working on most concerned with?
    - Virus, leakage of salary information, …?

GROUP
ENTERPRISE

Common Attacks

OWASP Top Ten (2013 Edition)

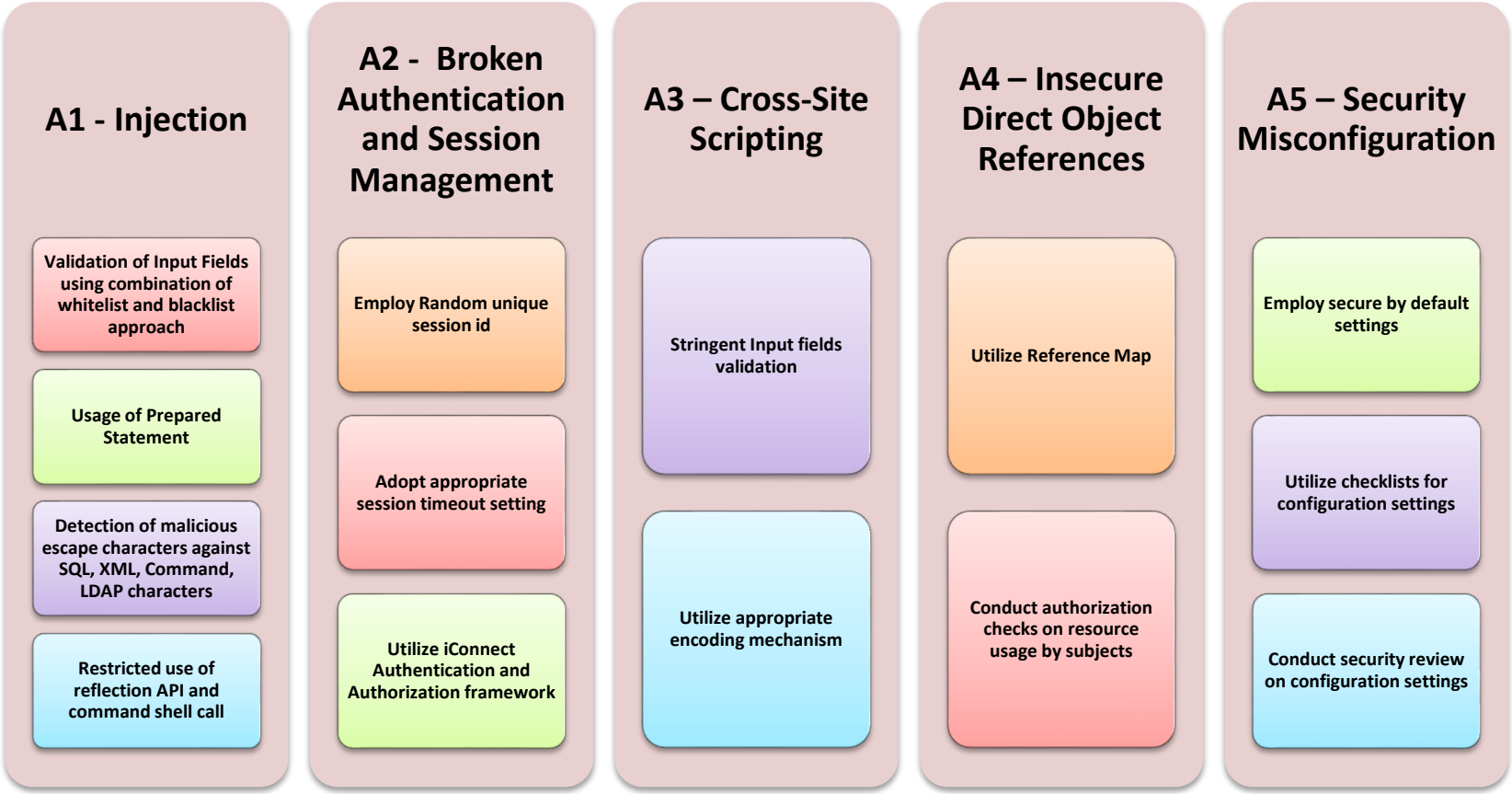


OWASP
The Open Web Application Security Project
http://www.owasp.org

https://www.owasp.org/images/1/17/OWASP_Top-10_2013--AppSec_EU_2013_-_Dave_Wichers.pdf

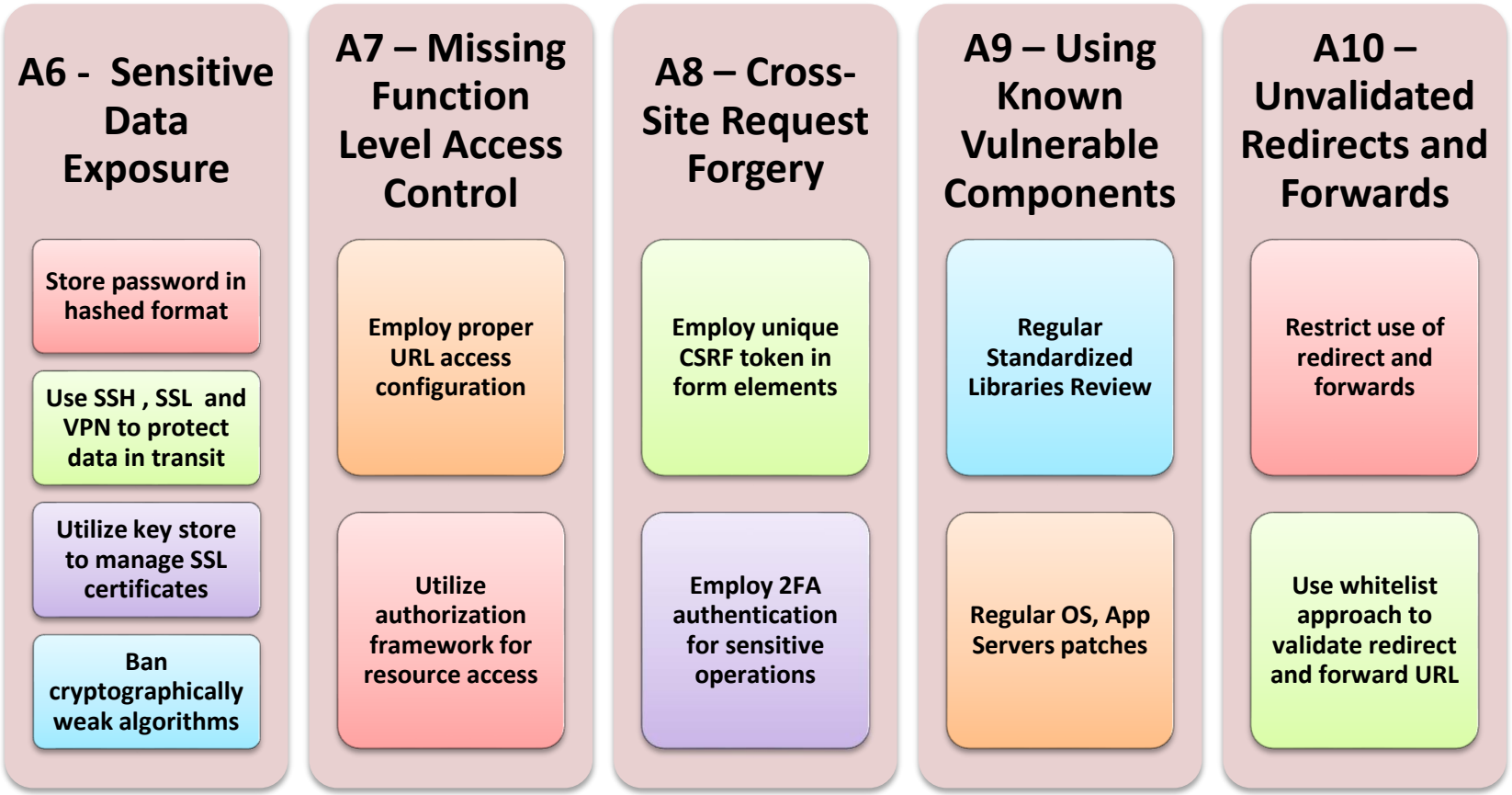
OWASP Top 10 - Secure and Defensive Measures (1)

These are the secure and defensive coding techniques that are employed by NCS teams to address OWASP Top 10 concerns.



OWASP Top 10 - Secure and Defensive Measures (2)

These are the secure and defensive coding techniques that are employed by NCS to address OWASP Top 10 concerns.



Buffer Overflow Attack

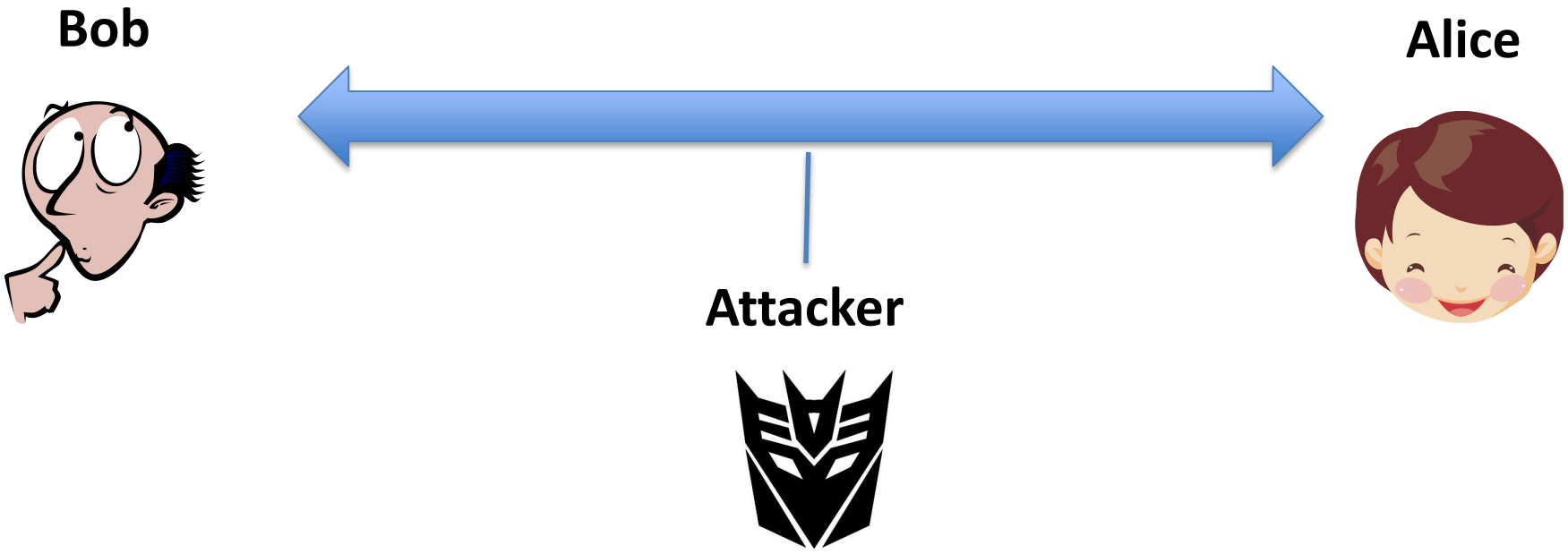
- A buffer overflow occurs when a program or process tries to store more data in a buffer than it is intended to hold. During writing, overruns buffer boundary and overwrite adjacent memory.
- Prevalent in non type-safe programming languages such as c.
- Buffer Overflow attack can result in service crashing, change data, disclose confidential information and even execution of code.

Dangerous c API that may cause buffer overflows

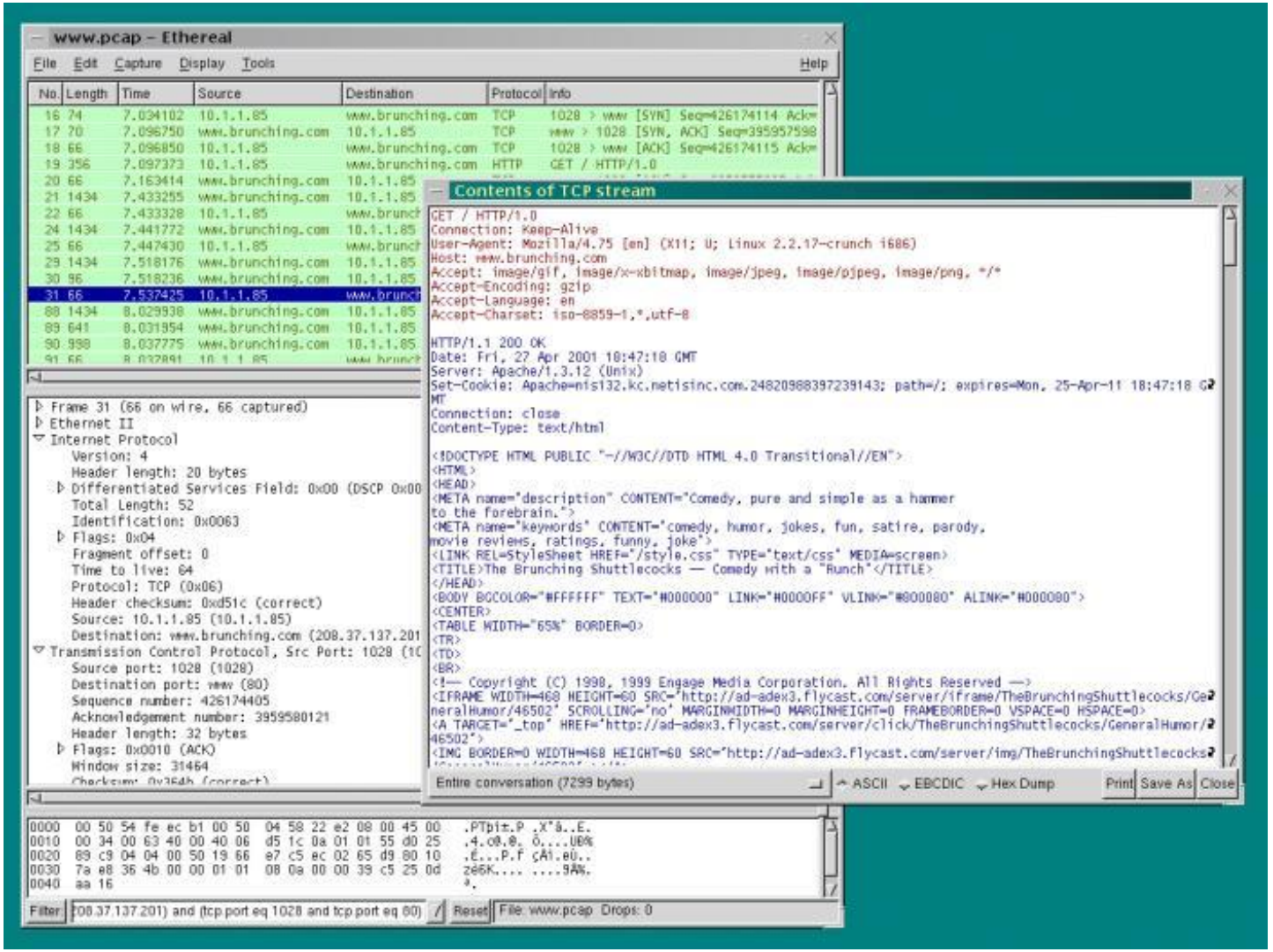
strcpy
strcat
getwd
gets
scanf
realpath
sprintf

Eavesdropping

- Eavesdropping is secretly listening to private conversation of others without their consent. Can be done over telephone lines, email, instant messaging.
- If the traffic is not encrypted, simple tools such as Wireshark or any packet sniffer is sufficient to do eavesdropping.

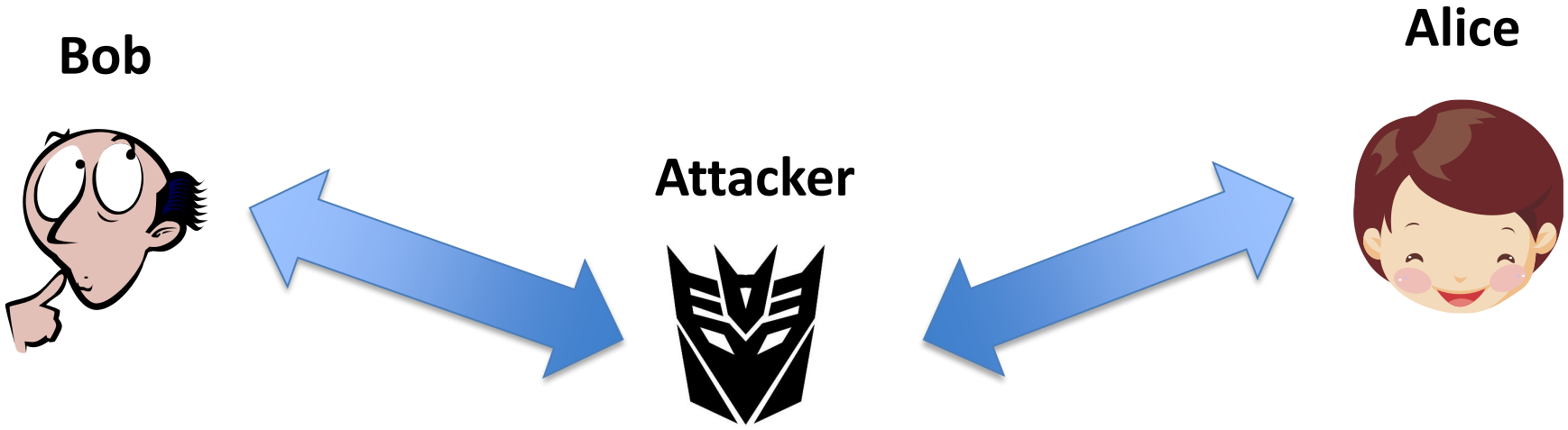


Eavesdropping - Wireshark / Ethereal in Action



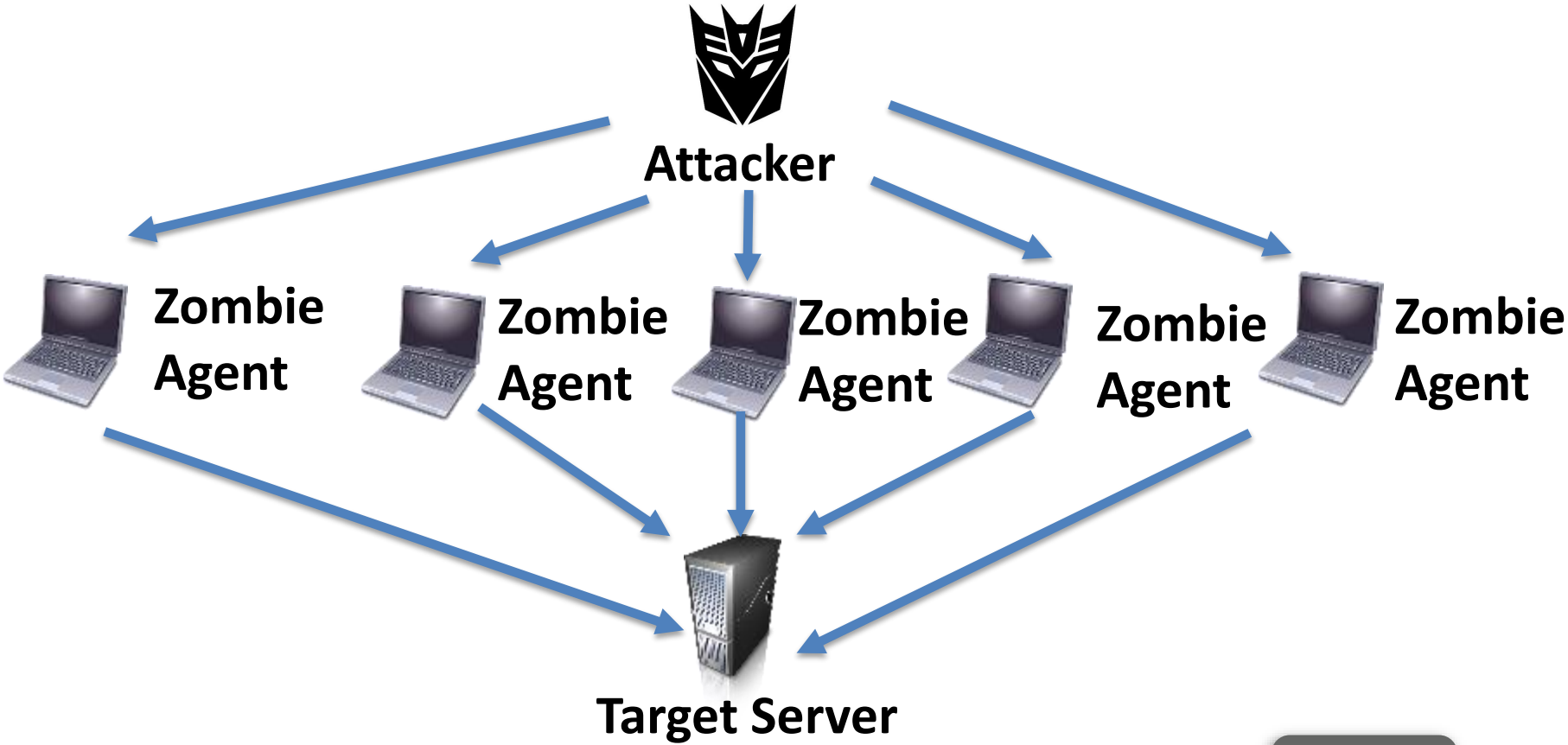
Man In the Middle Attack (MITM)

- MITM is a special form of **active eavesdropping**. Attacker makes independent connections with the victims and relays messages between them. The attacker is able to **intercept all messages** between the two victims.
- This can happen in
  - Compromised Wi-Fi wireless access points
  - Insecure SSL channel



Denial of Service (DOS)

- A denial of service (DOS) attack happens when a user or organization is deprived of services they would normally have.
- In distributed denial of service, large numbers of compromised systems attack a single target.



Network Level DDOS Prevention

- Cyber Watch Centre in Data Centre monitoring the traffic.
- CleanPipe Service
  - Uses a network-based DDoS mitigation service that performs macro-level (IP flow) analysis and real-time monitoring to address security threats round the clock.
    - Comparing signatures and examining different attributes of the traffic, including IP addresses, cookie variations, http headers, and Javascript footprints to detect DDoS.
  - Potential DDoS attack traffic is re-routed to fully operational Scrubbing Centers to be filtered and processed, leaving only legitimate traffic and transactions to be forwarded back to your business.
  - By providing protection at the core router level.
  - Internet Clean Pipe is able to deploy three types of anomaly detection (Misuse, Profile and Fingerprint) to provide comprehensive protection.
  - Offered by Singtel, Starhub and e-Cop.

Denial Of Service Example

MANY SINGAPORE GOVERNMENT WEBSITES 'DOWN'

Post date: 2 Nov 2013 - 4:03pm

Oops! Google Chrome could not find mom.gov.sg Try reloading: mom.gov.sg	Oops! Google Chrome could not find www.prisons.gov.sg Try reloading: www.prisons.gov.sg	Oops! Google Chrome could not find ica.gov.sg Try reloading: ica.gov.sg
Oops! Google Chrome could not find www.spf.gov.sg Try reloading: www.spf.gov.sg	Oops! Google Chrome could not con www.ava.gov.sg Try reloading: www.ava.gov.sg	Oops! Google Chrome could not find www.moh.gov.sg Try reloading: www.moh.gov.sg
Oops! Google Chrome could not find www.moe.gov.sg Try reloading: www.moe.gov.sg	Oops! Google Chrome could not find www.cpe.gov.sg Try reloading: www.cpe.gov.sg	Oops! Google Chrome could not find www.acra.gov.sg Try reloading: www.acra.gov.sg
Oops! Google Chrome could not find www.mha.gov.sg Try reloading: www.mha.gov.sg	Oops! Google Chrome could not find www.cnb.gov.sg Try reloading: www.cnb.gov.sg	Oops! Google Chrome could not find www.parliament.gov.sg Try reloading: www.parliament.gov.sg

Over the past few hours, most key government websites have been 'down' and users were unable to access them as of 3pm.

It is unknown who is behind the attacks but it is suspected that this is the work of the Anonymous Collective following earlier threats made in a "declaration of war" video.

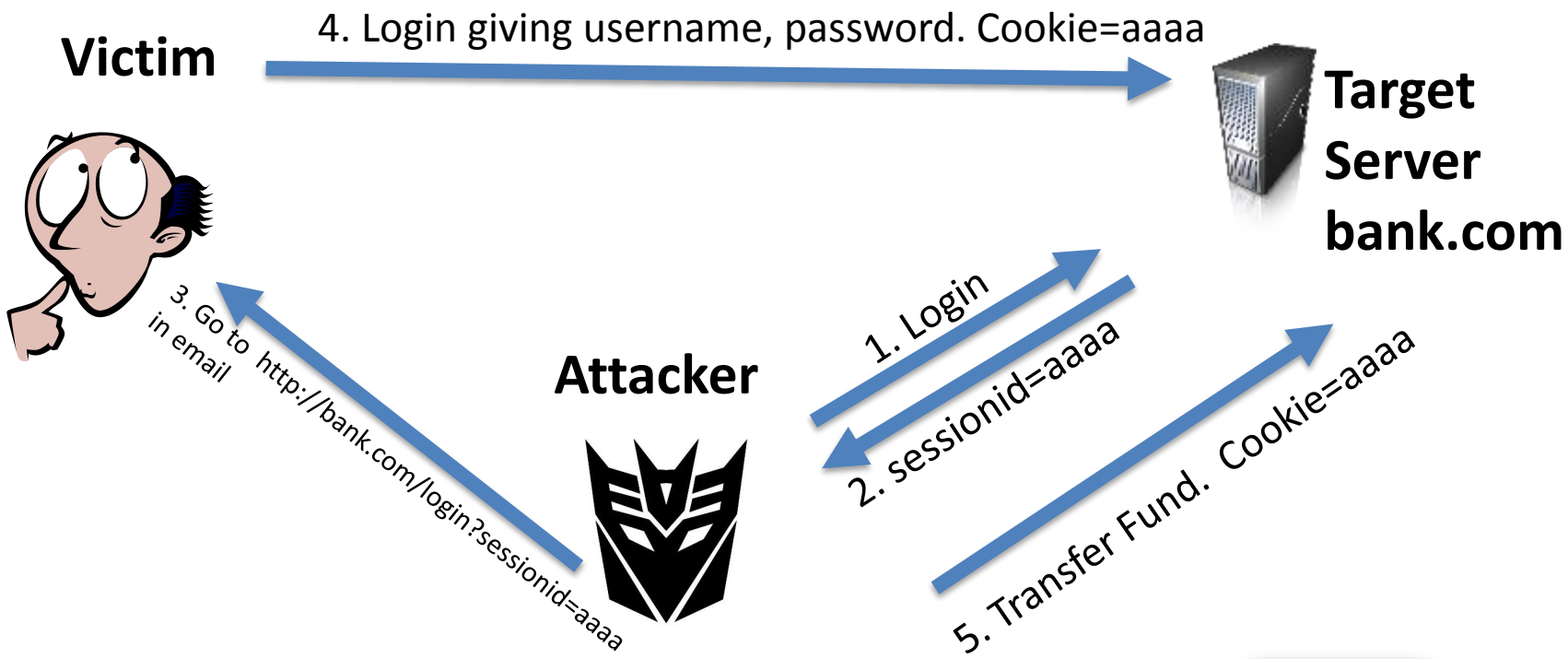
Many international news media sites such as BBC and Times of India have provided 'The Messiah' with the necessary coverage for more International Anonymous hackers to hear about the plans to take down the Singapore government websites. It is highly likely that it is a DDOS attack where many hackers collaboratively come together to constantly send requests and traffic to a certain website to max out its bandwith such that ordinary users are unable to access them.

Affected Sites:

Prisons.gov.sg
PMO.gov.sg
MSF.gov.sg
SEAB.gov.sg
SPF.gov.sg
AVA.gov.sg
LTA.gov.sg
ACRA.gov.sg
IDA.gov.sg
Singpass.gov.sg
SGDI.gov.sg
Careers.gov.sg
PUB.gov.sg
GOV.SG
MHA.gov.sg
CNB.gov.sg
ISD.gov.sg
ICA.gov.sg
NRF.gov.sg
CPIB.gov.sg
WDA.gov.sg
PSC.gov.sg
ISTANA.Gov.sg
SMC.gov.sg
SCDF.gov.sg
SLA.gov.sg
Supremecourt.gov.sg
NEA.gov.sg
BCA.gov.sg
CUSTOMS.gov.sg
SPRING.gov.sg
AGC.gov.sg

Session Fixation

- Session Fixation is an attack that permits attacker to hijack a valid user session.
- Exploits the way web application manages the session ID.
- Attackers fixes an established session, induce user to authenticate with the session ID, and then hijack the user-validated session using the fixed session ID.



GROUP
ENTERPRISE

Techniques to Identify Sources of Attack

Common Vulnerability Database

The National Vulnerability Database (NVD)

- U.S. government repository of vulnerabilities. Contains security checklists, security on software flaws, misconfigurations of products, impact metrics.
- https://nvd.nist.gov/

US Computer Emergency Response Team (CERT) Vulnerability Notes Database

- Looks at both developed and deployed software.
- Focused on vulnerability Discovery
- http://www.us-cert.gov/related-resources

US-CERT
UNITED STATES COMPUTER EMERGENCY READINESS TEAM

Common Vulnerabilities and Exposures

- Dictionary of common names for known security vulnerabilities
- Referred by many other databases
- CVE-yyyy-nnnn format
- http://cve.mitre.org/data/downloads/allitems.html

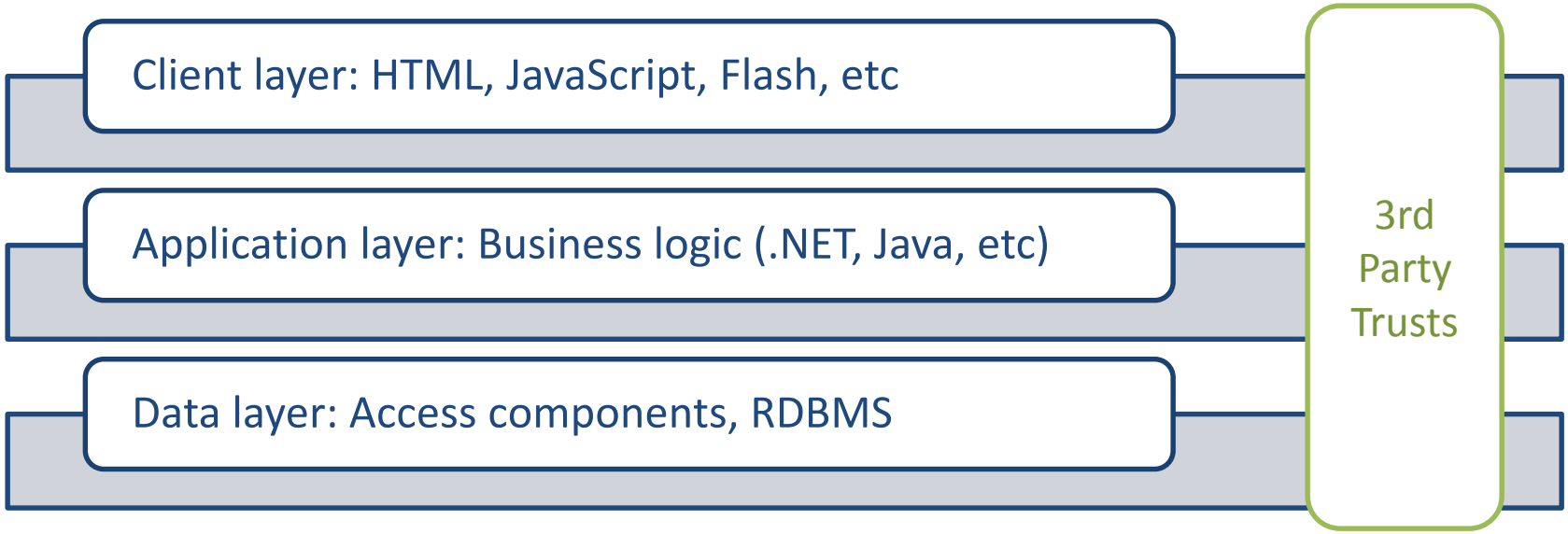
Open Source Vulnerability Database

- Independent and open source database created by and for the security community.
- http://osvdb.com/

OSVDB

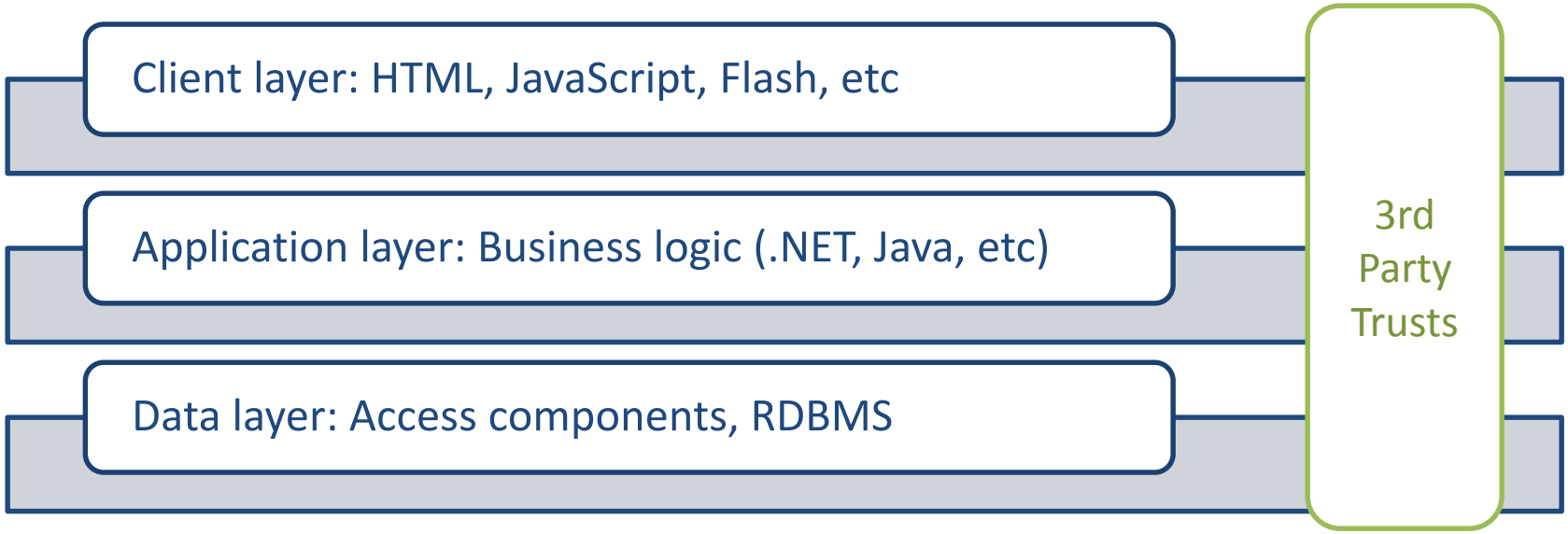
Attack Surface Example

Web application attack surface



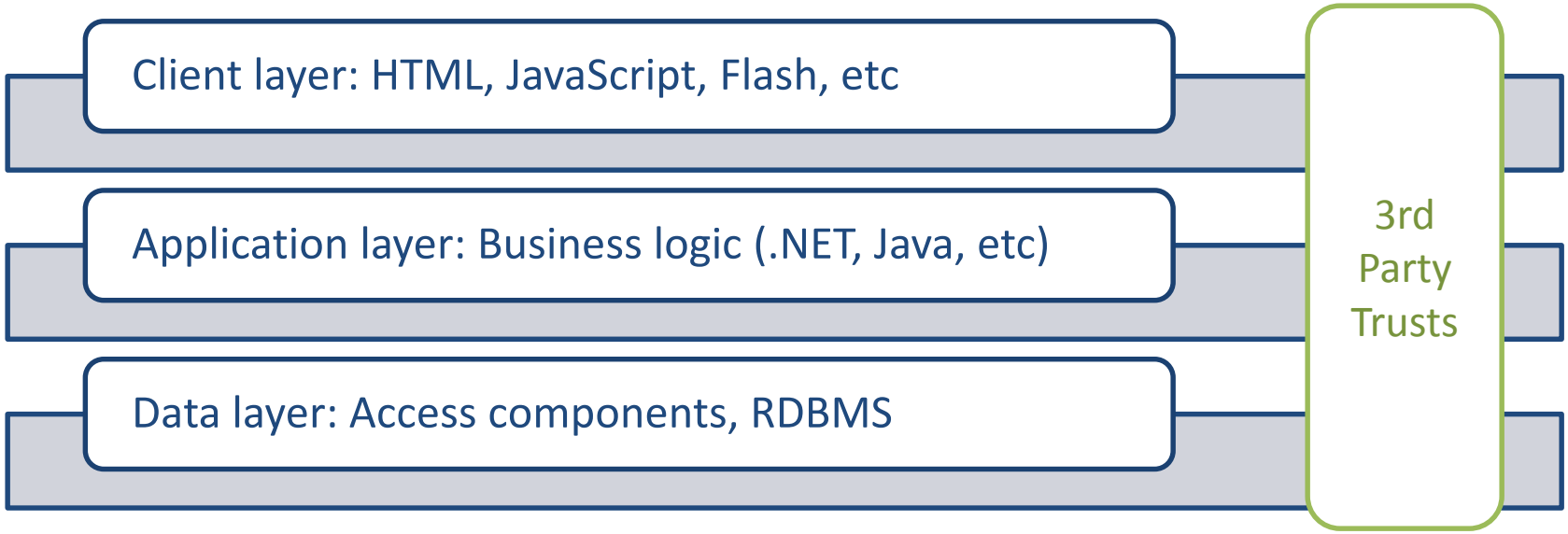
- Client layer:
  - Code and functionality executed on the client-side
  - Malicious javascript can do harm to the client system
  - Various vulnerabilities
  - Two key points:
    - Never trust client-supplied data
    - Don’t introduce vulnerabilities by trying to offload cycles from the server to the client

Web application attack surface



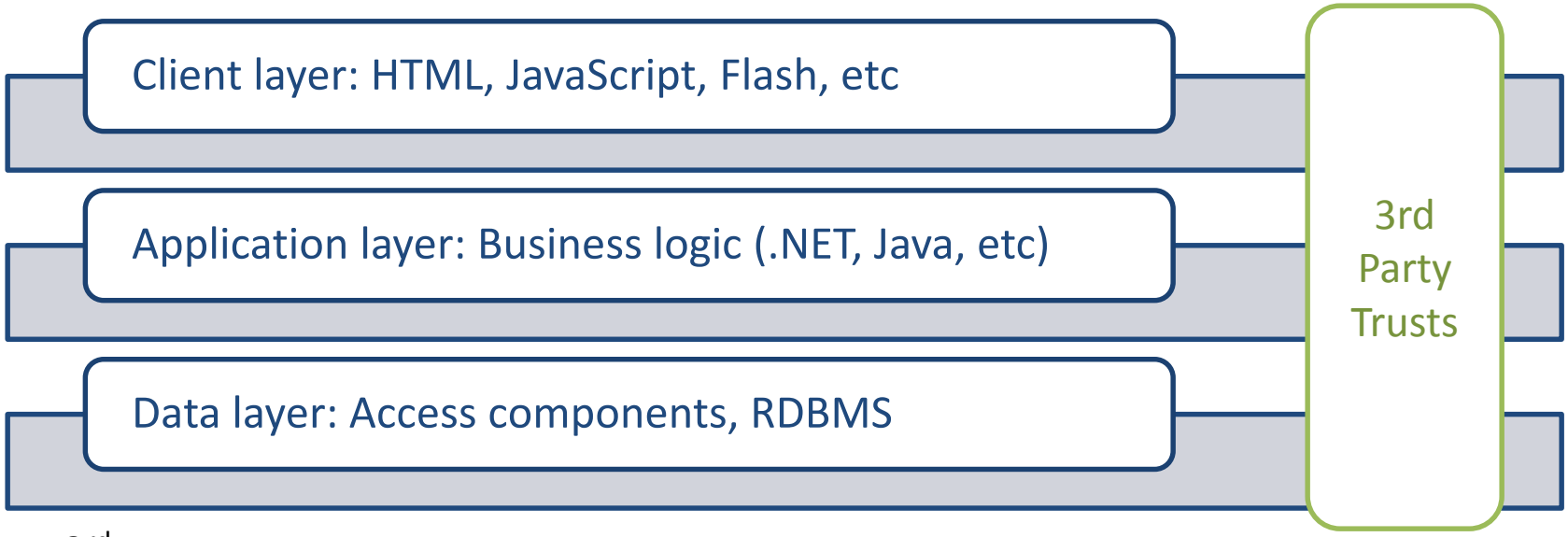
- Application layer:
  - Expression and capture of business rules and business policy logic in code;
  - Besides inputs from web request, are parameters from environment variables, input files safe?;
  - Attacks on specific technology and poor implementation. Does these technologies have vulnerabilities? Are these frameworks and configuration patched?

Web application attack surface



- Data layer:
  - Likely the most valuable component of your application to your organization
  - Less focused on code and logic, more focused on good implementation and maintenance, proper technology
  - Does the communication from data to application layer have a chance to be listened to by attackers?

Web application attack surface



- 3rd Party Trusts:
  - Critical to maintain awareness of system interactions
    - Potential to affect all layers of the application
  - Repeat the attack surfaces at the client, application and data layers for each 3rd party trust

What is Threat Modeling?

A process to understand security threats to a system, determine risks from those threats, and establish appropriate mitigations with the intention to:

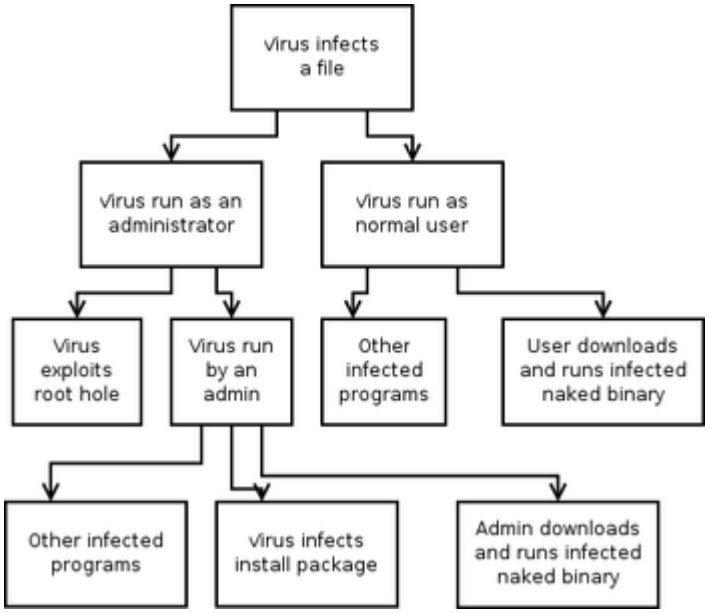
1. Identify where an application is vulnerable
2. Determine which threats require mitigation
3. Reduce risk to an acceptable level through mitigation

What to Threat Model?

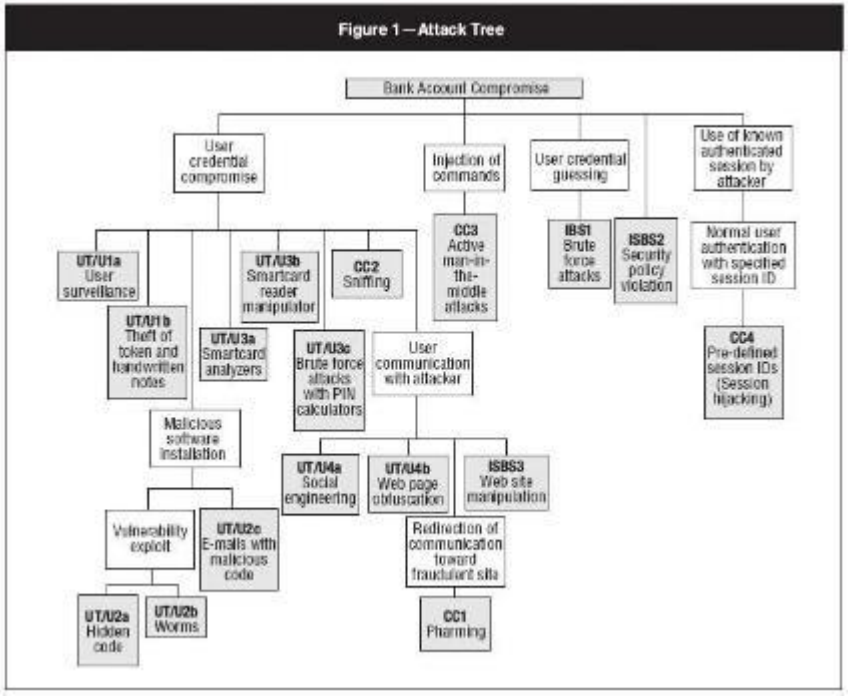
Application as a whole	Security and Privacy Features
Features whose failures have security or privacy implications	Features that cross trust boundaries

Attack Trees

Attack trees are conceptual diagrams showing how an asset, or target, might be attacked. Attack trees have been used in a variety of applications. In the field of information technology, they have been used to describe threats on computer systems and possible attacks to realize those threats. Popularized by Bruce Schneider.

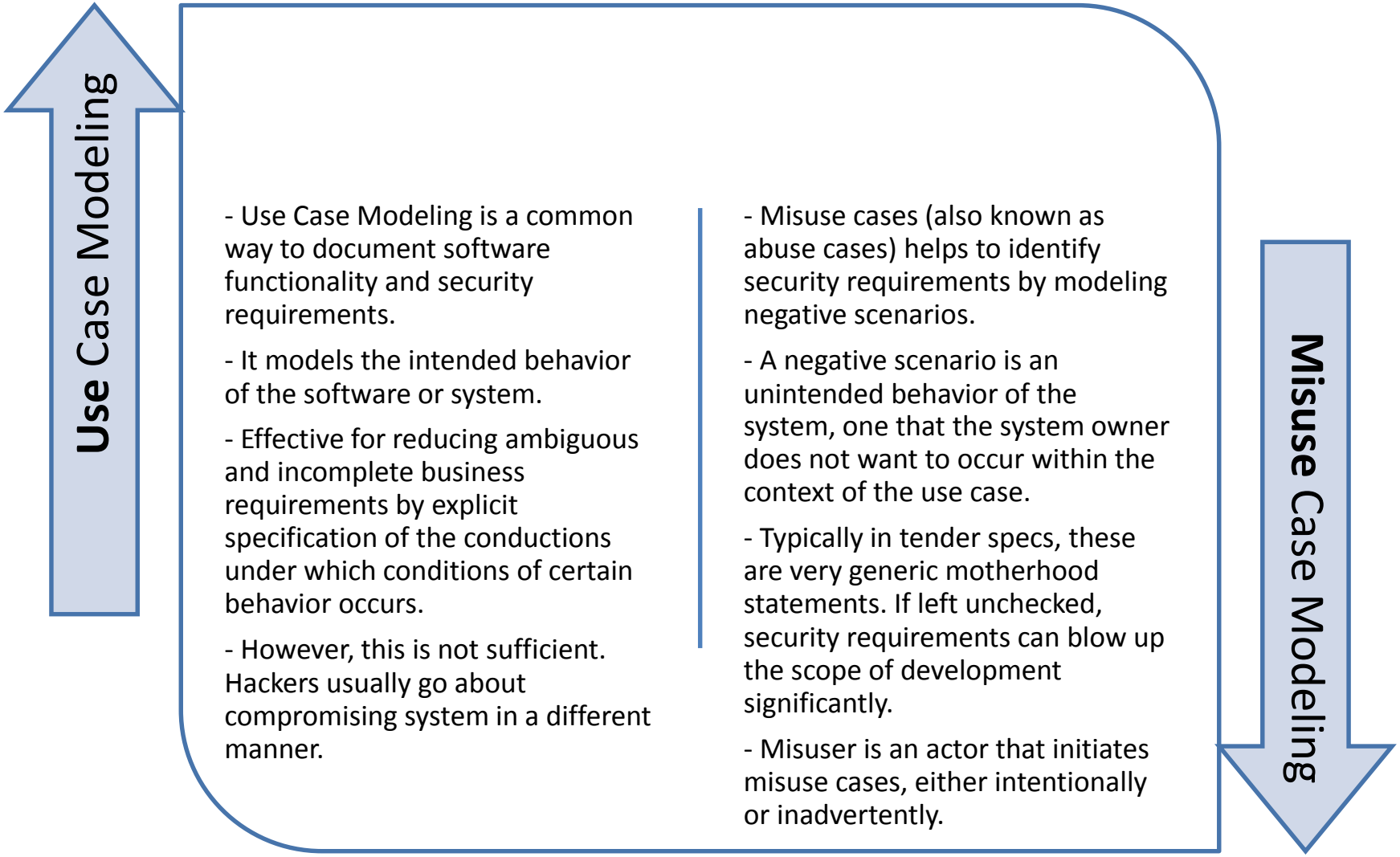


Attack Tree for Computer Viruses
Source: Wikipedia

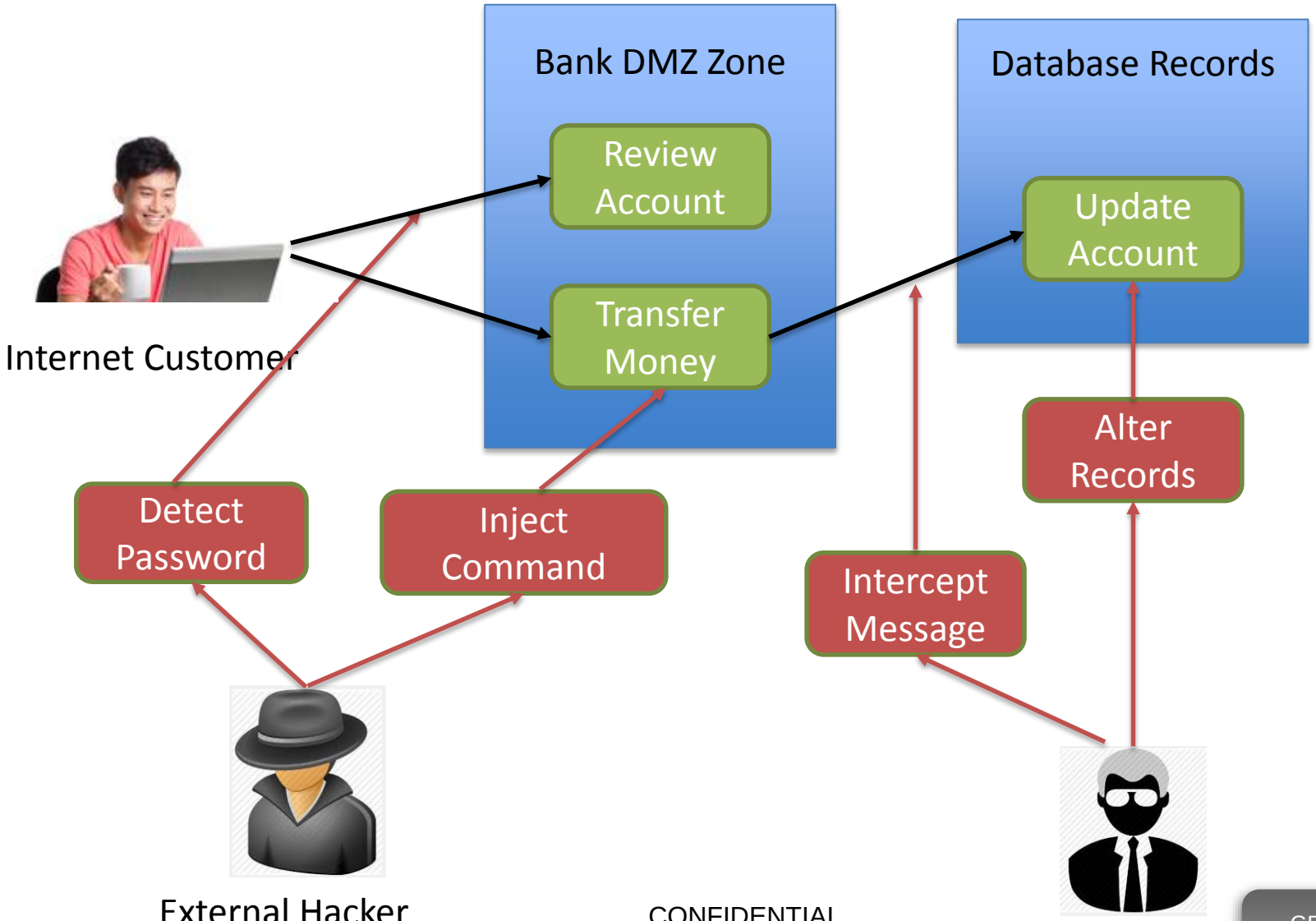


Attack Tree for Bank Account Compromise
Source: http://www.isaca.org/

Misuse Case Modeling



Sample Use Case and Misuse Case



Detect Password Misuse Case

Misuse Case: Detect Password

Summary:
A external hacker obtains and later misuse the internet customer passwords for internet banking by tapping messages sent through a compromised channel.

Basic Flow:
1. The Internet Customer is redirected to a rogue site that looks exactly like the original bank website. This is achieved through DNS poisoning.
2. Internet Customer enters in user name and password.
3. External Hacker sniffs the user name and password, records down for subsequent logins.

Alternate Flow 1:
1. The Internet Customer is redirected to a rogue site that looks exactly like the original bank website. This is achieved through phishing emails sent out promoting special offers. The URL of the rogue site looks very similar to the original one. The rest follows the basic flow.

Alternate Flow 2:
1. The External Hacker calls up the user through phone that cannot be traced by caller id, alerting on potential banking transaction error. Ask the customer for login information to correct these bank errors.
2. Customer gives user name and password through phone. The rest follows the basic flow.

GROUP
ENTERPRISE

Classes of Hackers

TOP "BLACK HAT" HACKERS

- FIRST JUVENILE (16) SENT TO PRISON FOR HACKING
- INSTALLED A BACK DOOR INTO A DEFENSE THREAT REDUCTION AGENCY SERVER
- HACKED INTO NASA COMPUTERS STEALING SOFTWARE WORTH $1.7M, COSTING NASA $41,000 IN REPAIR

HACKED INTO NY TIMES AND MICROSOFT USING WIFI AT COFFEE SHOPS

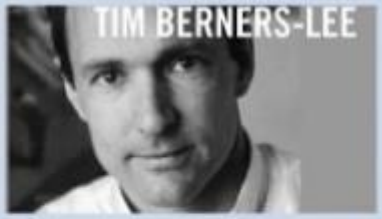
- VIEWED PERSONAL INFO AND HIGH-PROFILE SUBJECT MATTER
- ORDERED TO PAY $65,000 IN RESITUTION
- SENTENCED TO 6 MONTHS HOUSE ARREST AND 2 YEARS OF PROBATION
- IS NOW AN AWARD WINNING SPEAKER AND JOURNALIST

SPENT 2 YEARS STEALING CORPORATE SECRETS AND BREAKING INTO THE US NATIONAL DEFENSE WARNING SYSTEM

- SERVED 5 YEARS TOTAL, INCLUDING 8 MONTHS SOLITARY CONFINEMENT
- CURRENTLY A COMPUTER SECURITY CONSULTANT

TOP "WHITE HAT" HACKERS

CREATED BLUE BOX, A DEVICE THAT BYPASSES TELEPHONE SWITCHING MECHANISMS TO MAKE FREE LONG DISTANCE CALLS



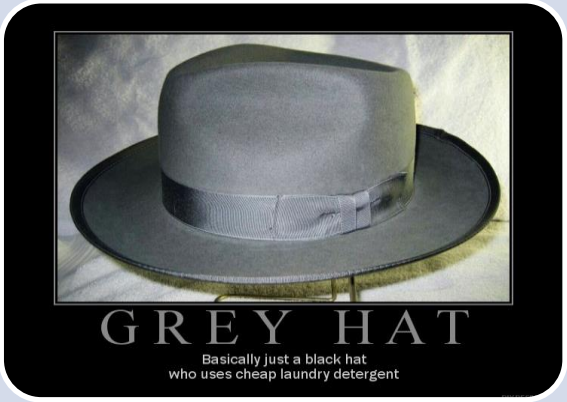
CREATED THE WORLD WIDE WEB BY COMBINING HYPERTEXT FROM A NUCLEAR RESEARCH SYSTEM WITH THE INTERNET

CREATED LINUX

Linux

CREATED GNU PROJECT THAT ENABLED SOFTWARE TO BE USED, DISTRIBUTED, COPIED, AND MODIFIED FOR FREE

What Are the Hacker Hats?



Black Hat

- Violate computer security for pure malicious reasons: Steal credit card number, harvesting personal info.
- Can range from script kiddies to cybercriminal.

Grey Hat

White Hat

- Ethical Hackers.
- Computer security experts to test other organization computers.
- Report found vulnerabilities.
- Ask for permission before attack.

Black Hat Classification (Based on Motivation)

Hacktivists

Hacktivists include hackers with a socio-political focus who use the internet as a tool for protest in order to facilitate socio-political change - their aim being, by their actions, to improve the situation for ordinary people.

Criminal Hackers

Criminal hackers are also known as "Crackers". Like safe-crackers they operate in secret.

These are Black Hat hackers who act for personal (typically financial) gain for themselves or others.

Espionage Hackers

Hackers hired by government to break into computer and networks. For surveillance purpose, monitoring of criminal activities.

Done in secret. Denied by most governments until recently.

Cyber Terrorists

Blackest of the Black Hat. Hacking with the intent and consequence to disrupt normal activities and vital infrastructure of a country or countries.

Seeds the ideas for many block-buster movies.

Hacker Skills Levels

Skills Level

Elite Hacker

- A social status among hackers, *elite* is used to describe the most skilled. Newly discovered exploits circulate among these hackers. Elite groups such as Masters of Deception conferred a kind of credibility on their members

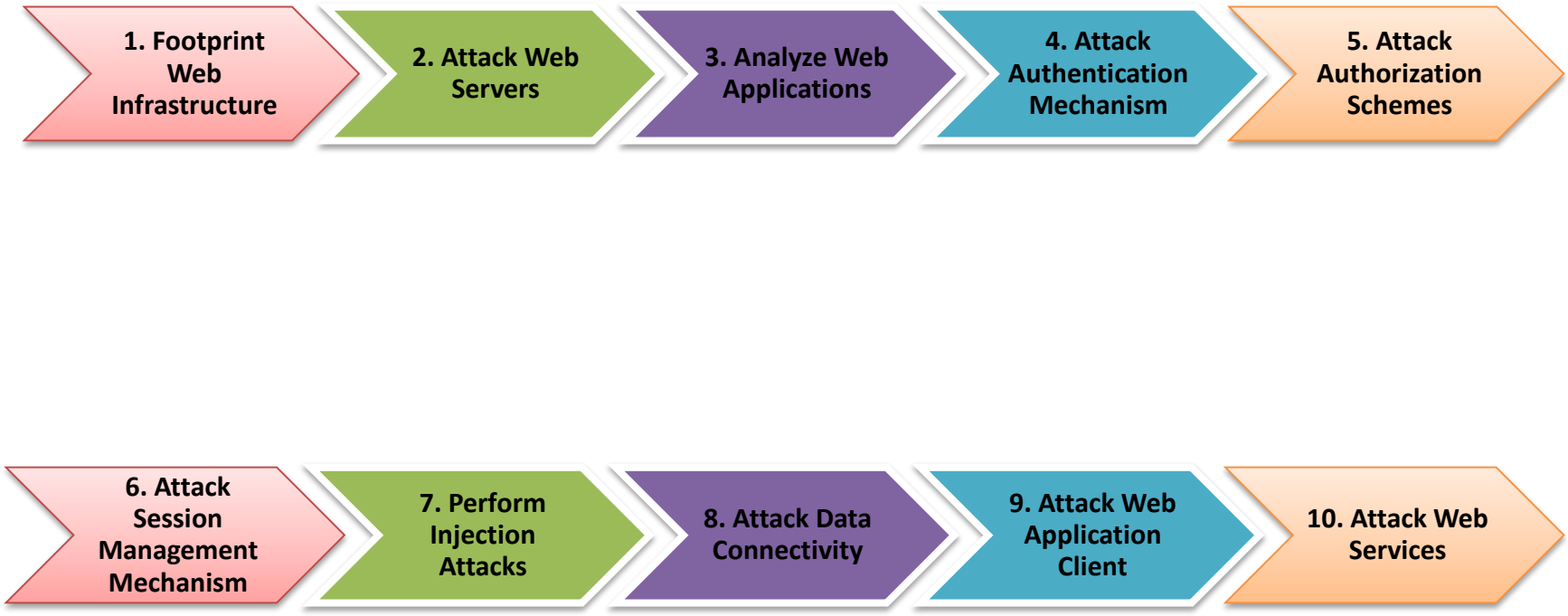
Script Kiddies

- A script kiddie (also known as a *skid* or *skiddie*) is a non-expert who breaks into computer systems by using pre-packaged automated tools written by others, usually with little understanding of the underlying .

Neophyte

- A neophyte, "n00b", or "newbie" is someone who is new to hacking or phreaking and has almost no knowledge or experience of the workings of technology, and hacking.

Web Application Hacking Steps



GROUP
ENTERPRISE

Secure Development Lifecycle

NCS Secure Software Development Lifecycle

1. Requirements

- [Rigor 1 or 2] Conduct detailed security risk assessments of application
- [Rigor 1 or 2] Identify security and privacy requirements, making security requirements explicit
- Identify known vulnerabilities in selected software components and frameworks
- [Rigor 1] Collaborate with customers to tightly define input fields constraints

2. Design

- ❑ [Rigor 1] Conduct Threat Modeling
- ❑ [Rigor 1 or Rigor 2] Design in Applicable Security Control Measures to reduce attack surface area
- ❑ Design Applicable measures to deal with known security vulnerability
- ❑ [Rigor 1 or Rigor 2] Acceptance of residue risk by customers
- ❑ Pass Design Quality Gate

3. Construction

- ❑ [Rigor 1 or Rigor 2] Implement the identified security and privacy requirements
- ❑ Adopt secure and defensive coding practices
- ❑ Employ code review processes
- ❑ Utilize static code analysis tools in identifying coding defects
- ❑ Leverage upon established NCS libraries and framework
- ❑ Pass Construction Quality Gate
- ❑ Compile Code with tight compiler / linker options as default

4. Testing

- ❑ [Rigor 1 or Rigor 2] Conduct vulnerability testing
- ❑ [Rigor 1] Conduct Fuzz Testing
- ❑ Review application, server, OS configurations
- ❑ Conduct risk assessment on detected issues and sequence fixes using a risk based approach
- ❑ [Rigor 1] Attack Surface Review
- ❑ Pass pre-Deployment Quality Gate

Education

- ❑ Developers are trained in secure and defensive coding practices through customized internal courses
- ❑ Project Managers and Architects are equipped with secure software development principles
- ❑ Regular IT security education refresher

5. Go Live

- ❑ Harden server and infrastructure settings
- ❑ Utilize latest version of stable COTS product whenever possible to reduce risk exposure
- ❑ [Rigor 1 or Rigor 2] Create response procedures to deal with security threat
- ❑ Constant monitoring of application security health status. Review of logs to identify abnormal activities
- ❑ Apply regular security patches

Security Risk Assessment – Privacy Risk Assessment

- A security risk assessment (SRA) is an exercise to identify functional aspects of the software that might require deep security review. Given that program features and intended functionality might be different from project to project, it is wise to start with a simple SRA and expand it as necessary to meet the project scope.
- Such assessments must include the following information:
  - What portions of the project will require threat models before release.
  - What portions of the project will require security design reviews before release.
  - What portions of the project will require penetration testing (*pen testing*) by a mutually agreed-upon group that is external to the project team. Any portion of the project that requires pen testing must resolve issues identified during pen testing before it is approved for release.
  - Any additional testing or analysis requirements the security advisor deems necessary to mitigate security risks.
  - Clarification of the specific scope of *fuzz testing* requirements. (Verification Phase: Security and Privacy Testing discusses fuzz testing.)

Risk Assessment Questionnaire (I)

- **Audience**
  - What type of user access does your application offer (internal, external [Internet-facing], both, or neither)?
  - What is the basic authentication and authorization for the external-facing (Internet) portion of your application?
  - Are there anonymous users?
  - Is there a secure channel? What is that channel?
- **Data Classification**
  - What type of data is contained in your application?
  - Does your application contain personal data?
  - How business-sensitive is the data managed by your application?
- **Functionality**
  - What function does your application fulfill? How critical is its role?
- **Architecture**
  - What is the authentication mechanism used by the client population?
  - Does your application have multiple user roles (for example, user and admin)?
  - Is code executed on the client machine (for example, ActiveX control, assembly)?
  - Where will your application be deployed?
- **Process Control**
  - What type of source control do you use for your application?
- **Privacy Release Issues**
  - Will the privacy statement or legal notice that was used in the existing application version change for this release? Is there a new privacy statement or legal notice available?
- **Security Release Issues**
  - Does this version include changes to the authentication mechanism?
  - Does this web application or service provide functionality to other applications?

Risk Assessment Questionnaire (II)

- **Security Impact**
  - Does this application handle personal information (employees, customers, business partners)?
  - Does this application handle business sensitive data?
  - Is this application key to providing a service or generating a product?
  - Is this application key to running the business (finances, for example)?
  - Who will have access to this application?

GROUP
ENTERPRISE

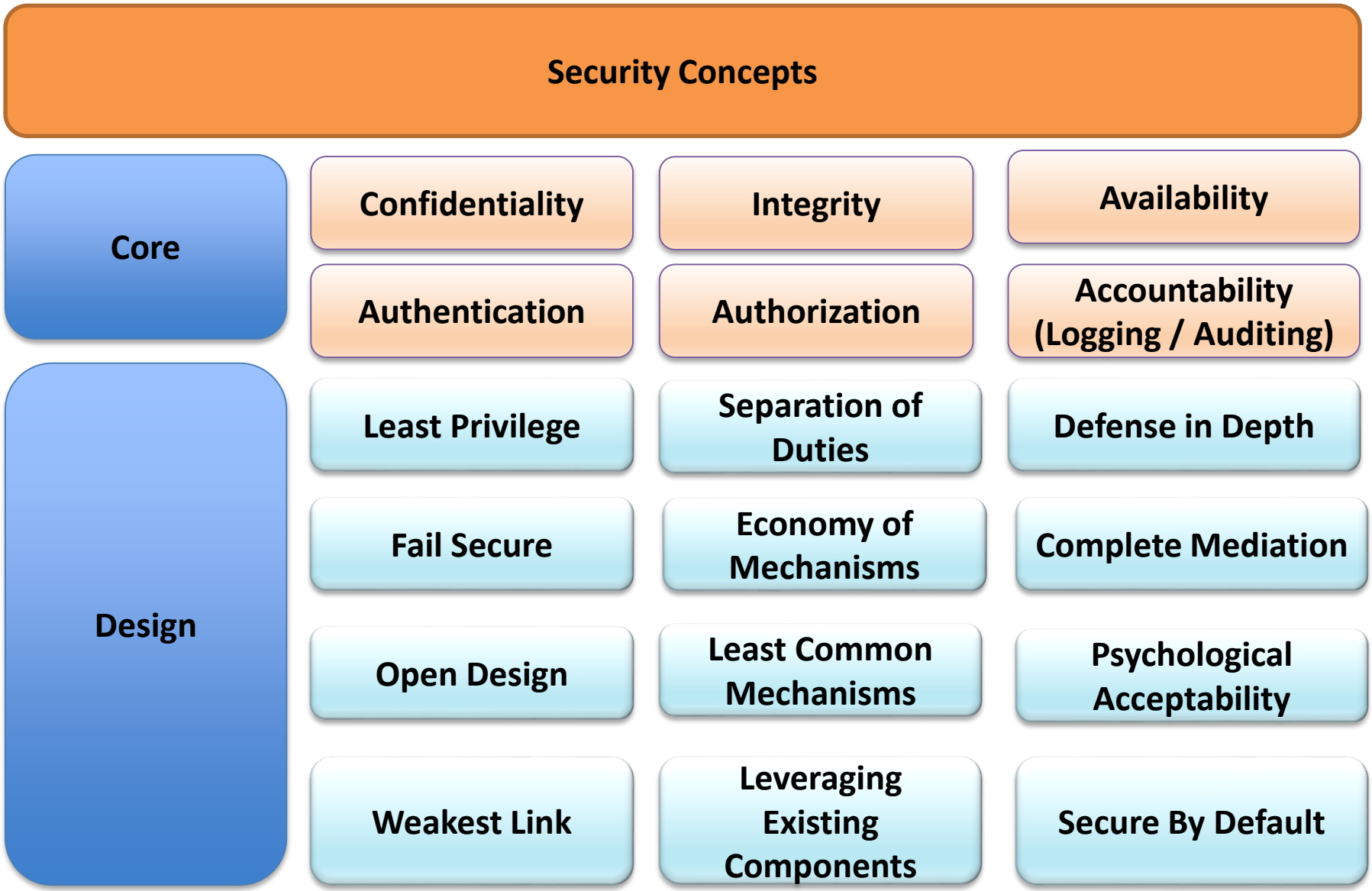
Secure Development Principles

Antipatterns: Security As an Afterthought

- In many past IT projects, secure features are often added on at a later stage. Many added on security features are knee-jerk reactions to perhaps security breeches in production or vulnerabilities discovered during the end-point security tested.
- End Result: High cost of fixing security features at the end. Some changes may require architectural and platform changes which are prohibited expensive.

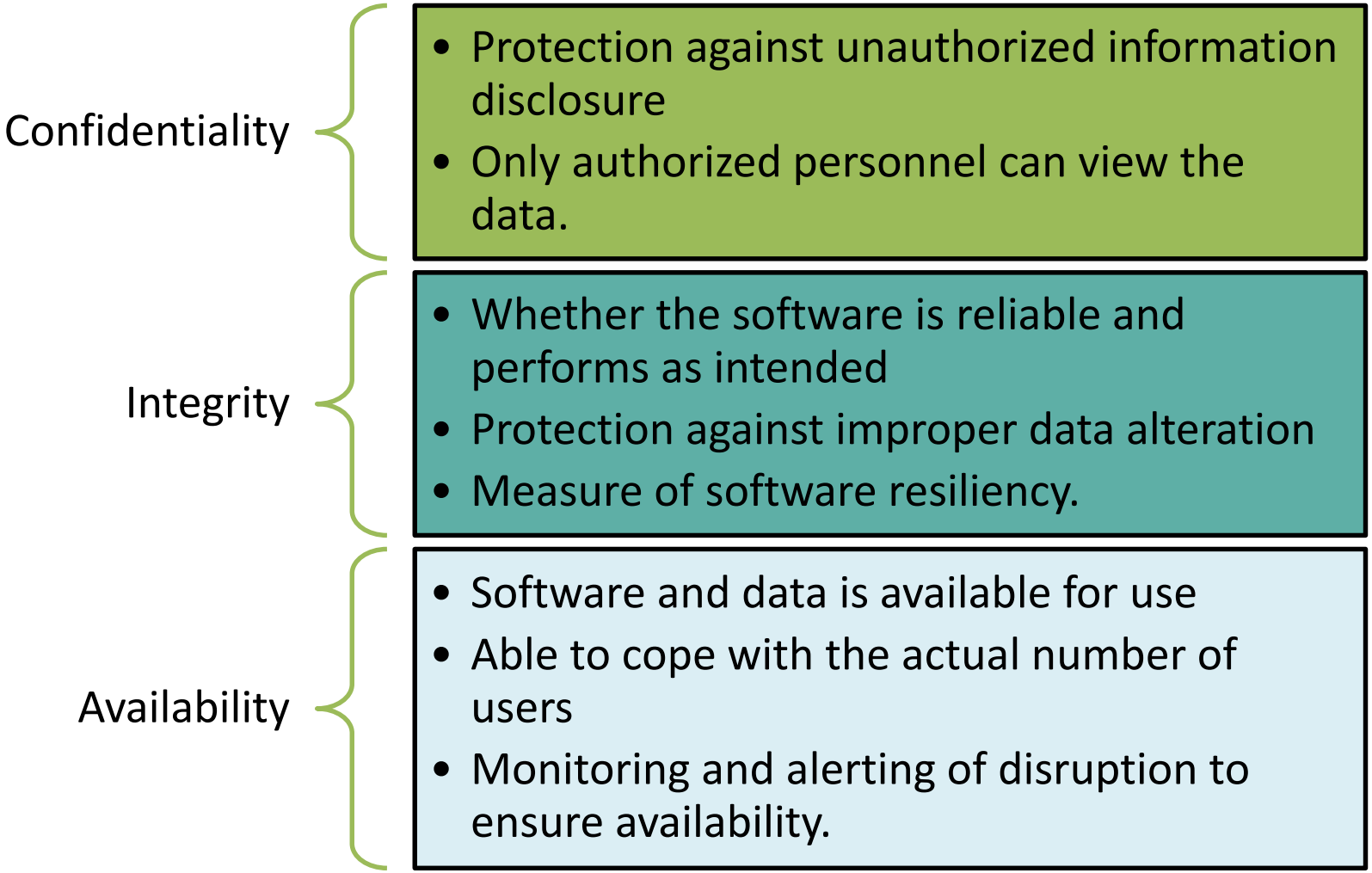
An end to end secure Development Lifecycle is required. Security should be built into the software, not bolted on.

Secure Development Principles



Source: CSSLP Common Body of Knowledge.

Core Security Principles (1)



Core Security Principles (2) – Triple A

Authentication

- Answers are you whom you claim to be?
- Ensures information are available to the right people at the right time.
- Authentication Scheme
  - Knowledge : Something one knows. PIN, password etc
  - Ownership : Something one own or have. Security token, smart card
  - Characteristics: Something one is. Biometrics such as thumbprint, retina scan etc.
- Whether there is need for Multifactor authentication.

Authorization

- Access to objects is controlled based on rights and privileges.
- Granted by the owner of the data or system according to policy.
- Layered on top of authentication.

Accountability

- Auditing privileged and critical business transactions, with history of events, timelog for troubleshooting and forensic evidence. Passive control mechanism. Auditing is a both a detective and deterrent control.
- Non repudiation address the deniability of actions taken by a user or software.

Design Security Principles (1)

Least Privilege	Separation of Duties / Compartmentalization Principle	Defense in Depth	Fail Secure
• Person and process is given only minimum level of access rights. • Additional rights are given incrementally. • May result in a lot of administrative overhead activities.	• Successful completion of a single task is dependent of two or more conditions to be met. • No one holds all the keys. • Require carefully planned out coordination across departments and may limit organizational knowledge sharing.	• Single points of compromise are eliminated by multiple layers of security safeguards. • Can be expensive to build these multiple levels of defense. • Can give false perception of safety when people start to think there are other people or mechanism who will catch the problem.	• Maintaining confidentiality, integrity and availability by defaulting to secure state, rapid recovery upon design or implementation failure.

Least Privilege

- When software is operating with least privilege, it means only the necessary and minimum level of access rights has been given.
- Containment of damage that can result from a security breach, similar to the “need-to-know” clearance level classification practice seen in military.
- This is achieved through
  - Running programs with non-administrative OS accounts that has just sufficient rights to accomplish its tasks.
  - Employment of modular programming inside application software. Accessibility to modules are controlled by access control mechanism inside the application.

Separation of Duties

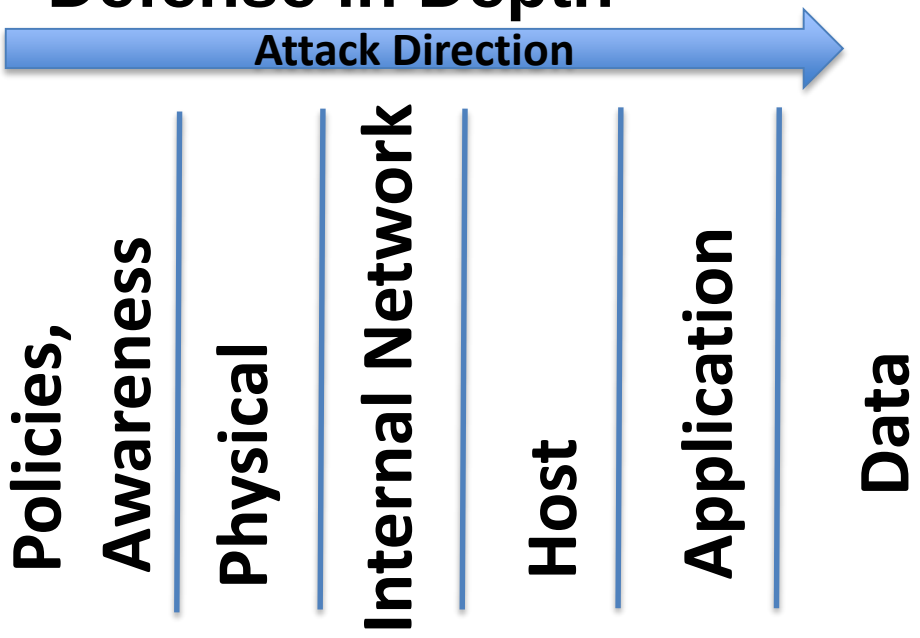
- Compartmentalizing software functionality into two or more conditions, all of which is necessary before an operation can be completed, is referred, separation of duties.
- Examples
  - Sometimes one man cannot be trusted so certain locks require two, or more, people to unlock, simultaneously. This is a example of separation of duties.
  - Developer cannot review his own code, developer has no access to deploy code to production environment.
  - Chinese Wall Concept
- When implemented correctly, this will reduce extent of damage caused by one person. In conjunction with auditing, this can discourage insider fraud.

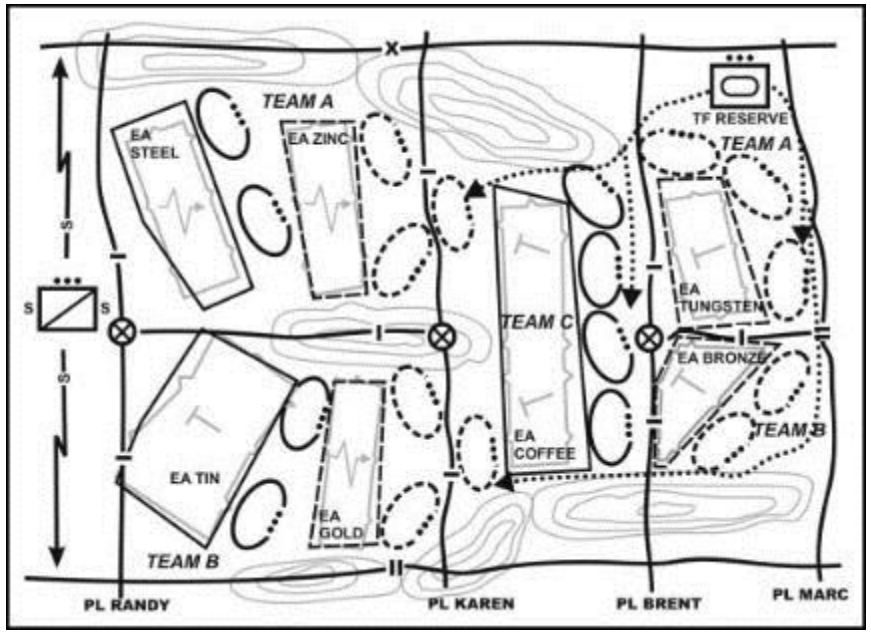
Chinese Wall

A Chinese wall is a barrier that separates two or more groups, usually as a means of restricting the flow of information. Typically, the wall is purely conceptual, although groups may be divided by physical barriers (areas of a building, for example) as well as policies. The concept of the Chinese wall is employed in a wide variety of environments, including the financial industry, business, software development, project management, network security, law, and journalism.

Source: http://www.investmentbanking.net/training_guide/index.cfm?chapter=1

Defense in Depth





- Defense in Layer Concept is age old principle.
- Started in military thinking
  - Security screening forces
  - Main Defense Area
  - Rear Defense Area
  - Key Asset always hidden in the internal layer
- However, most applications today **can be compromised when single, and often only, layer of defense is breached** (application)! Weakest Link concept sometimes will apply.

Design Security Principles (2)

Economy of Mechanisms	Complete Mediation	Open Design	Least Common Mechanisms
• Keep it Simple. • Complexity increases the likelihood of vulnerabilities.	• Ensure authority is not circumvented by checking for authorization upon every request for object : For example, login required per every action. • May lead to performance issue.	• Open review of security design will lead to strong protection mechanism. • Complete different mindset from Design Obscurity. • Requires quick patches upon public exposure to weaknesses.	• Disallows sharing of mechanisms that are common to user or process.

Design Security Principles (3)

Psychological Acceptability	Weakest Link	Leveraging Existing Components	Secure By Default
• Security principle aims at maximizing usage and adoption of security functionality by ensuring security functionality is easy to use and transparent to the user.	• Resiliency to attacks will depend of the protection of the weakest link.	• Ensure attack surface is not increased by promoting reuse of existing software components, code and functionality. • Issue in reused libraries and components may affect more projects.	• Standard configuration file is tight by Default. No Access. • Development team members will need to slowly loosen it. This however may cause additional inconvenience.

Secure By Default

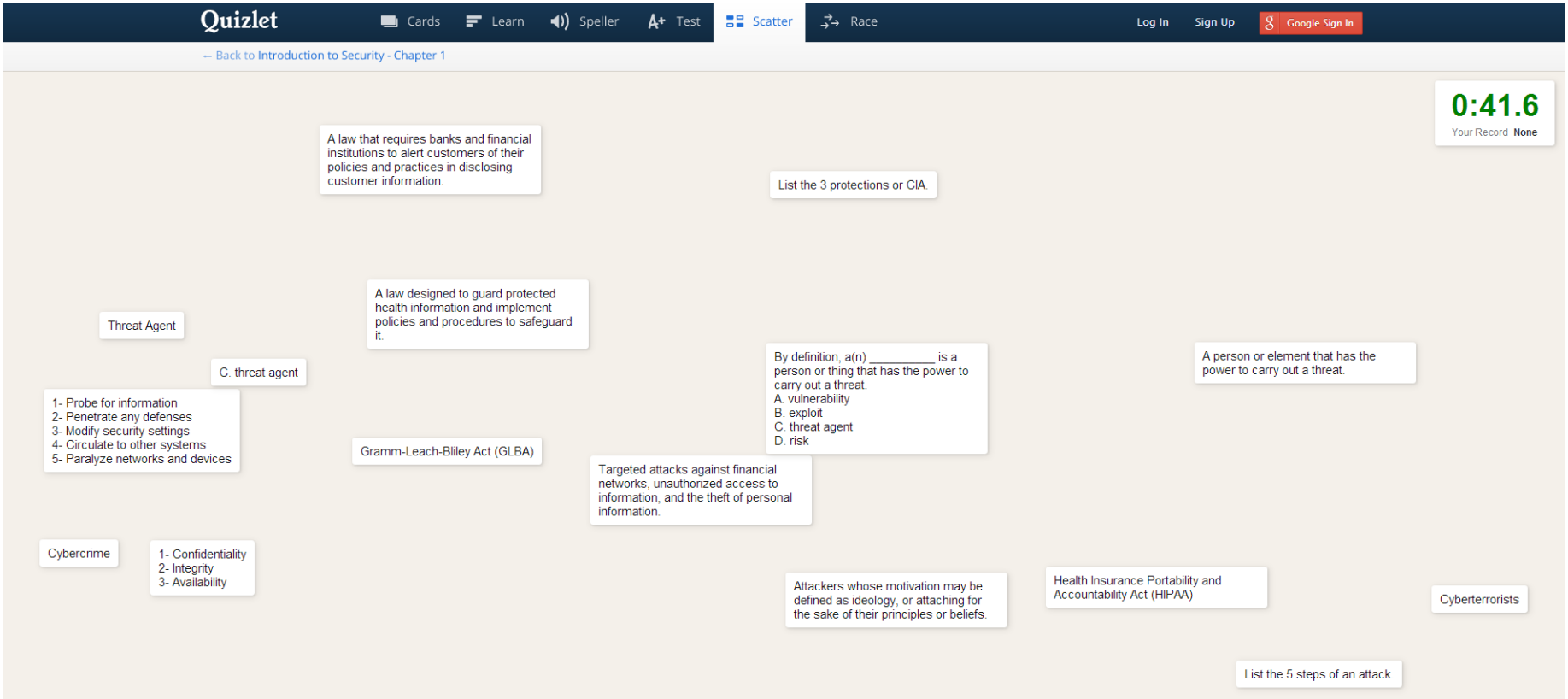
Application Component	Secure Defaults Principle
Firewall	Firewall ON by default
SSL Socket	Requires last latest SSL version (v3, TLS, etc.) by default
User can access application anonymous or authenticated	Application requires authenticated user sessions by default
Password complexity can be enforced	Password complexity is required by default
Store user passwords as hashes or clear text	Store user passwords as hashes by default

Standard configuration file is tight by Default. No Access unless required.

Development team members will need to slowly loosen it. This however may cause additional inconvenience.

Let's Play A Quiz on General Security Terms

- http://quizlet.com/23988644/scatter



GROUP
ENTERPRISE

Password Policies

Authentication Vulnerabilities

- CWE-645: The software contains an account lockout protection mechanism, but the mechanism is too restrictive and can be triggered too easily. This allows attackers to deny service to legitimate users by causing their accounts to be locked out.

- CWE-306: The software does not perform any authentication for functionality that requires a provable user identity or consumes a significant amount of resources
  - If an attacker can directly invoke a privileged function they may be able to bypass authentication.

- CWE-640: The software contains a mechanism for users to recover or change their passwords without knowing the original password, but the mechanism is weak.
  - Yahoo! mail used birthdate, zip code, and where you met your spouse to reset passwords.

Password Strength

Weak Passwords

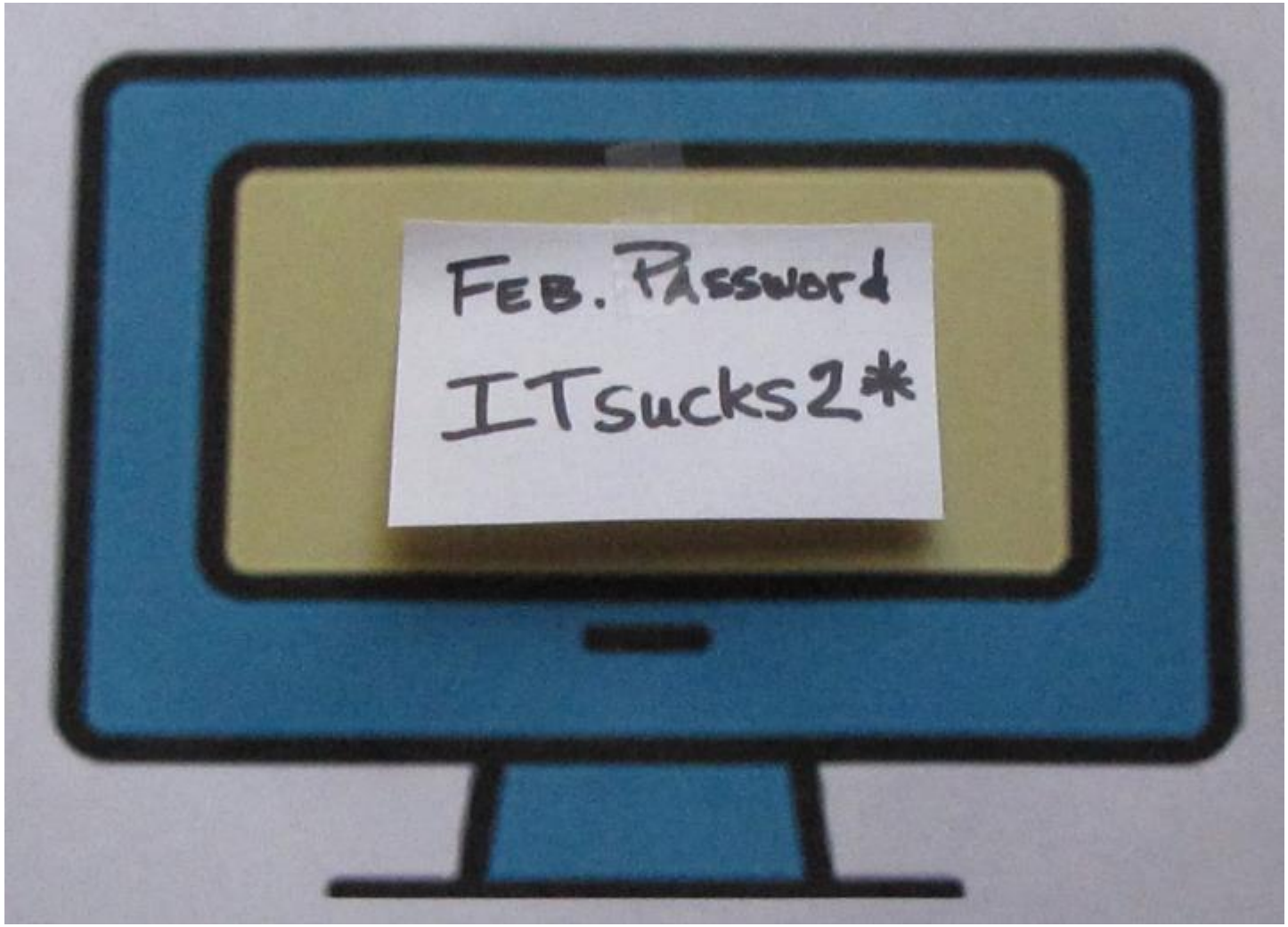
- The password contains less than 15 characters.
- The password is a word found in a dictionary (English or foreign)

Strong Passwords

- Contain both upper and lower case characters
- Have digits and punctuation characters
- Are at least 15 alphanumeric characters long and is a passphrase.
- Are not a word in any language , slang , dialect , jargon.
- Are not based on personal information.
- Passwords should never be written down or stored on-line.
- Regular forced password change every 3 months.

Source: Password Policy. SANS 2006

Side Effects of Too Strong A Password



Policies for Password Storage

- Passwords should be stored as a hash
  - SHA-512 is preferred, but SHA-256 is ok

- Add uniqueness value (aka, “salt”) to password before hashing
  - Salts should be per-user account to defend against dictionary attacks
  - Salt value should be randomly generated
  - Salt value does not need to be protected and can be stored in plaintext

One Time Passwords

- A **one-time password** (OTP) is a password that is valid for only one login session or transaction.
  - OTPs avoid a number of shortcomings that are associated with traditional (static) passwords.
  - OTP prevents replay attacks.
- Requires additional cost to implement OTP

Group Activities

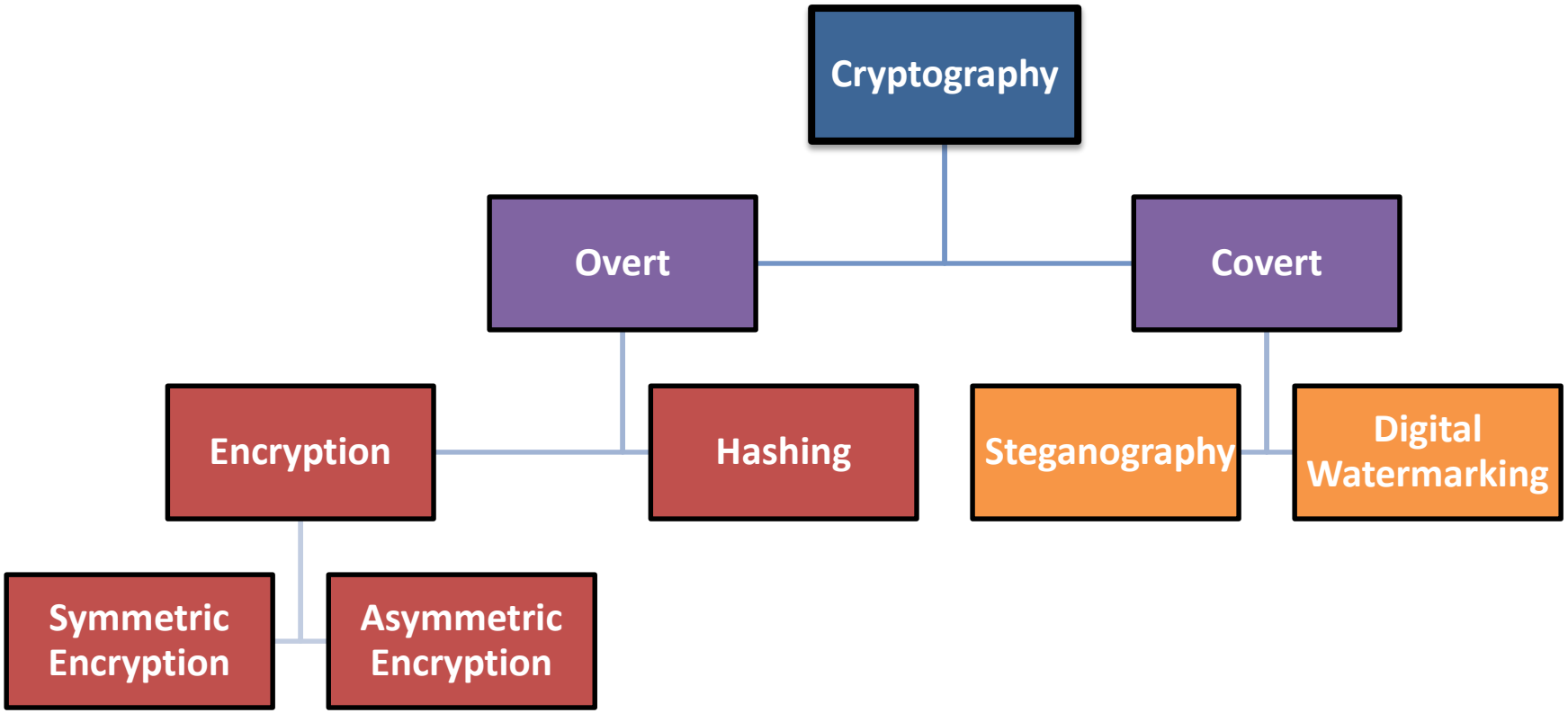
- A) At class level, each individual to share on
  - A1. Who is the current customer? What is the current role?
  - A2. The interesting SW security matters faced in the past projects
  - A3. The topics which will like this course to cover

- B) At group level, discuss through the following questions with respective to secure coding practices:
  - B1. What type of activities are well carried out in the current NCS Projects?
  - B2. What are the current gaps areas that probably needs to be improved upon? And at which SDLC stages will these activities be appropriate to be injected in

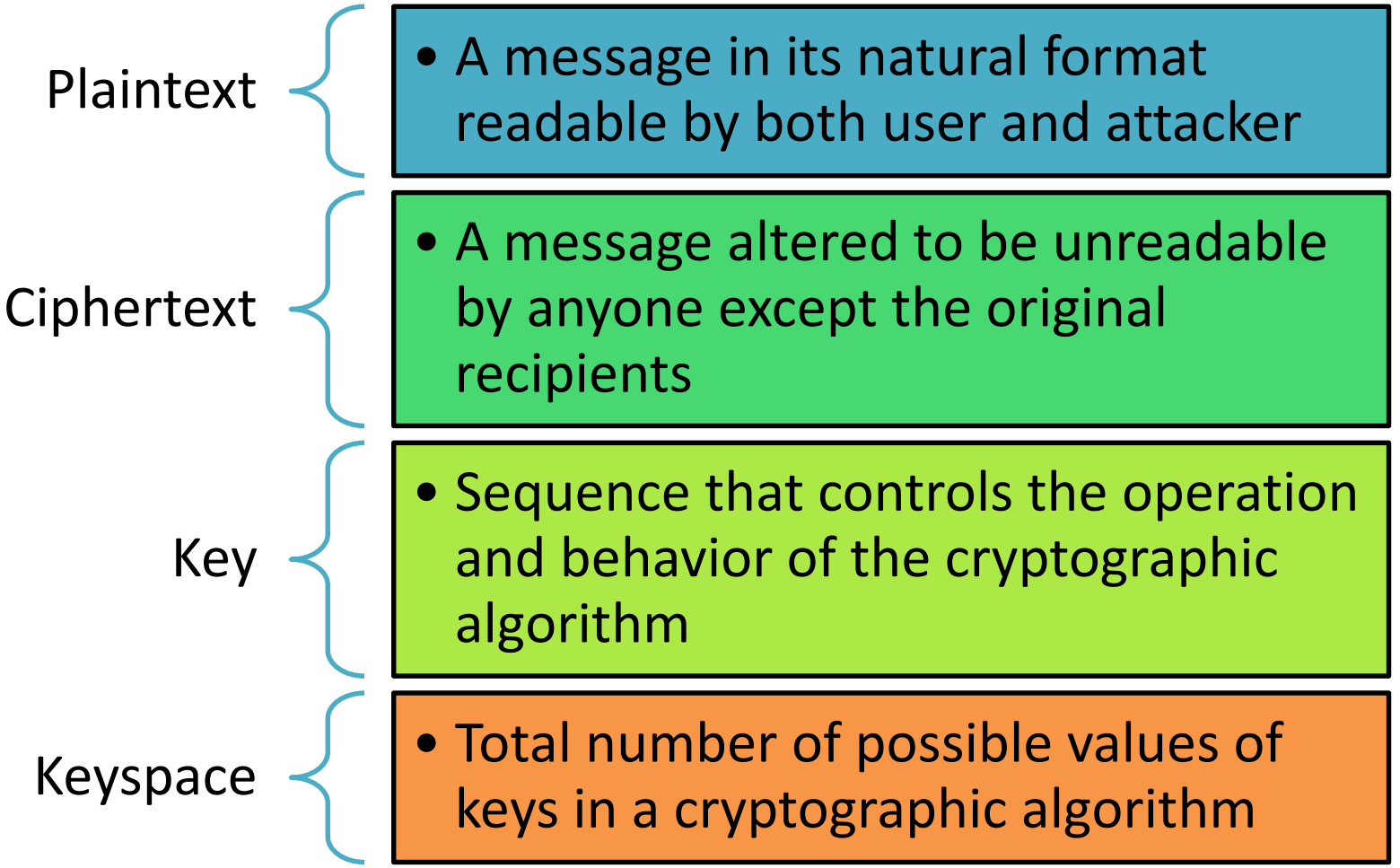
GROUP
ENTERPRISE

Cryptography

Types of Cryptography – Confidentiality Control

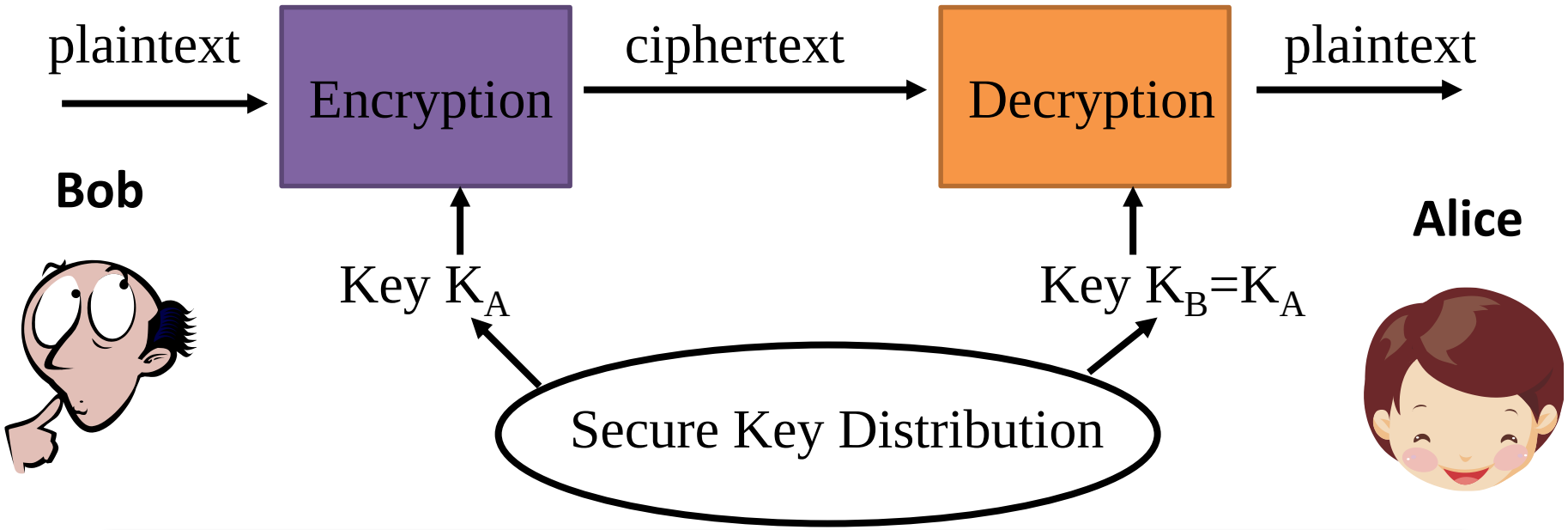


Cryptography Terminology



Symmetric Encryption

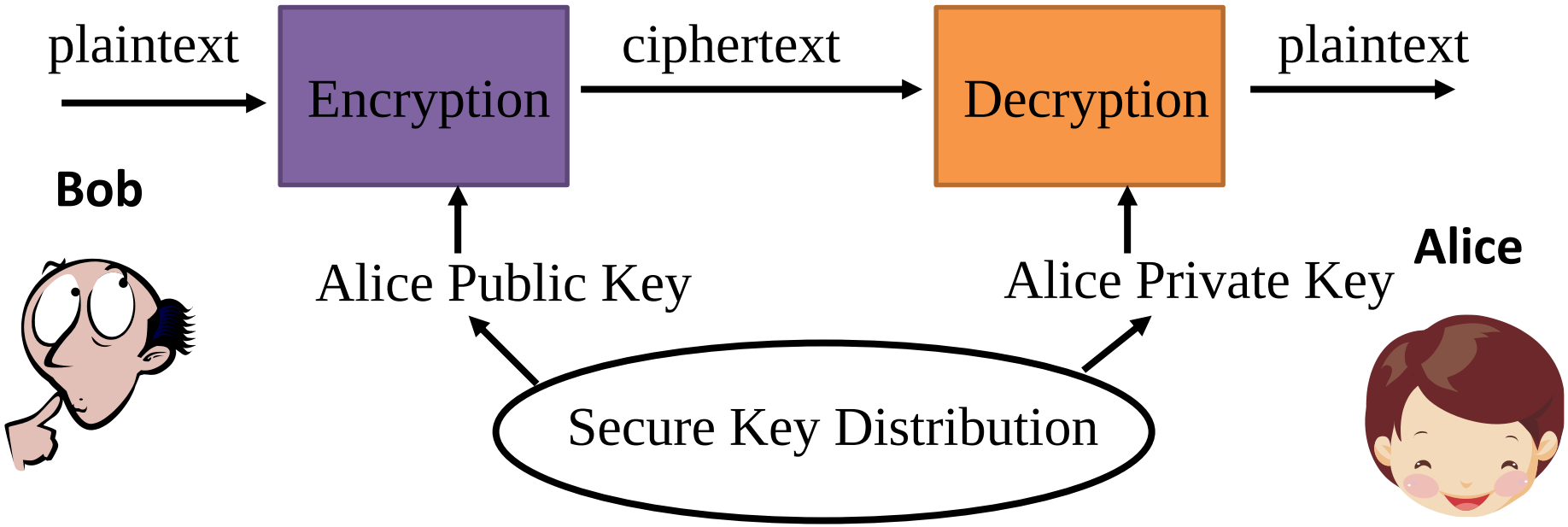
- Uses same Key for encryption and decryption
- Strength of secret depends on the possession of the key
- Big key distribution problems
- Faster than asymmetric encryption



DES, 3DES (Triple DES), AES, IDEA, Blowfish, RC4, RC5, CAST, SAFER, Twofish

Asymmetric Encryption

- Uses a key pair { private and public key}. Key pair is mathematically related.
- Slower than symmetric encryption



Diffie-Hellman, RSA, El Gamal, Elliptic Curve Cryptography

RSA Key Pair

RSA Key pair

(including Algorithm identifier)

[2048 bit]

Private Key

```
3082 010a 0282 0101 00b1 d311 e079 5543 0708 4ccb 0542 00e2 0d83 463d e493 bab6
06d3 0d59 bd3e c1ce 4367 018a 21a8 efbc ccd0 a2cc b055 9653 8466 0500 da44 4980
d854 0aa5 2586 94ed 6356 ff70 6ca3 a119 d278 be68 2a44 5e2f cfcc 185e 47bc 3ab1
463d 1ef0 b92c 345f 8c7c 4c08 299d 4055 eb3c 7d83 deb5 f0f7 8a83 0ea1 4cb4 3aa5
b35f 5a22 97ec 199b c105 68fd e6b7 a991 942c e478 4824 1a25 193a eb95 9c39 0a8a
cf42 b2f0 1cd5 5ffb 6bed 6856 7b39 2c72 38b0 ee93 a9d3 7b77 3ceb 7103 a938 4a16
6c89 2aca da33 1379 c255 8ced 9cbb f2cb 5b10 f82e 6135 c629 4c2a d02a 63d1 6559
b4f8 cdf9 f400 84b6 5742 859d 32a8 f92a 54fb ff78 41bc bd71 28f4 bb90 bcff 9634
04e3 459e a146 2840 8102 0301 0001
```

Public Key

```
3082 01e4 f267 0142 0f61 dd12 e089 5547 0f08 4ccb 0542 00e2 0d83 463d e493 bab6
0673 0d59 bf3e c1ce 4367 012a 11a8 efbc ccd0 a2cc b055 9653 8466 0500 da44 4980
d8b4 0aa5 2586 94ed 6356 ff70 6ca3 a119 d278 be68 2a44 5e2f cfcc 185e 47bc 3ab1
463d 1df0 b92c 345f 8c7c 4c08 299d 4055 eb3c 7d83 deb5 f0f7 8a83 0ea1 4cb4 3aa5
b35f 5a22 97ec 199b c105 68fd e6b7 a991 942c e478 4824 1a25 193a eb95 9c39 0a8a
cf42 b250 1cd5 5ffb 6bed 6856 7b39 2c72 38b0 ee93 a9d3 7b77 3ceb 7103 a938 4a16
6c89 2aca da33 1379 c255 8ced 9cbb f2cb 5b10 f82e 6135 c629 4c2a d02a 63d1 6559
b4f8 cdf9 f400 84b6 5742 859d 32a8 f92a 54fb ff78 41bc bd71 28f4 bb90 bcff 9634
04de 45de af46 2240 8410 02f1 0001
```

Private Keys in Hardware Tokens

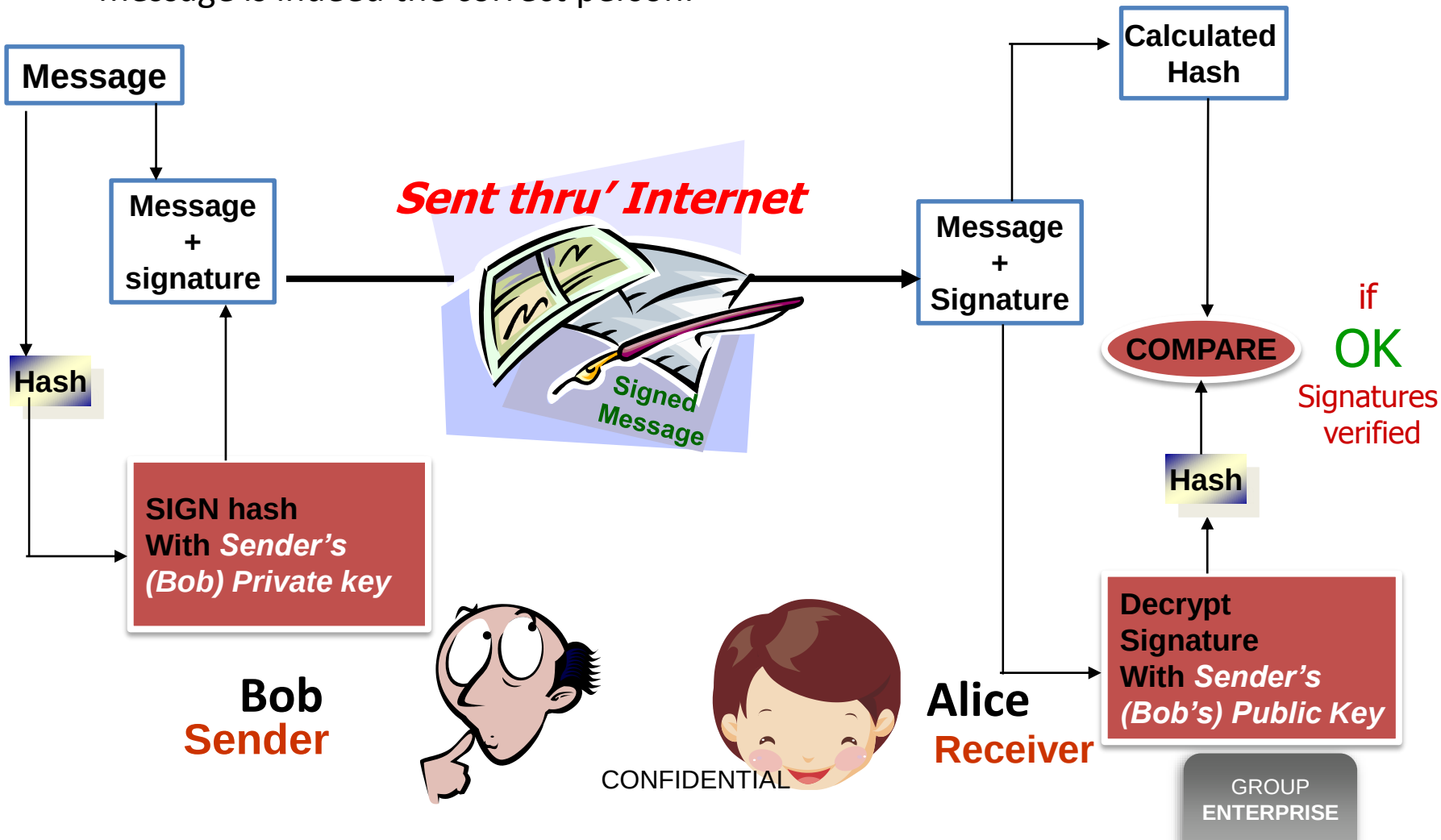
Asymmetric keys are used quite prevalently. The private keys are stored inside iKey and Smart Cards.

iKey

Smart Card

Digital Signatures

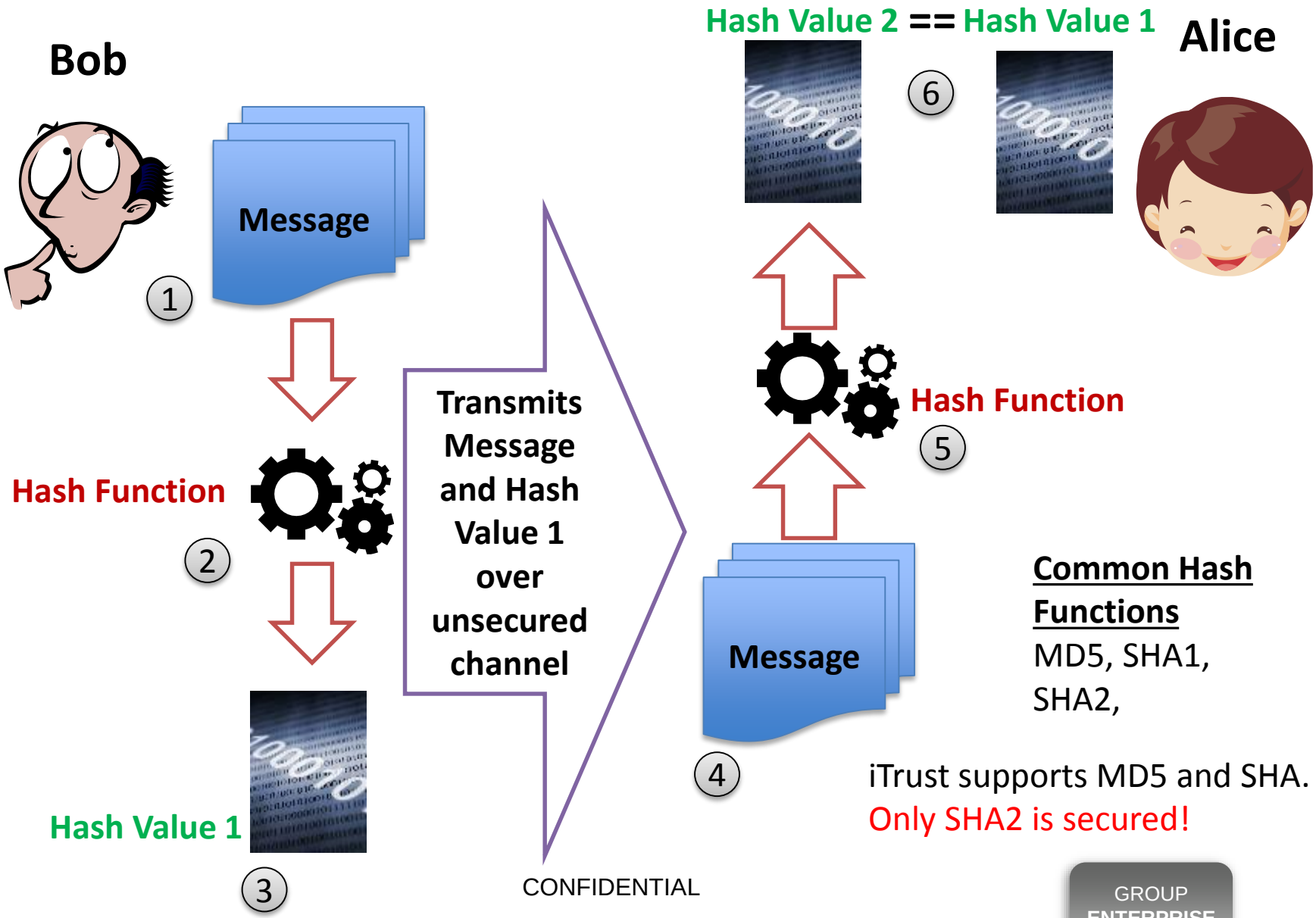
- Digital Signature used asymmetric cryptography to simulate the security properties of a signature in digital rather than the written form.
- It provides benefits of non-repudiation. ie, the person who send the message is indeed the correct person.



X.509 v3 Digital Certificate

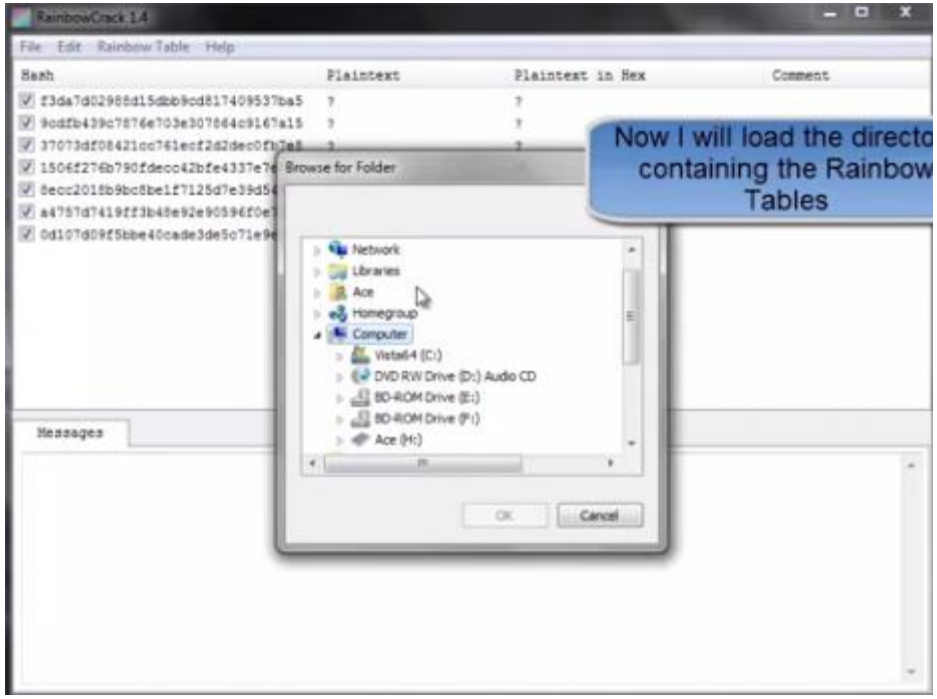
X.509 v3 Certificate			
Required	Certificate	Version	
		Algorithm Identifier	
		Serial Number (unique identifier for each certificate the CA issues)	
	Issuer	Distinguished CA name	
	Validity Period	Period from and to which the certificate will be valid	
	Subject	Name (same as the Issuer for a root certificate)	
		Public key	Algorithm Identifier
			Value
Optional	Unique Identifier	Issuer	
		Subject	
	Extensions	Additional certificate and policy information	
Digital Signature of the CA			

Ensuring Data Integrity Via One Way Hashing



Rainbow Table

A rainbow table is a pre-computed table for reversing cryptographic hash functions. Tables are usually used in recovering plaintext password up to a certain length. This helps to speed up brute-force attacks where calculating hashes is computationally expensive.



You tube video on how to use Rainbow Table to crack passwords

http://www.youtube.com/watch?v=OfdbSPUDgsg

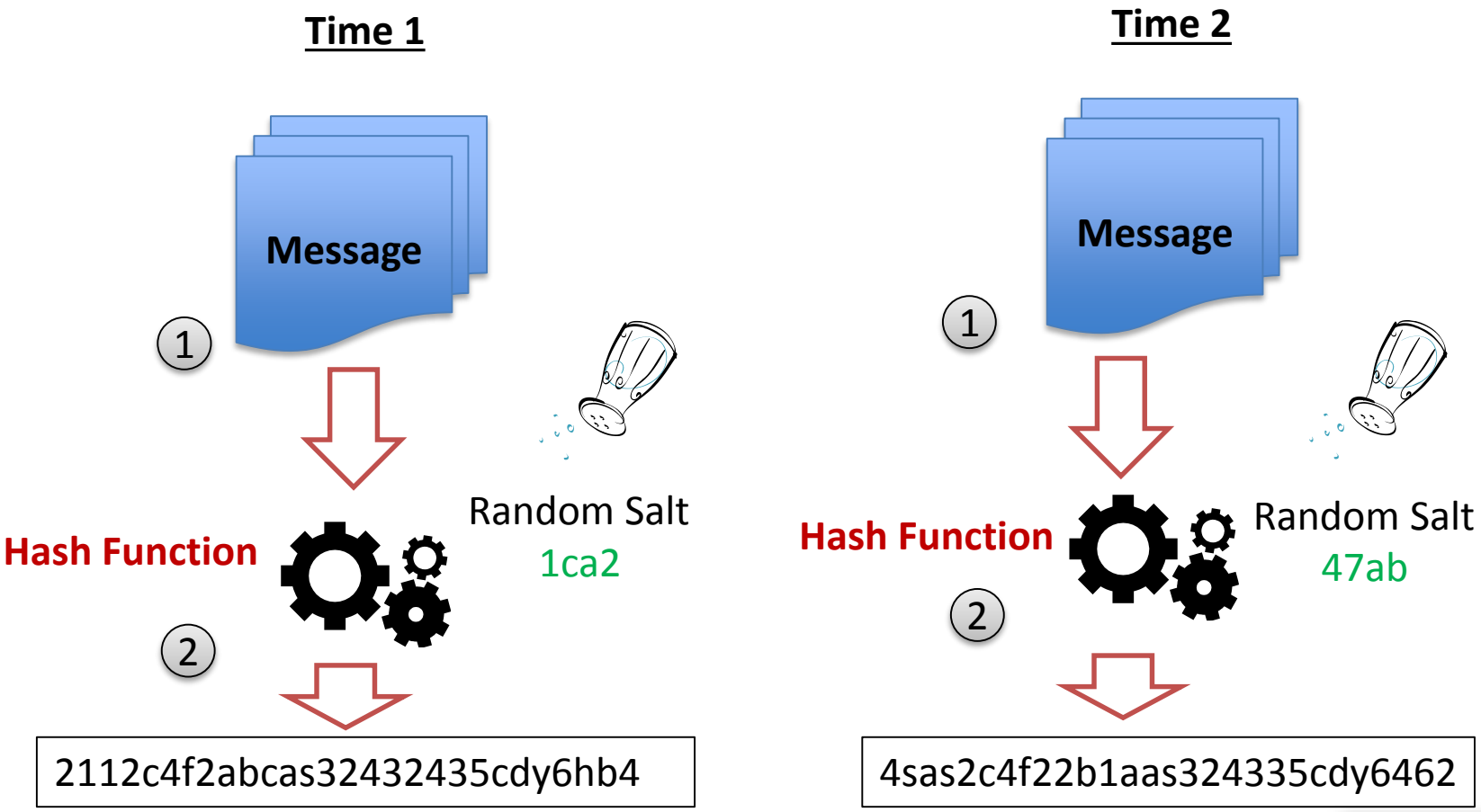
Dictionary Attacks

- Programs can be written to try out all common passwords against the login module.

- To protect against dictionary attacks, implement account lock algorithms once a certain number of failed login occurs.

Salted Hashing

A salt is a random data that is used as additional input to hash functions, to defend against dictionary attacks and rainbow table.
Hashing is often now using on password. If attacker knows the hashing function in use, he can potentially guess the password via brute force using a dictionary attack mechanism.



Code Dive Into Salted Hashing Implementation

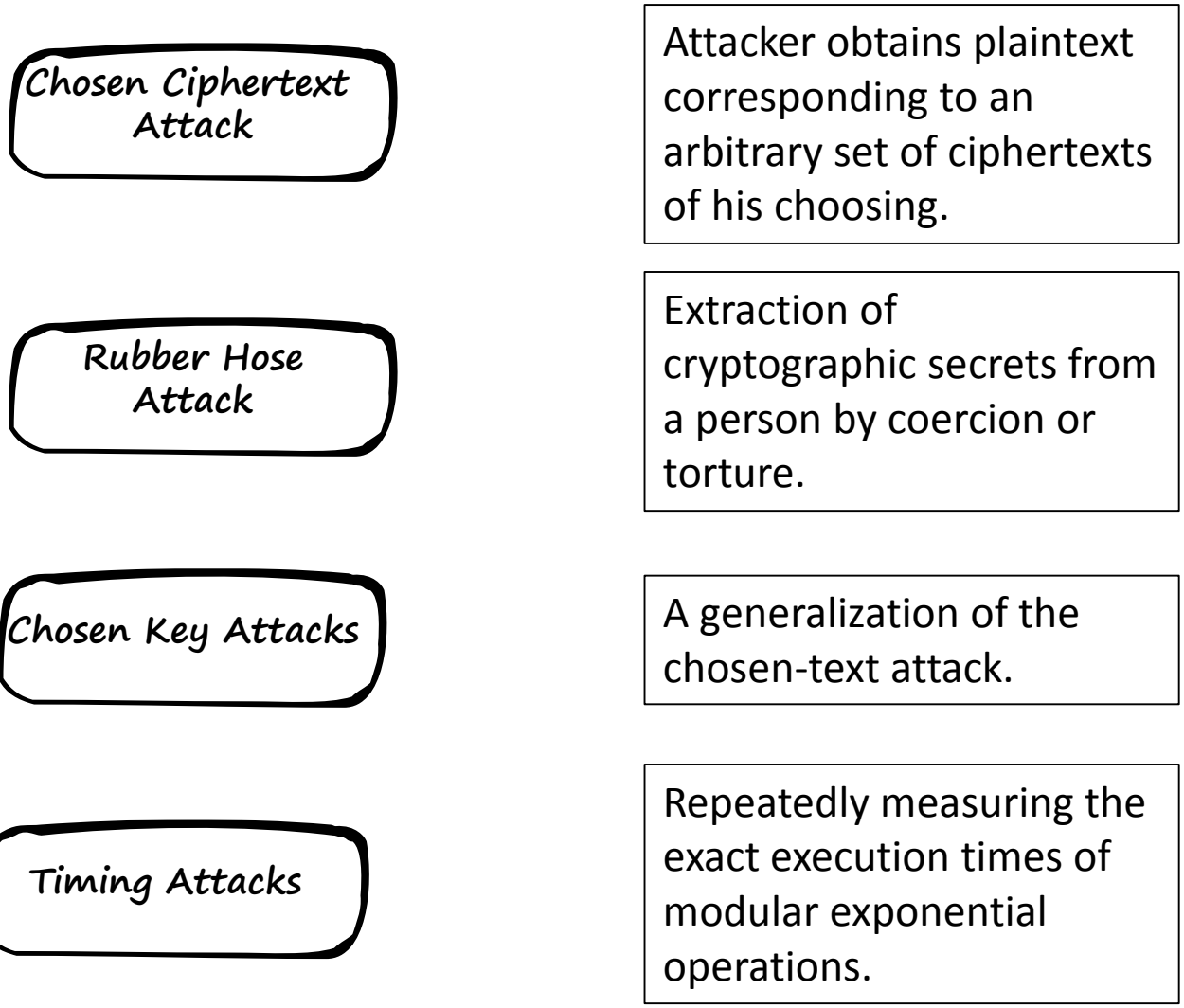
- Java Version

PasswordHash.java

- C# Version

PasswordHash.cs

Cryptography Attacks



Strength of Encryption And Key Length

Power /Cost	40 bits (5 char)	56 bits (7 char)	64 bits (8 char)	128 bits (16 char)
1 PC, $2K	1.4 min	73 days	50 years	10^20 years
Company, $100K	2 sec	35 hours	1 year	10^19 years
Huge Org, State, $1M	0.2 sec	3.5 hours	37 days	10^18 years

Length of time will decrease with Moore's Law when CPU power becomes cheaper and cheaper!

64 bits and below encryption are not secured enough! Can be broken.

Decryption Capabilities Today

NSA Building Quantum Computer that is capable of Cracking Code Real Fast

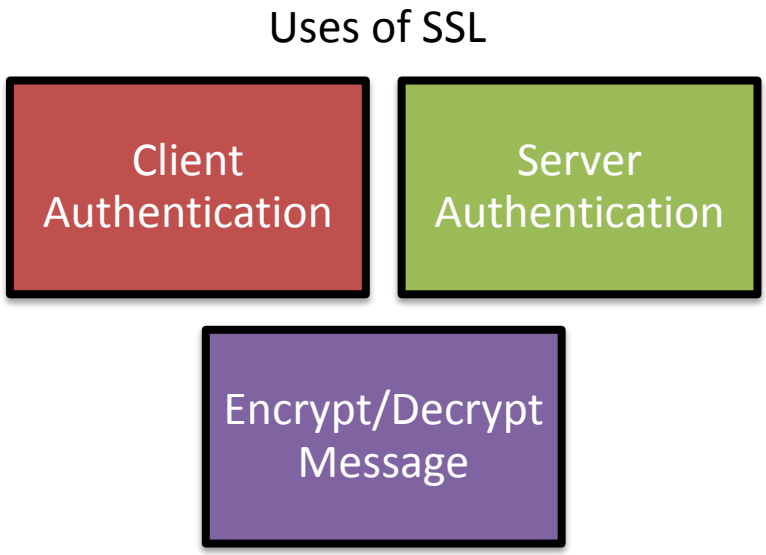
- http://www.youtube.com/watch?v=zmg4Q6l-sXk

Microsoft Banned and Acceptable Encryption Algorithms

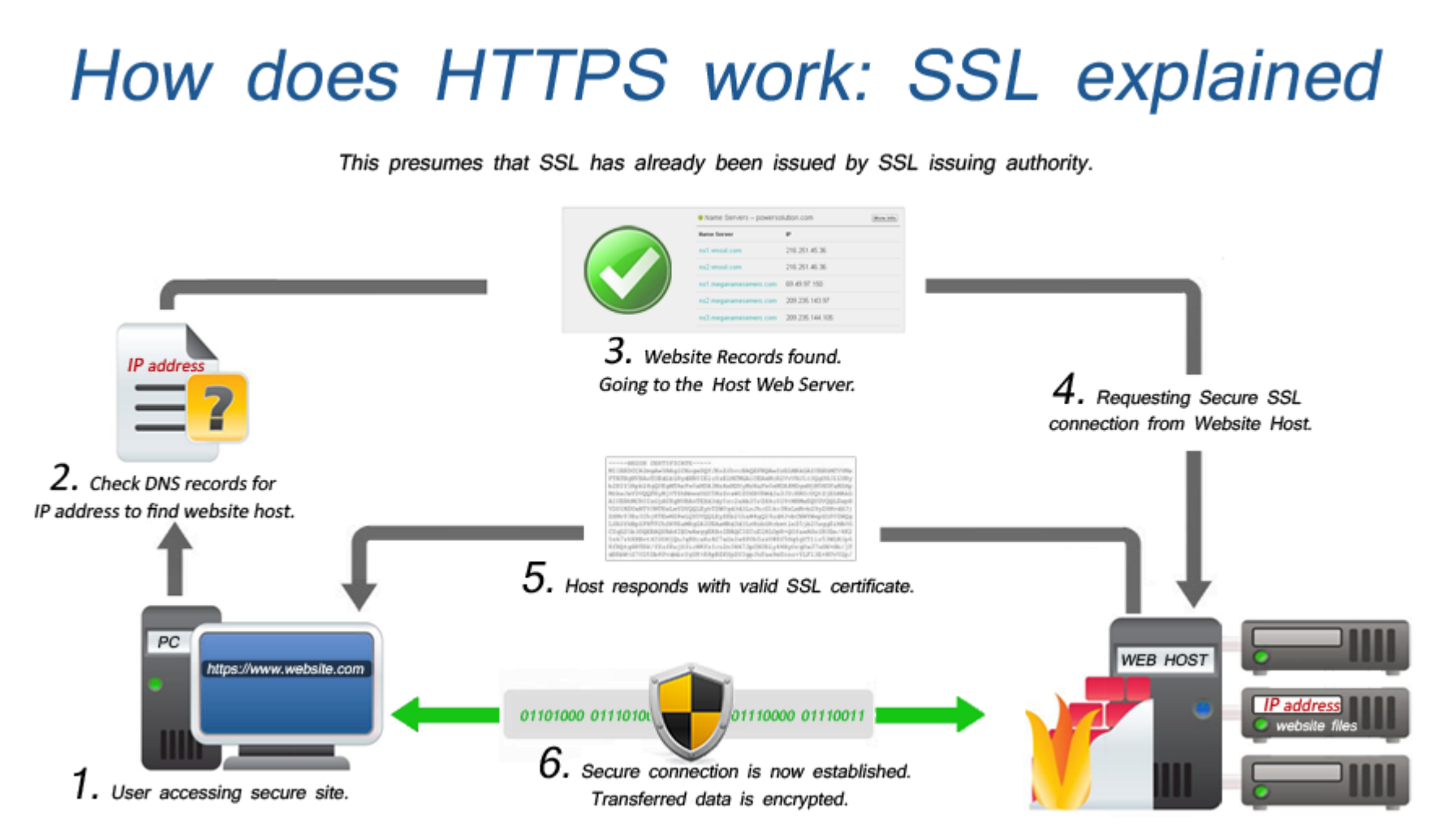
Type of Algorithm	Banned Algorithm	Acceptable Recommended Algorithm
Symmetric	DES, DESX, RC2, SKIPJACK, SEAL, CYLINK, MEK, RC4 (<128 bits)	3DES (2 or 3), RC4 (>=128 bits), AES
Asymmetric	RSA (<2048 bits), Diffie-Hellman(<2048 bits)	RSA (>=2048 bits), Diffie-Hellman (>=2048 bits), ECC (>=256 bits)
Hashing	SHA-0 (SHA), SHA-1, MD2, MD4, MD5	SHA-2 (SHA-256, SHA-384, SHA-512)

Secure Socket Layer

- Transport Layer Security
- Subsequently evolved into Internet Standard known as TLS (Transport Layer Security)
- Uses TCP to provide a reliable end to end service
- Often used together with HTTP traffic to provide transport layer security.



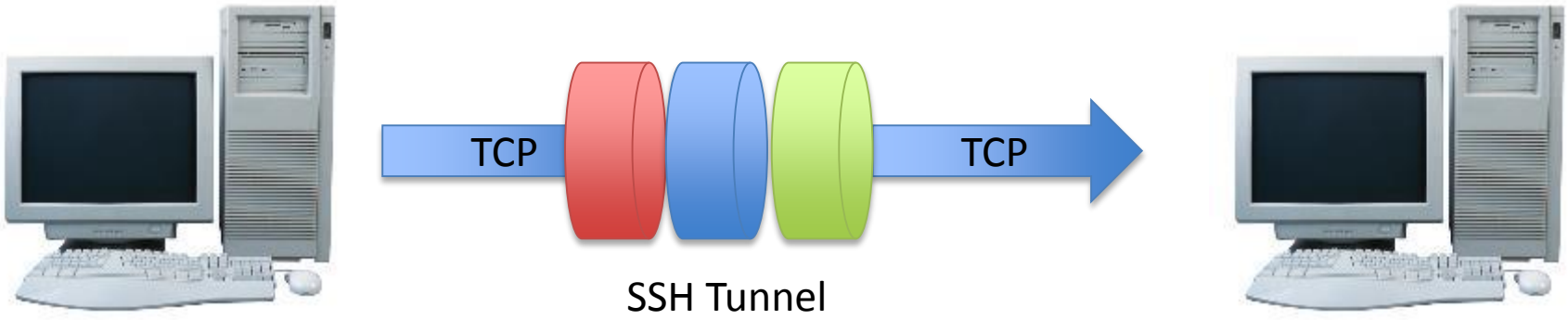
HTTPS Mechanism



Source: http://www.powersolution.com/wp-content/uploads/2013/04/SSL-flowchart.png

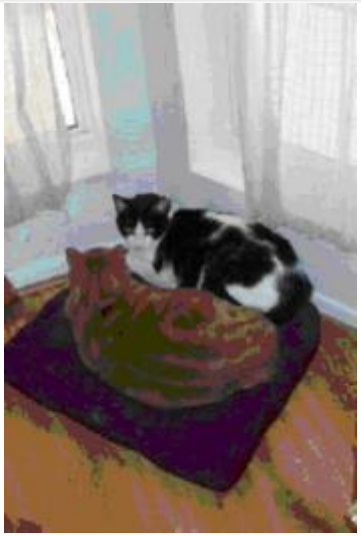
Secure Shell Channel

Remote Communication	Secure Channel	Strong Authentication
• SSH is a secure replacement for telnet, rcp, rlogin etc	• Provides encrypted channel for remote logging, command execute and file transfer.	• Can provide both strong host-to-host and user authentication.



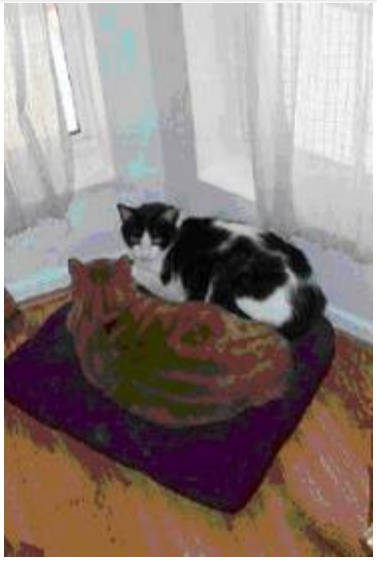
Steganography

Steganography is a sophisticated encryption technique to hide large amount of information within image and audio files.

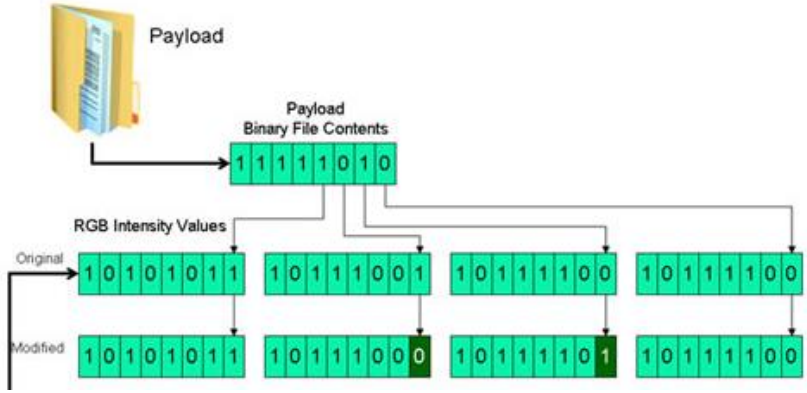


Picture without embedded text

Any visual differences?



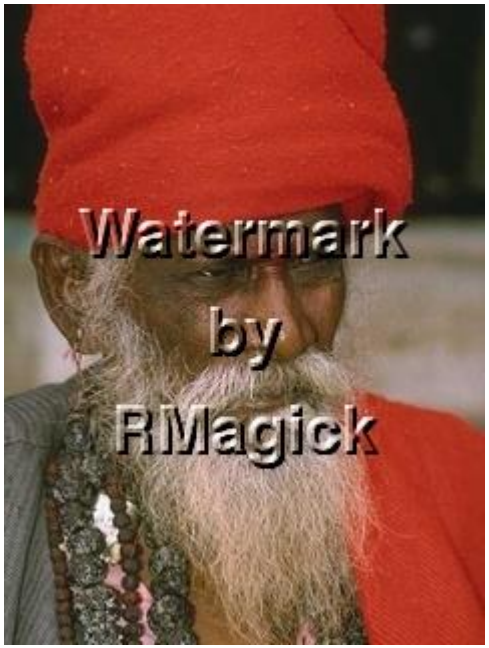
Picture with embedded text

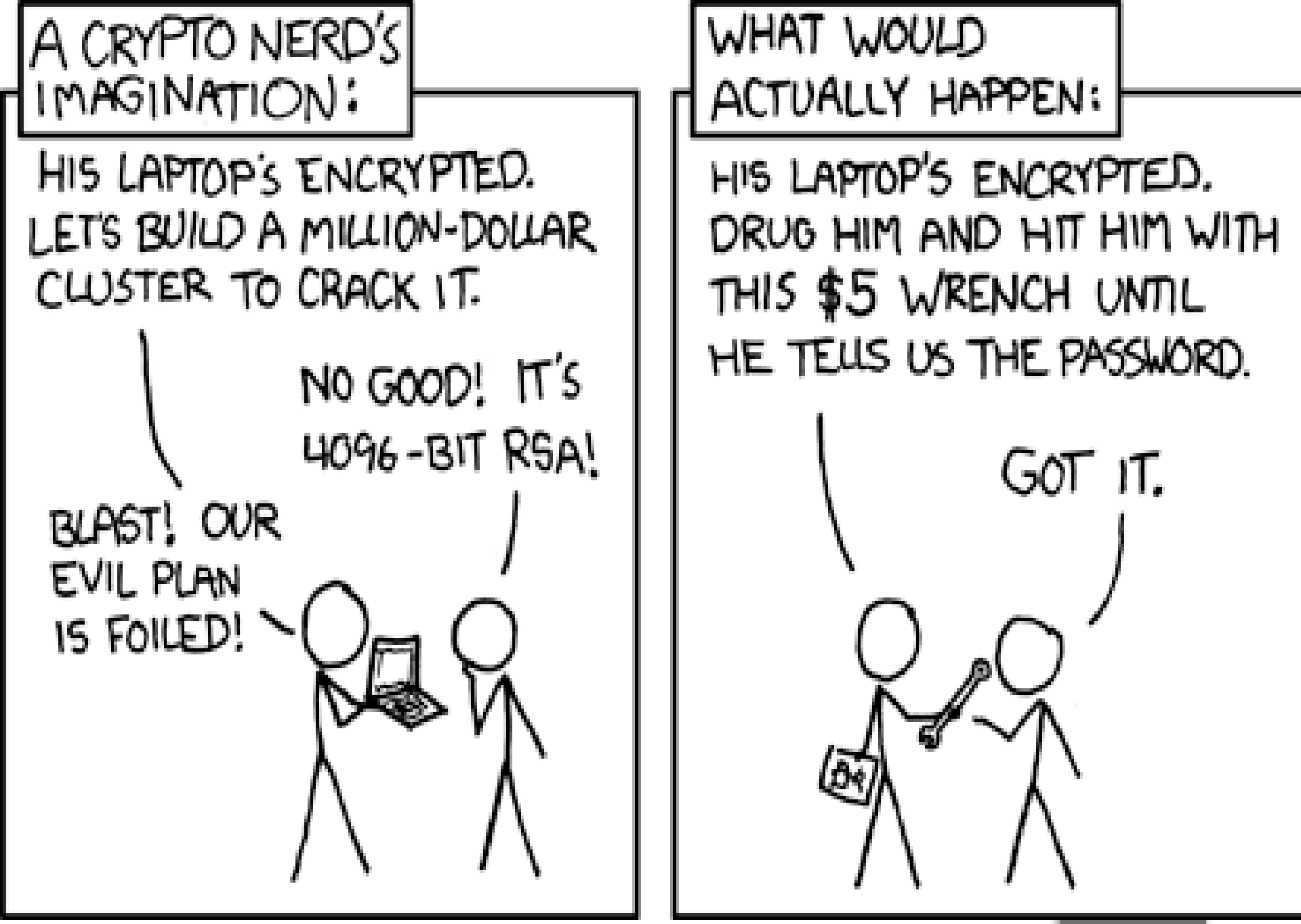


Slight changes in the Least Significant Bit of a pixel is almost not detectable using our naked eye.

Watermarking

Watermark is an intellectual property protection mechanism through distortion of original image to prevent people from using copyrighted content illegally for digital content. Existence of a watermark is obvious to the audience.





GROUP
ENTERPRISE

Wifi Security

Wifi Encryption Standards

Wired Equivalent Privacy(WEP)

- Old standards.
- 64 bit WEP uses a 40-bit key, 128 bit WEP uses 104 bit key, 256 bit WEP uses 232 bit. RC4 encryption.
- Considered insecure, can be cracked. Preshared keys are rarely changed. RC4.
- Weak Initialization Vector
- Developed without public review

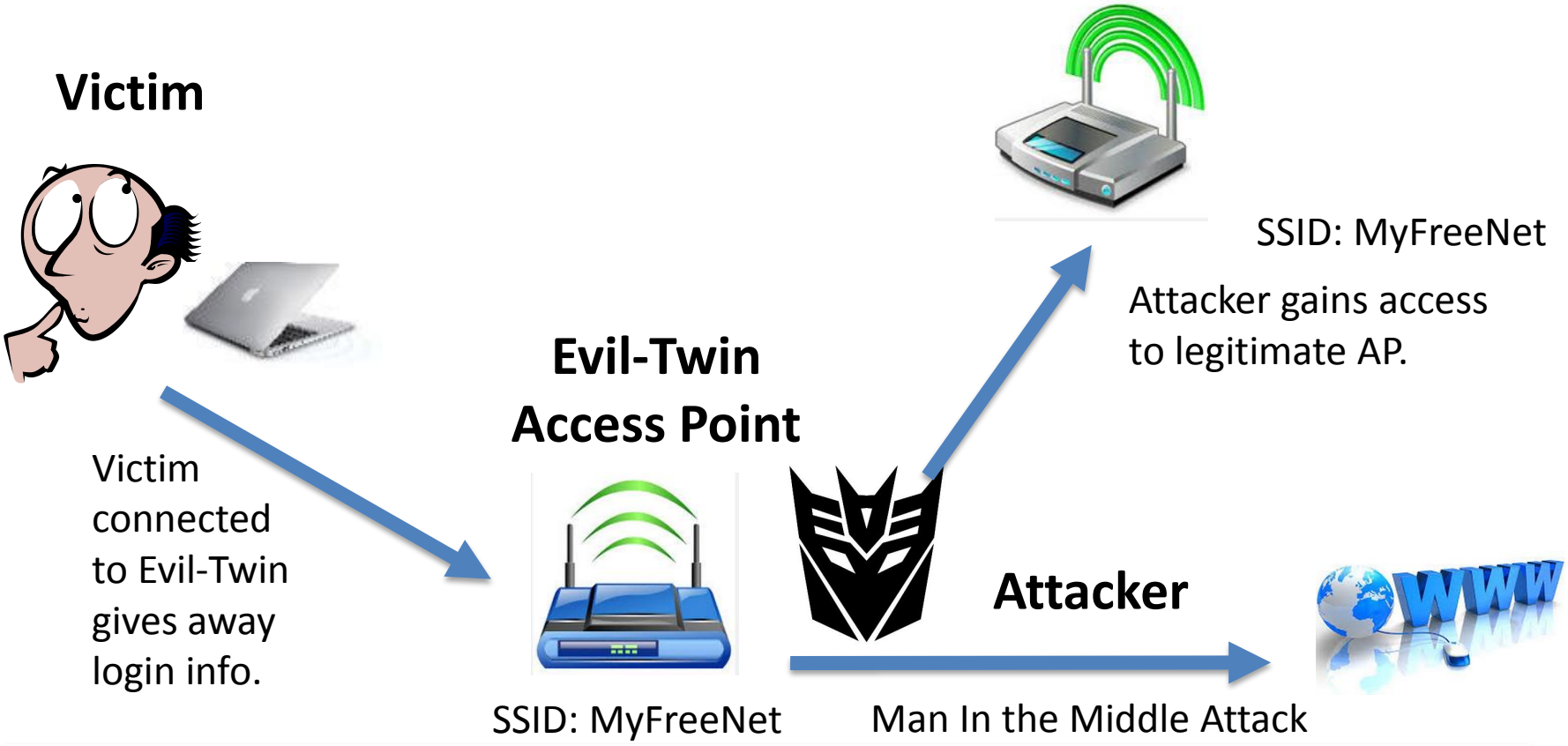
Wi-fi Protected Access (WPA)

- Temporal key Integrity Protocol (TKIP)
- Use 47 bit Initialization Vector (IV), 32-bit CRC

WPA-2

- Provides strong data protection and network access control
- Uses AES (128 bit) and Counter Cipher Mode with Block Chaining Message Authentication Code Protocol (CCMP) for wireless data encryption

Evil Twin (Wireless Network)



- Evil Twin is a wireless Access Point (AP) that pretends to be a legitimate AP by imitating another network name.
- Lures user to sign into the wrong AP.
- Can be used to steal passwords of unsuspecting users.

Wifi Security Measures

Change default SSID after WLAN configuration

Disable SSID Broadcast

Enable MAC Address filtering for high secure network

Use WPA2 or WPA Wifi Standards

GROUP
ENTERPRISE

Regular Expression

Regular Expression

- Regular Expression is often used as a means to perform validation (both input and output).
- java.util.regex.Pattern

```
Example validating the parameter “zip” using a regular expression.

private static final Pattern zipPattern = Pattern.compile("^\d{5}(-\d{4})?$");

public void doPost( HttpServletRequest request, HttpServletResponse response)
{
              try { String zipCode = request.getParameter( "zip" );
              if ( !zipPattern.matcher( zipCode ).matches()
              {
                            throw new YourValidationException( "Improper zipcode format." );
              } .. do what you want here, after its been validated .. }
              catch(YourValidationException e )
              {
                            response.sendError( response.SC_BAD_REQUEST, e.getMessage() );
              }
}
```

Regular Expression Syntax

Regular Expressions Anchors	
^	Start of string, or start of line in multi-line pattern
\A	Start of string
$	End of string, or end of line in multi-line pattern
\Z	End of string
\b	Word boundary
\B	Not word boundary
\<	Start of word
\>	End of word

Regular Expressions Character Classes	
\c	Control character
\s	White space
\S	Not white space
\d	Digit
\D	Not digit
\w	Word
\W	Not word
\x	Hexadecimal digit
\O	Octal digit

Regular Expressions POSIX	
[:upper:]	Upper case letters
[:lower:]	Lower case letters
[:alpha:]	All letters
[:alnum:]	Digits and letters
[:digit:]	Digits
[:xdigit:]	Hexadecimal digits
[:punct:]	Punctuation
[:blank:]	Space and tab
[:space:]	Blank characters
[:cntrl:]	Control characters
[:graph:]	Printed characters
[:print:]	Printed characters and spaces
[:word:]	Digits, letters and underscore

May have to enter double \\ in some programming languages,
Since \ is an escape character. “\d” => “\\d”

Regular Expression Syntax

Regular Expressions Assertions	
?=	Lookahead assertion
?!	Negative lookahead
?<=	Lookbehind assertion
?!= or ?<!	Negative lookbehind
?>	Once-only Subexpression
?()	Condition [if then]
?()|	Condition [if then else]
?#	Comment

Regular Expressions Escape Sequences	
\	Escape following character
\Q	Begin literal sequence
\E	End literal sequence

"Escaping" is a way of treating characters which have a special meaning in regular expressions literally, rather than as special characters.

Regular Expressions Quantifiers	
*	0 or more
+	1 or more
?	0 or 1
{3}	Exactly 3
{3,}	3 or more
{3,5}	3, 4 or 5

Add a ? to a quantifier to make it ungreedy.

Regular Expression Common Metacharacters		
^	[	.
$	{	*
(	\	+
)	|	?
<	>	

The escape character is usually the backslash - \.

Regular Expression Syntax

Regular Expressions Special Characters	
\n	New line
\r	Carriage return
\t	Tab
\v	Vertical tab
\f	Form feed
\xxx	Octal character xxx
\xhh	Hex character hh

Regular Expressions Groups and Ranges	
.	Any character except new line (\n)
(a|b)	a or b
(...)	Group
(?:...)	Passive (non-capturing) group
[abc]	Range (a or b or c)
[^abc]	Not a or b or c
[a-q]	Letter from a to q
[A-Q]	Upper case letter from A to Q
[0-7]	Digit from 0 to 7
\n	nth group/subpattern
Ranges are inclusive.	

Regular Expressions Pattern Modifiers	
g	Global match
i	Case-insensitive
m	Multiple lines
s	Treat string as single line
x	Allow comments and white space in pattern
e	Evaluate replacement
U	Ungreedy pattern

Regular Expressions String Replacement	
$n	nth non-passive group
$2	"xyz" in /^(abc(xyz))$/
$1	"xyz" in /^(?:abc)(xyz)$/
$`	Before matched string
$'	After matched string
$+	Last matched string
$&	Entire matched string
Some regex implementations use \ instead of $.	

Regular Expression Video

Let’s spend 10 minutes watching this quick regular expression introduction.

- http://www.youtube.com/watch?v=DRR9fOXkfRE
- http://www.youtube.com/watch?v=s_PfopWcMwI

Common Validation Regular Expression

- http://regexlib.com/ is a common site with quite a number of regular expressions that is already defined for consideration.

Special Field	Regular Expression	Matches	Do not Match
SG NRIC Number	^[SFTG]\d{7}[A-Z]$	S9912345A T1234567V F0094839P	K4928940829F T1234567 M1234567C
SG Land Phone Line	^[6]\d{7}$	61234567 63829324 67654321	6123-4567 6-CALL-CPY 6123abcd
Visa Credit Card	(^4\d{12}$)\|(^4[0-8]\d{14}$)|(^(49)[^013]\d{13}$)\|(^(49030)[0-1]\d{10}$)|(^(49033)[0-4]\d{10}$)\|(^(49110)[^12]\d{10}$)|(^(49117)[0-3]\d{10}$)\|(^(49118)[^0-2]\d{10}$)|(^(493)[^6]\d{12}$)	4111111111111111 4903020000000008	

Simple Java Regular Expression Code

```
import java.util.regex.Matcher;
import java.util.regex.Pattern;

public class RegExMatcher
{
      final String PHONE_REG_EXP="^[6]\\d{7}$";

      public boolean isValidLandPhone(String phoneNumber)
      {
            Pattern pattern = Pattern.compile(PHONE_REG_EXP);
            Matcher matcher = pattern.matcher(phoneNumber);
            return matcher.matches();
      }


}
```

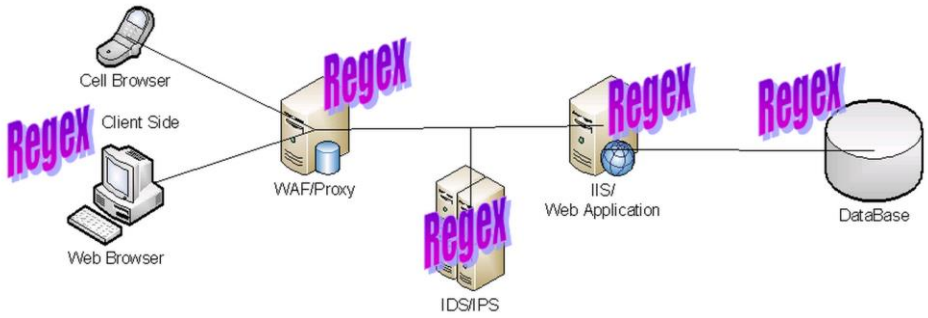
Regular Expression Exercise (20 mins)

- Split into groups of 2 to write the regular expression for these fields. Test out your answers on regexlib.com

1. Email Address
2. SG Postal Code
3. Valid numeric months (01, 08, 12, but not 15)
4. Valid numeric days (01, 25, 28, but not 35)
5. Valid birthdates for people registering born between 1965 to today in dd/mm/yyyy format.
6. IPv4
7. IPv6
8. IPv4 or IPv6

ReDoS

- Regular Expression Denial of Service (ReDos) is a denial of service attack that exploits regular expression implementation reaching extreme situations that cause them to work very slowly. An attacker can then cause a program using a regular expression to do validation and hang them for a long time.



Picture Source: OWASP

Vulnerable Regex 1: Email

```
^([a-zA-Z0-9])(([\-.]|[_]+)?([a-zA-Z0-9]+))*(@){1}[a-z0-9]+[.]{1}(([a-z]{2,3})|([a-z]{2,3}[.]{1}[a-z]{2,3}))$
```

Problem Input 1: Email

```
aaaaaaaaaaaaaaaaaaaaaaaa!
```

Vulnerable Regex 2: Username

```
^(([a-z])+.)+[A-Z]([a-z])+$
```

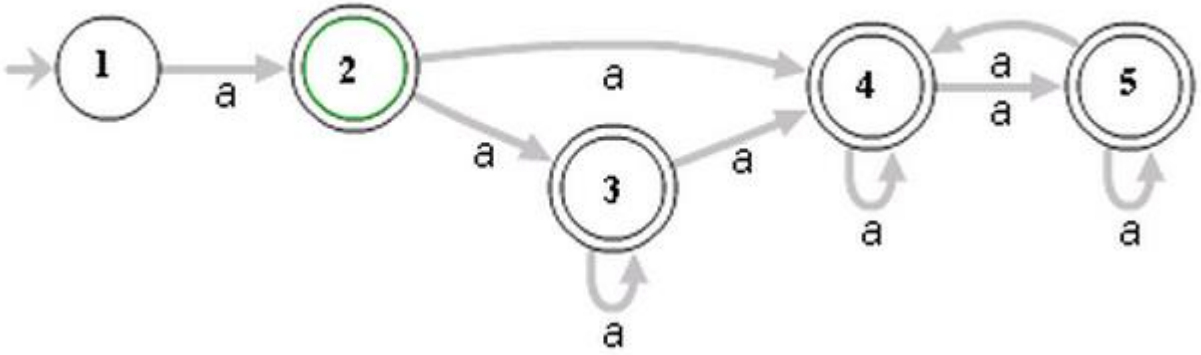
Problem Input 2: Username

```
aaaaaaaaaaaaaaaaaaaaaaaa!
```

Avoiding ReDoS

- Careful on what is being used for regular expression.
- Do not accept Regular Expression pattern to be part of the input from the user.
- Many Regular Expression uses a NFA (Nondeterministic Finite Automaton) internally. Many of the search pass can open up a lot of various possibility of potential matches.

For example, the Regex ***^(a+)+$*** is represented by the following NFA:



Evil Regexes

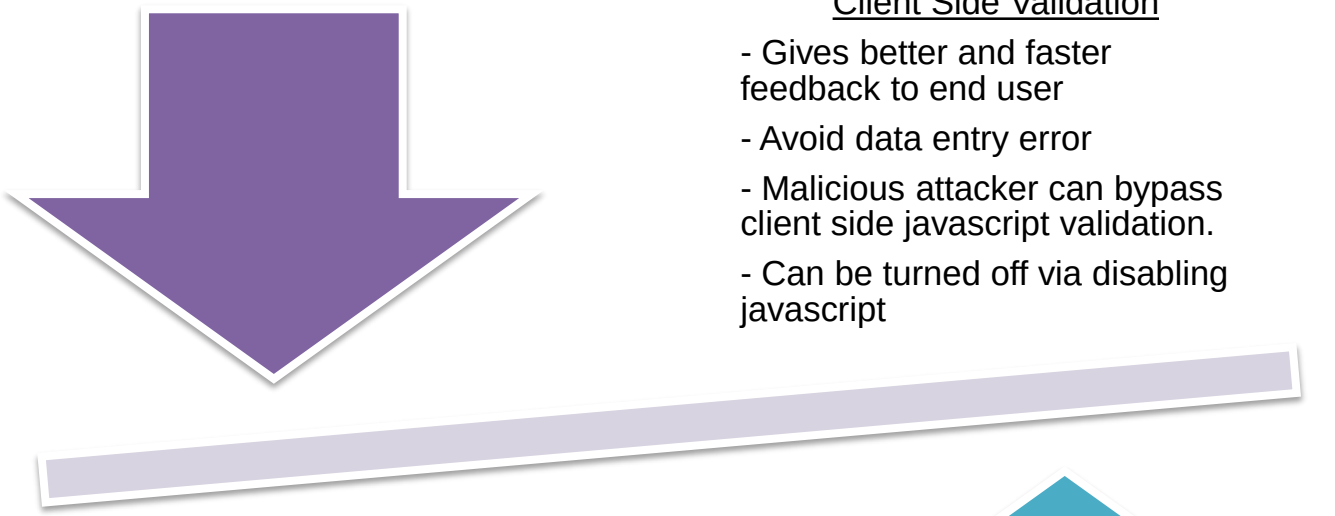
- Evil Regex contains:
  - Grouping with repetition
  - Inside repeated group:
    - Repetition
    - Alternation with overlapping
- More examples of Evil Regex
  - (a+)+
  - ([a-zA-Z]+)*
  - (a|aa)+
  - (a|a?)+
  - (.*a){x} | for x > 10

Show Demo of this slowness in action inside Eclipse.

GROUP
ENTERPRISE

Validation Strategies

Client Side Versus Server Side Validation



Client Side Validation

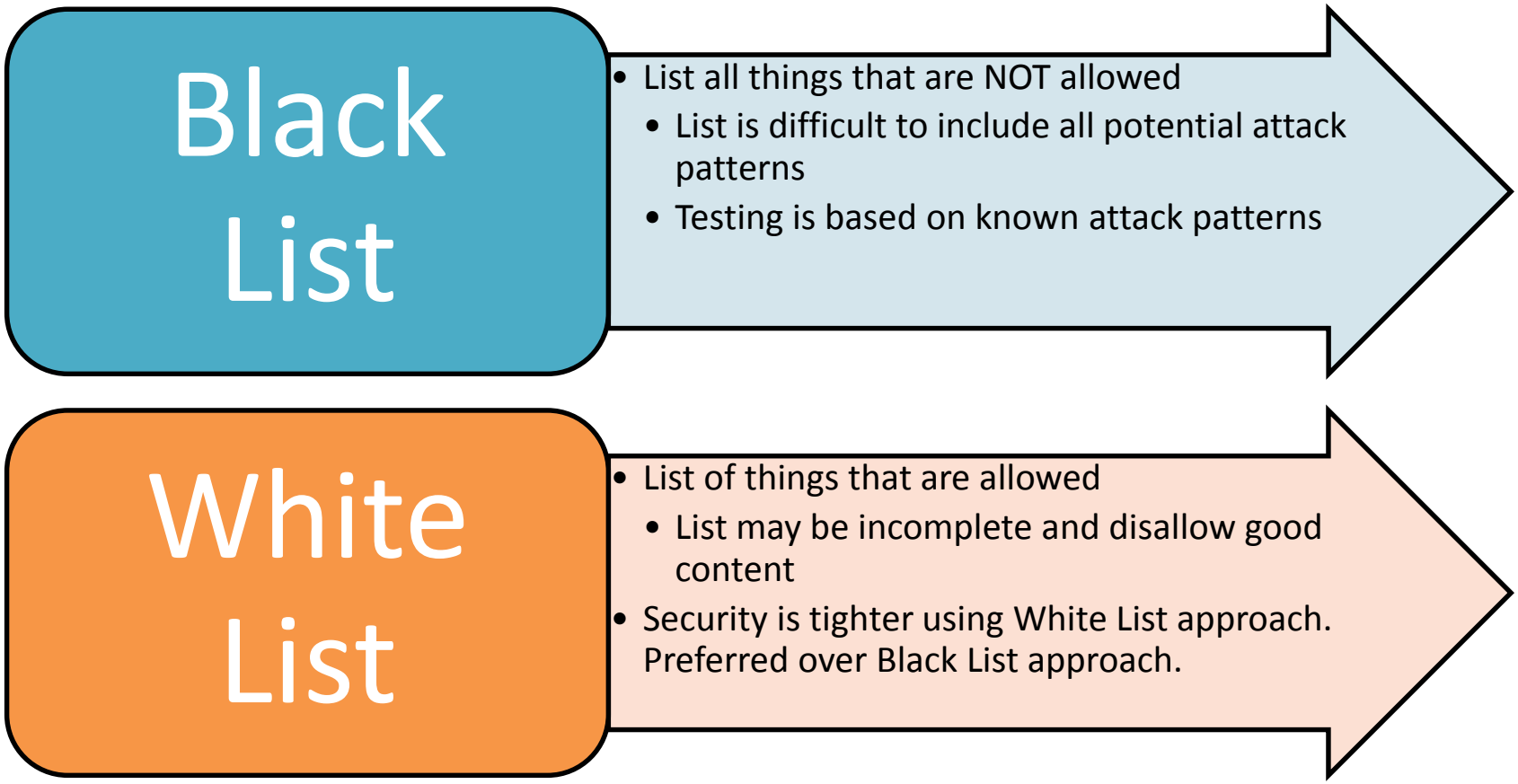
- Gives better and faster feedback to end user
- Avoid data entry error
- Malicious attacker can bypass client side javascript validation.
- Can be turned off via disabling javascript

Server Side Validation

- Protect against malicious user
- Requires round trip, takes longer time
- Will always be validated in order for transaction to go through

The reality is: Both type of validations are required in the design of a web application.

Validation Strategies



Black List and White List can be used together to create the desired results.

Input Validation

- Principle problem
  - Location of Check ≠ Location of Use
- Principle solution
  - Canonicalization of input
    - Transform input into a canonical form
  - Decision is made on input in the same form that program uses

Canonicalization

Canonicalization is the process of converting data that has more than one possible representation to confirm to a standard canonical form.

- Two major program errors:
  - Misunderstanding definition of canonical form
  - Stopping canonicalization process to early

Alternate Forms	http://crackzone.com http://www%2ecrackzone%2ecom http://208.87.33.151
Canonical Form	http://www.crackzone.com

Sanitization

Sanitization is the process of converting something that is considered dangerous into its innocuous form.

Stripping

- Removing harmful characters from user supplied input

(65)-9422-1242

65\";<DROP TABLE USER;'94221242

Strip non numeric values

6594221242

Substitution

- Replacing user supplied input with other alternatives

'65-94221242'

Substitute with safer forms. ' is an escape character in SQL.

"65-94221242"

Masking

- Replacing certain fields with a Mask character.

65-94221242

Telephone number masking

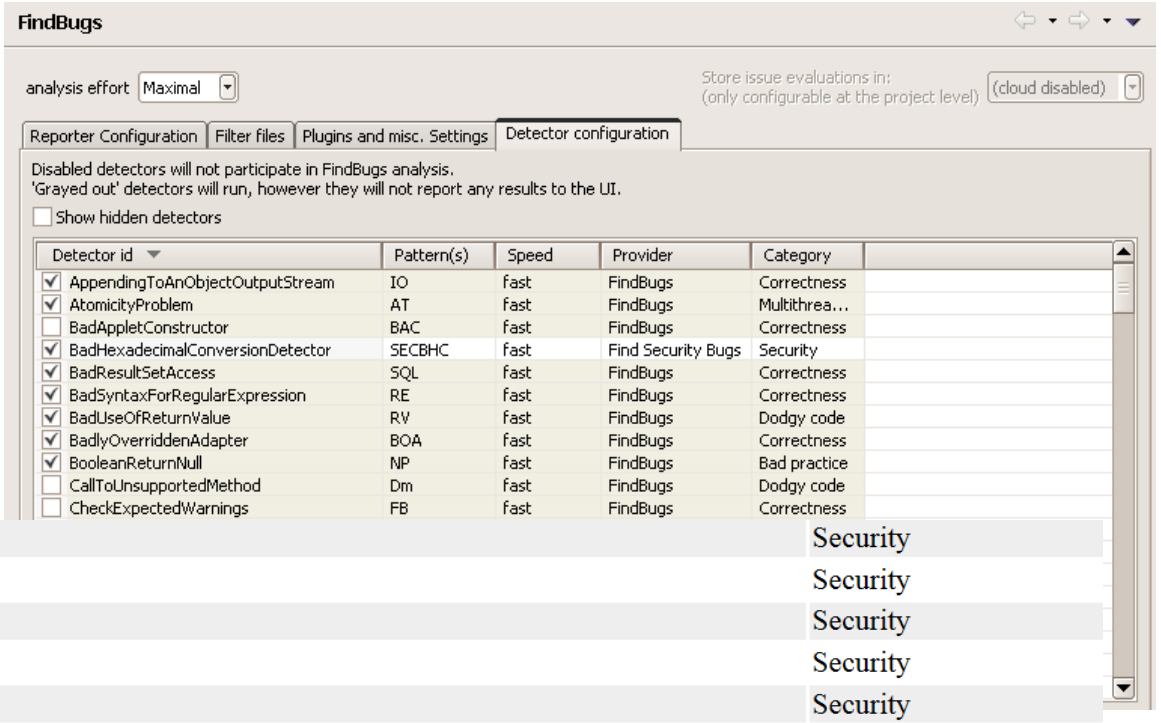
65-9XXX1242

GROUP
ENTERPRISE

Static Secure Code Analyzer

FindBug Security Plugin

- http://h3xstream.github.io/find-sec-bugs/
- http://findbugs.sourceforge.net/bugDescriptions.html



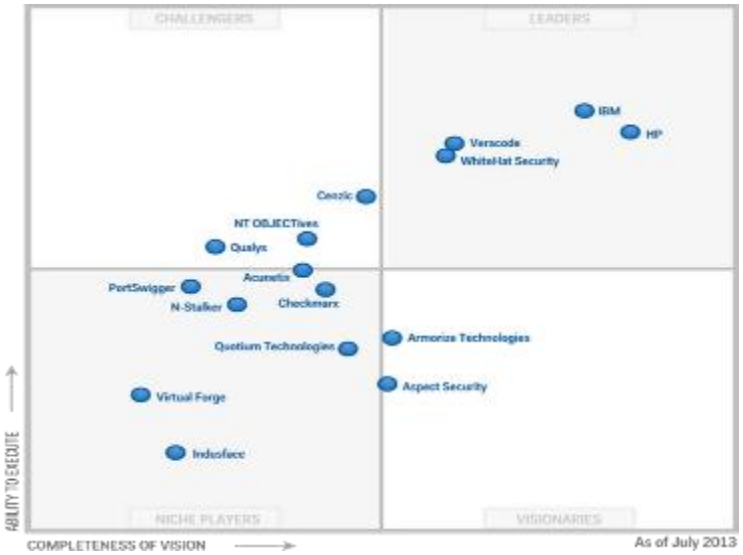
Dm: Hardcoded constant database password	Security
Dm: Empty database password	Security
HRS: HTTP cookie formed from untrusted input	Security
HRS: HTTP Response splitting vulnerability	Security
PT: Absolute path traversal in servlet	Security
PT: Relative path traversal in servlet	Security
SQL: Nonconstant string passed to execute method on an SQL statement	Security
SQL: A prepared statement is generated from a nonconstant String	Security
XSS: JSP reflected cross site scripting vulnerability	Security
XSS: Servlet reflected cross site scripting vulnerability in error page	Security
XSS: Servlet reflected cross site scripting vulnerability	Security

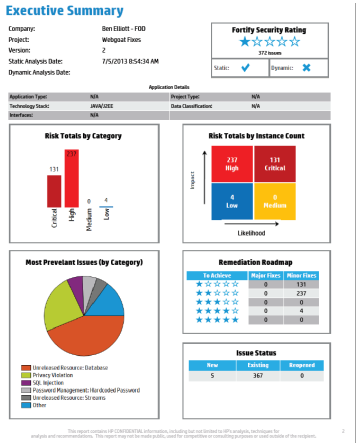
Magic Quadrant for Static Application Security Testing

- http://www.gartner.com/technology/reprints.do?id=1-1GTXLFB&ct=130703&st=sb

HP Fortify SCA (Utilized in Quality Gates)

HP Fortify SCA is a security focused code static analyzer that is categorized in the top leader quadrant in the 2013 Gartner Magic Quadrant for Application Security Testing. It maps closely to OWASP vulnerabilities listing.

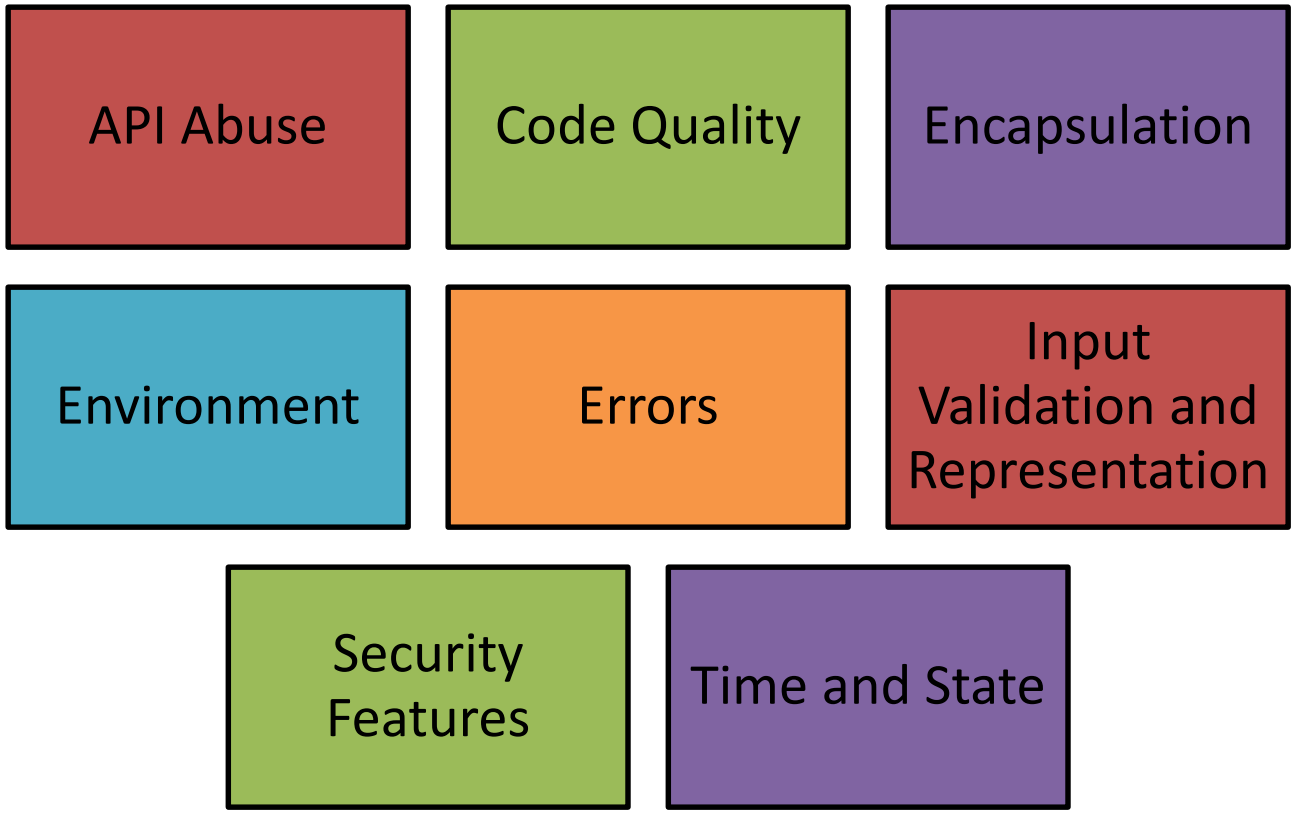




Supported Language	Versions
ABAP/BSP	6
ActionScript/MXML (Flex)	3, 4
ASP.NET, VB.NET, C# (.NET)	1.1, 2.0, 3.0, 3.5, 4.0
Classic ASP (with VBScript)	2, 3
COBOL	IBM Enterprise Cobol for z/OS 3.4.1 with IMS, DB2, CICS, MQ
CFML	5, 7, 8
HTML	4 and earlier
Java	1.3, 1.4, 1.5, 1.6, 1.7
JavaScript/AJAX	1.7
JSP	1.2, 2.1
PHP	5.0 – 5.2
PL/SQL	8.1.6
Python	2.6
T-SQL	SQL Server 2005 and 2008
Visual Basic	6
VBScript	2.0, 5.0
XML	1.0
Objective-C	XCode 4.6 and iOS 6.1

FORTIFY® SOFTWARE

Fortify SCA


API Abuse
Code Quality
Encapsulation
Environment
Errors
Input Validation and Representation
Security Features
Time and State

Parasoft JTest / dotTest (Utilized in Quality Gates)

Parasoft dotTest is a comprehensive static code check tools covering a wide area of coding best practices. Contains 400+ rules that cover Microsoft .NET framework design guidelines, CLS Compliance, Object Oriented Metrics, Security.

Covers

- static code analysis
- data flow analysis
- metric analysis

Security Rules Checked by Parasoft includes:

- Potential Injection Attacks
- Potential XSS vulnerability
- Presence of Sanitization of Input Activities
- Proper Output Encoding
- Proper Threading Usage
- Correct Exception Handling
- Proper Resource Handling
- Usage of Secure Cryptographic Algorithms

Utilization of multiple commercial grade code scanners to increase coverage of potential issues detected.

Parasoft Security Rules

- **Security [SECURITY]**
  - **Backdoor Vulnerabilities [SECURITY.BV]**
    - Do not access the class loader in a web component [SECURITY.BV.ACL-3]
    - Inspect usage of 'Date', 'Time' objects and 'System.currentTimeMillis()' method invocations [SECURITY.BV.ADT-5]
    - Inspect usage of 'getName()' from 'java.lang.Class' object [SECURITY.BV.AUG-5]
    - Use "read-only" AccessMode for Castor queries [SECURITY.BV.CQRO-4]
    - Do not set custom security managers outside of the 'main' method [SECURITY.BV.DSSM-2]
    - Ensure all sensitive method invocations are logged [SECURITY.BV.ENFL-2]
    - Do not call 'Socket.setSocketImplFactory()' or 'URL.setURLStreamHandlerFactory()' in a web component [SECURITY.BV.NSF-3]
    - Wrap "privileged" method invocations in "final" methods [SECURITY.BV.PCFM-4]
    - Wrap "privileged" method invocations in "private" methods [SECURITY.BV.PCPM-4]
    - Avoid using dynamically loaded classes in "privileged" code blocks [SECURITY.BV.PDLC-3]
    - Do not access or set System properties [SECURITY.BV.SYSP-2]
  - **Deadlocks and Race Conditions [SECURITY.DRC]**
    - Do not use threads in web components [SECURITY.DRC.THR-3]
  - **Erratic Application Behavior [SECURITY.EAB]**
    - Avoid calling specified methods from web components and EJBs [SECURITY.EAB.ACWC-4]
    - Do not use AWT classes in Web components [SECURITY.EAB.AWT-3]
    - Do not pass byte arrays to ObjectOutputStream in the 'writeObject()' method [SECURITY.EAB.CBA-2]
    - Do not compare Class objects by name [SECURITY.EAB.CMP-2]
    - Enforce returning a defensive copy in 'clone()' methods [SECURITY.EAB.CPCL-3]
    - Do not stop the JVM in a web component [SECURITY.EAB.JVM-3]
    - Limit the number of "AccessController.doPrivileged" calls per class [SECURITY.EAB.LDP-4]
    - Do not pass user-given mutable objects directly to certain types [SECURITY.EAB.MPT-3]
    - Implement 'readObject()' and 'writeObject()' for all 'Serializable' classes [SECURITY.EAB.OROM-5]
    - Limit the number of lines in "privileged" code blocks [SECURITY.EAB.PCL-4]
    - Do not declare "static" fields in web components [SECURITY.EAB.SF-3]
    - Do not change the input streams of 'java.lang.System' in a web component [SECURITY.EAB.SIS-3]
    - Do not store user-given mutable objects directly into variables [SECURITY.EAB.SMO-3]
    - Inspect 'static' fields which may have intended to be declared 'static final' [SECURITY.EAB.SPFF-4]

Parasoft Security Rules

- **Exposing Sensitive Data [SECURITY.ESD]**
  - Avoid writing to Consoles [SECURITY.ESD.ACW-5]
  - Do not log confidential or sensitive information [SECURITY.ESD.CONSEN-5]
  - Clear sensitive data after use [SECURITY.ESD.CSD-3]
  - Do not pass exception messages into output in order to prevent the application from leaking sensitive information [SECURITY.ESD.PEO-2]
  - Avoid storing sensitive data in plaintext in a cookie [SECURITY.ESD.PLC-3]
  - Avoid methods that might expose internal representations by returning arrays or other mutable fields [SECURITY.ESD.RA-3]
  - Store sensitive data in mutable objects [SECURITY.ESD.SDM-2]
  - Inspect instance fields of serializable objects to make sure they will not expose sensitive information [SECURITY.ESD.SIF-3]
  - Avoid calling print methods of 'System.err' or 'System.out' [SECURITY.ESD.SIO-5]
  - Do not expose data with a 'FileNotFound' exception [SECURITY.ESD.SNFD-3]
  - Do not interrogate or modify security policy information in a web component [SECURITY.ESD.SPI-3]
  - Declare "transient" fields "private" [SECURITY.ESD.TFP-3]
  - Avoid "transient" fields in serialPersistentFields array [SECURITY.ESD.TSPF-3]
  - Use 'post' instead of 'get' for credential transfers [SECURITY.ESD.UPCT-2]
- **Input-Based Attacks [SECURITY.IBA]**
  - Do not extend from the Struts classes 'ActionForm' and 'DynaActionForm' [SECURITY.IBA.AEAF-3]
  - Avoid temporary files [SECURITY.IBA.ATF-3]
  - Avoid using "SELECT *" in SQL queries [SECURITY.IBA.AUSS-2]
  - Canonicalize all data before validation [SECURITY.IBA.CDBV-3]
  - Always call 'super.validate()' from validation methods in 'ActionForm' classes [SECURITY.IBA.CSVFV-3]
  - Use wrapper methods to secure native methods [SECURITY.IBA.NATIW-3]
  - Use 'prepareCall' or 'prepareStatement' instead of 'createStatement' [SECURITY.IBA.UPS-3]
  - Encapsulate all dangerous data returning methods with a validation function [SECURITY.IBA.VPPD-2]
  - Encapsulate all redirect and forward URLs with a validation function [SECURITY.IBA.VRD-1]
  - Avoid XPath injection when evaluating XPath queries [SECURITY.IBA.XPIJ-2]

Parasoft Security Rules

- **Unsafe Environment Configuration [SECURITY.UEC]**
  - Ensure that 'axis.development.system' is set to "false" in Axis 'server-config.wsdd' files [SECURITY.UEC.ADS-3]
  - Ensure that 'axis.enableListQuery' is set to "false" in Axis 'server-config.wsdd' files [SECURITY.UEC.AELQ-3]
  - Ensure that 'axis.disableServiceList' is set to "true" in Axis 'server-config.wsdd' files [SECURITY.UEC.DSL-3]
  - Avoid defining multiple security roles with the same name in 'web.xml' files [SECURITY.UEC.DSR-2]
  - Ensure that the 'Encrypt' directive is specified for each 'items' tag in Axis2 configuration files [SECURITY.UEC.EDAR-3]
  - Ensure that each filter mapped in a 'web.xml' file has a corresponding definition [SECURITY.UEC.FMCD-3]
  - Use 'https' instead of 'http' for the 'transportReceiver' and 'transportSender' in 'axis2.xml' configuration files [SECURITY.UEC.HTTPS-5]
  - Ensure that 'InflowSecurity' and 'OutflowSecurity' parameters are specified in Axis2 configuration files [SECURITY.UEC.ISOS-3]
  - Include an appropriate '<login-config>' element to specify the type of authentication to be performed in 'web.xml' files [SECURITY.UEC.LCA-3]
  - Avoid storing usernames and passwords in plain text in Castor 'jdo-conf.xml' files [SECURITY.UEC.PCCF-1]
  - Avoid using plain text passwords in Axis 'wsdd' files [SECURITY.UEC.PTPT-3]
  - Ensure that passwords are not stored in plain text and are sufficiently long [SECURITY.UEC.PWD-1]
  - Restrict access to JSPs in 'web.xml' files by including a security constraint for '*.jsp' files [SECURITY.UEC.RAJ-3]
  - Ensure that "REST" is disabled in 'axis2.xml' configuration files [SECURITY.UEC.REST-5]
  - Ensure that the 'Signature' directive is specified for each 'items' tag in Axis2 configuration files [SECURITY.UEC.SDAR-3]
  - Always specify error pages in web.xml [SECURITY.UEC.SEP-3]
  - Session identifiers should be at least 128 bits long to prevent brute-force session guessing [SECURITY.UEC.SLID-3]
  - Avoid using the SOAP Monitor module [SECURITY.UEC.SMM-3]

- Ensure that each security role referenced in a 'web.xml' file has a corresponding definition [SECURITY.UEC.SRCD-3]
- Ensure that sessions are configured to time out within a reasonable amount of time in 'web.xml' files [SECURITY.UEC.STTL-3]
- Ensure that the 'Timestamp' directive is specified for each 'items' tag in Axis2 configuration files [SECURITY.UEC.TDAR-3]
- Ensure that all constrained resources are protected with a '<user-data-constraint>' element in 'web.xml' files [SECURITY.UEC.UDC-3]
- Avoid using plain text passwords in Axis2 configuration files [SECURITY.UEC.UTAX-3]
- Ensure SOAP messages are encrypted in WebSphere 'ibm-webservicesclient-ext.xmi' files [SECURITY.UEC.WCMC-2]
- Ensure SOAP messages are digitally signed in WebSphere 'ibm-webservicesclient-ext.xmi' files [SECURITY.UEC.WCMI-2]
- Avoid misconfiguring timestamps in WebSphere 'ibm-webservicesclient-ext.xmi' files [SECURITY.UEC.WCMT-3]
- Ensure WS-Security is enabled in WebSphere 'ibm-webservicesclient-ext.xmi' files [SECURITY.UEC.WCMWS-1]
- Avoid unencrypted passwords in WebSphere 'ibm-webservicesclient-ext.xmi' files [SECURITY.UEC.WCPWD-1]
- Avoid unsigned timestamps in WebSphere 'ibm-webservicesclient-ext.xmi' files [SECURITY.UEC.WCUTS-3]
- Ensure all web content directories have a "welcome file" [SECURITY.UEC.WELC-1]
- Ensure SOAP messages are encrypted in WebSphere 'ibm-webservices-ext.xmi' files [SECURITY.UEC.WMC-2]
- Ensure SOAP messages are digitally signed in WebSphere 'ibm-webservices-ext.xmi' files [SECURITY.UEC.WMI-2]
- Avoid misconfiguring timestamps in WebSphere 'ibm-webservices-ext.xmi' files [SECURITY.UEC.WMT-3]
- Ensure WS-Security is enabled in WebSphere 'ibm-webservices-ext.xmi' files [SECURITY.UEC.WMWS-1]
- Avoid unencrypted passwords in WebSphere 'ibm-webservices-ext.xmi' files [SECURITY.UEC.WPWD-1]
- Ensure that the Rampart WS-Security module is enabled in Axis2 configuration files [SECURITY.UEC.WSS-3]
- Avoid unsigned timestamps in WebSphere 'ibm-webservices-ext.xmi' files [SECURITY.UEC.WUTS-3]

Parasoft Security Rules

- **Weak Security Controls [SECURITY.WSC]**
  - Avoid using anonymous "privileged" classes when invoking "AccessController.doPrivileged()" [SECURITY.WSC.ACDP-4]
  - Do not call the 'printStackTrace()' method of "Throwable" objects [SECURITY.WSC.ACPST-5]
  - Avoid hard-coding the arguments to certain methods [SECURITY.WSC.AHCA-3]
  - Avoid constructors and overriding methods which are more accessible than those of their super classes [SECURITY.WSC.AMA-3]
  - Inspect usage of standard API calls that bypass security [SECURITY.WSC.APIBS-3]
  - Avoid turning raw text into xml [SECURITY.WSC.ARXML-3]
  - Inspect usage of scripting API [SECURITY.WSC.ASAPI-5]
  - Do not use inner classes [SECURITY.WSC.AUIC-3]
  - Allow only certain providers to be specified for the 'Security.addProvider()' method [SECURITY.WSC.BP-3]
  - Keep all access control methods centralized to enforce consistency [SECURITY.WSC.CACM-3]
  - Keep all authentication methods centralized to enforce consistency [SECURITY.WSC.CAM-3]
  - Always clone array parameters which are stored to fields [SECURITY.WSC.CAP-3]
  - Only call "final" methods from specified code blocks in non-"final" classes [SECURITY.WSC.CFM-3]
  - Only "clone()" instances of "final" classes [SECURITY.WSC.CIFC-3]
  - Avoid using cryptographic keys which are too short [SECURITY.WSC.CKTS-3]
  - Do not define custom class loaders [SECURITY.WSC.CL-2]
  - Inspect instantiations of 'ClassLoader' objects [SECURITY.WSC.CLI-3]
  - Do not override any 'ClassLoader' method except 'findClass()' [SECURITY.WSC.CLO-3]
  - Make your 'clone()' method "final" for security [SECURITY.WSC.CLONE-3]
  - Do not pass mutable objects to 'ObjectOutputStream' in the 'writeObject()' method [SECURITY.WSC.CMO-2]
  - Do not define custom 'SecurityManager's [SECURITY.WSC.DCSM-3]
  - Avoid DNS lookups for decision making [SECURITY.WSC.DNSL-4]
  - Make your classes nondeserializeable [SECURITY.WSC.DSER-5]
  - Make immutable classes final [SECURITY.WSC.FIMU-5]
  - Ensure that Secure Processing is used [SECURITY.WSC.FXMLP-5]
  - Avoid using hard-coded cryptographic keys [SECURITY.WSC.HCCK-3]
  - Avoid passing hardcoded usernames/passwords/URLs to database connection methods [SECURITY.WSC.HCCS-2]
  - Avoid using insecure algorithms for cryptography [SECURITY.WSC.ICA-2]
  - Make immutable classes final [SECURITY.WSC.INIVF-5]
  - Make all member classes and interfaces "private" [SECURITY.WSC.INNER-2]
  - Always call 'HttpSession.invalidate()' before 'LoginContext.login()' [SECURITY.WSC.ISL-4]
  - Avoid non-random "byte[]" when using IvParameterSpec [SECURITY.WSC.IVR-4]
  - Make your classes noncloneable [SECURITY.WSC.MCNC-5]
  - Call authentication methods to enforce consistency [SECURITY.WSC.PAC-2]
  - Call access control methods to enforce consistency [SECURITY.WSC.PACC-2]
  - Declare subclasses of 'PrivilegedAction', 'PrivilegedExceptionAction', and 'PrivilegedActionException' "final" [SECURITY.WSC.PAF-3]
  - Declare subclasses of 'Permission' and 'BasicPermission' "final" [SECURITY.WSC.PBPSF-3]
  - Do not allow password fields to be autocompleted [SECURITY.WSC.PPF-2]

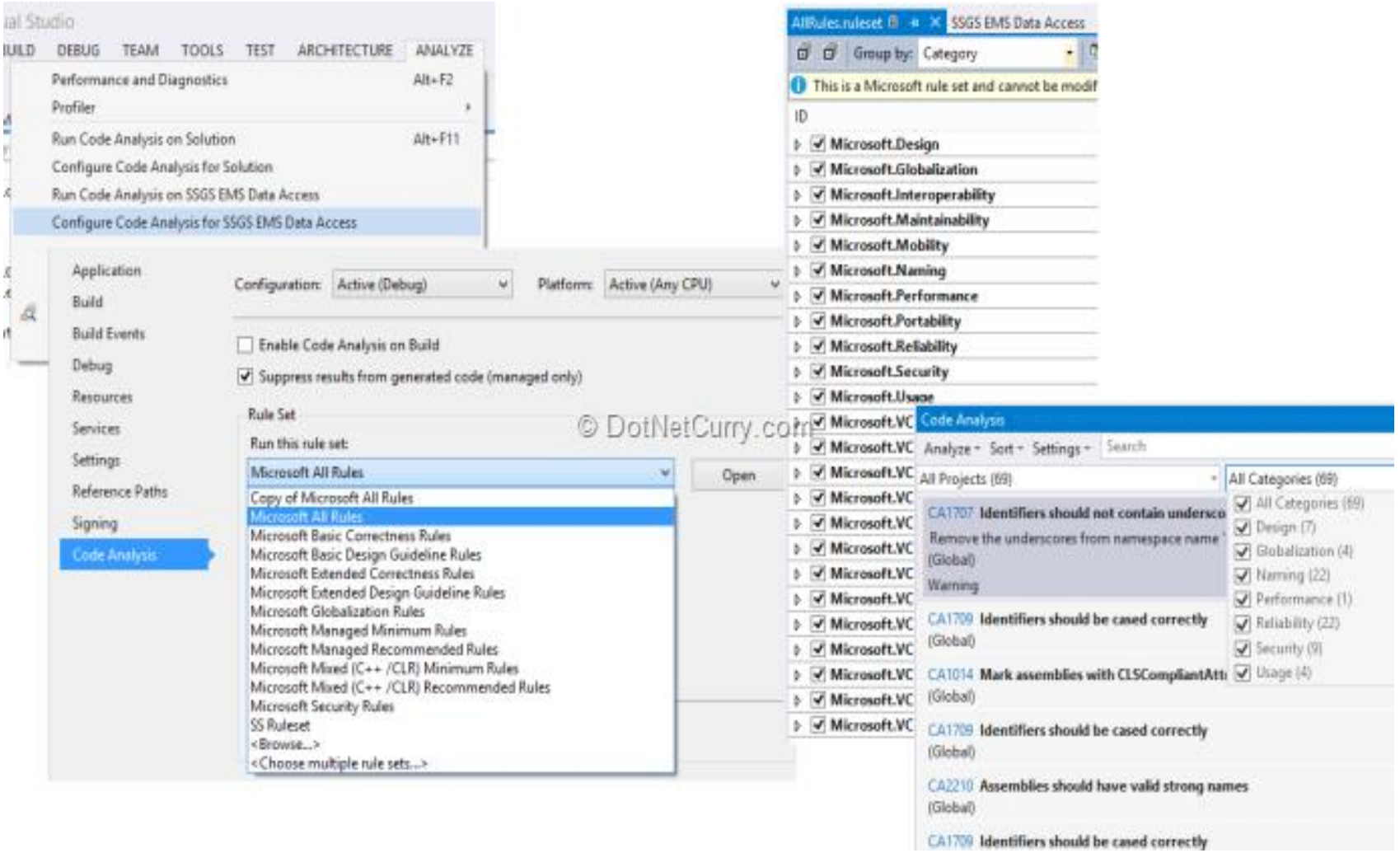
Parasoft Security Rules

- Ensure that all Permissions, PrivilegedActions, and PrivilegedActionExceptions are declared in the same package [SECURITY.WSC.PPKG-3]
- Declare the 'run()' method of 'PrivilegedAction' and 'PrivilegedExceptionAction' implementations "final" [SECURITY.WSC.PRMF-3]
- Do not declare fields as "public" "static" "final" 'Collection' or 'Map' objects [SECURITY.WSC.PSFC-3]
- Inspect 'Random' objects or 'Math.random()' methods that could indicate areas where malicious code has been placed [SECURITY.WSC.RDM-5]
- Enforce 'SecurityManager' checks before setting or getting fields [SECURITY.WSC.SCF-2]
- Enforce 'SecurityManager' checks in methods of 'Cloneable' classes [SECURITY.WSC.SCLONE-2]
- Enforce 'SecurityManager' checks in methods of 'Serializable' classes [SECURITY.WSC.SCSER-2]
- Ensure 'SecurityManager' check in constructor of "public" non-"final" sensitive type [SECURITY.WSC.SCSM-3]
- Make your classes nonserializeable [SECURITY.WSC.SER-5]
- Avoid string literals except in constant declarations and calls to System.out or System.err's 'print' or 'println' methods [SECURITY.WSC.SL-4]
- Ensure 'SecurityManager' checks before 'Socket' transfers or retrievals [SECURITY.WSC.SMSTR-4]
- Use 'java.security.SecureRandom' instead of 'java.util.Random' or 'Math.random()' [SECURITY.WSC.SRD-2]
- Do not use sockets in web components [SECURITY.WSC.SS-3]
- Ensure that an appropriate security manager is set [SECURITY.WSC.SSM-3]
- Do not call 'System.setProperty()' in a web component [SECURITY.WSC.SSP-3]
- Use library methods for string replacements of special characters in HTML and XML [SECURITY.WSC.STREP-5]
- Avoid 'main()' methods because they may allow unauthorized access to classes [SECURITY.WSC.UMAIN-5]
- Use the "getSecure()" and "setSecure()" methods to enforce the use of secure cookies [SECURITY.WSC.UOSC-3]
- Use the SSL-enabled version of classes when possible [SECURITY.WSC.USC-2]
- Use wrapper methods instead of calling dangerous or problematic methods directly (custom rule) [SECURITY.WSC.UWM-3]
- Always verify JarFile signatures [SECURITY.WSC.VJFS-3]
- Inspect usage of scripting API [SECURITY.WSC.ZOIS-3]

Visual Studio Code Analysis

- Microsoft Visual Studio Team System and higher versions provide the capabilities of PREFast and FxCop integrated into the development environment
  - http://msdn.microsoft.com/en-us/library/ms182025(VS.80).aspx
- Enabled via **/analyze** command-line switch or through Visual Studio project properties settings

Visual Studio Code Analysis Feature



Visual Studio Security Rule Set

- http://msdn.microsoft.com/en-us/library/dd264921.aspx

Rule	Description
CA2100	Review SQL queries for security vulnerabilities
CA2102	Catch non-CLSCompliant exceptions in general handlers
CA2103	Review imperative security
CA2104	Do not declare read only mutable reference types
CA2105	Array fields should not be read only
CA2106	Secure asserts
CA2107	Review deny and permit only usage
CA2108	Review declarative security on value types
CA2109	Review visible event handlers
CA2111	Pointers should not be visible
CA2112	Secured types should not expose fields
CA2114	Method security should be a superset of type
CA2115	Call GC.KeepAlive when using native resources
CA2116	APTCA methods should only call APTCA methods
CA2117	APTCA types should only extend APTCA base types
CA2118	Review SuppressUnmanagedCodeSecurityAttribute usage
CA2119	Seal methods that satisfy private interfaces
CA2120	Secure serialization constructors

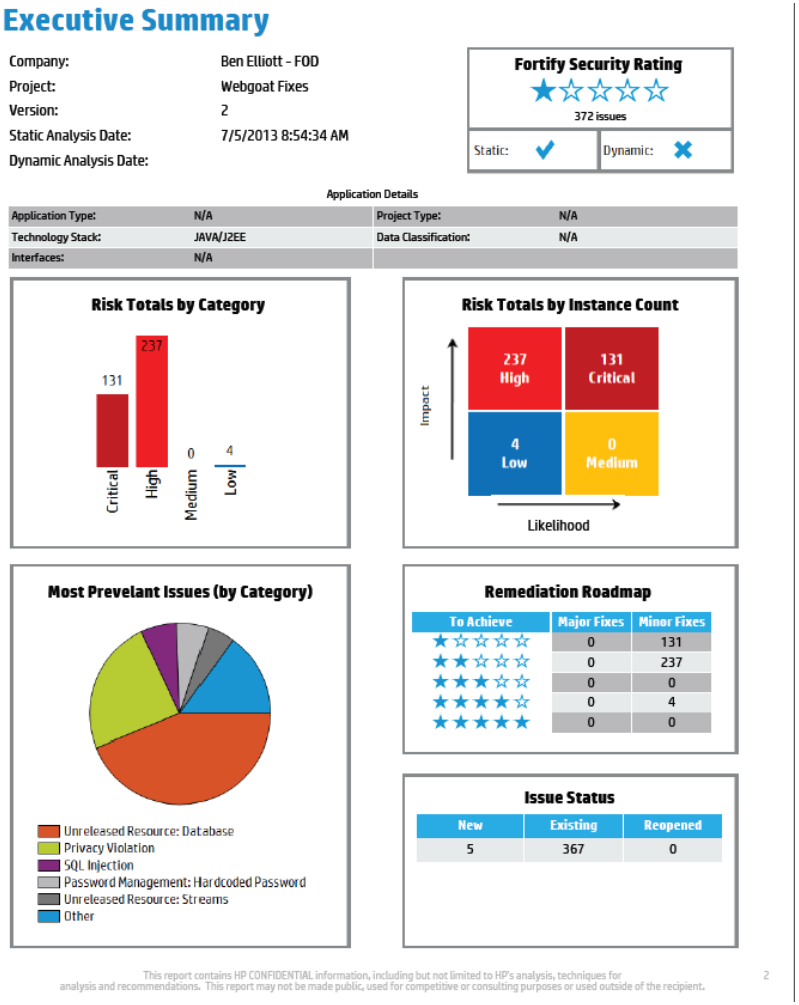
Visual Studio Security Rule Set

- http://msdn.microsoft.com/en-us/library/dd264921.aspx

CA2121	Static constructors should be private
CA2122	Do not indirectly expose methods with link demands
CA2123	Override link demands should be identical to base
CA2124	Wrap vulnerable finally clauses in outer try
CA2126	Type link demands require inheritance demands
CA2130	Security critical constants should be transparent
CA2131	Security critical types may not participate in type equivalence
CA2132	Default constructors must be at least as critical as base type default constructors
CA2133	Delegates must bind to methods with consistent transparency
CA2134	Methods must keep consistent transparency when overriding base methods
CA2135	Level 2 assemblies should not contain LinkDemands
CA2136	Members should not have conflicting transparency annotations
CA2137	Transparent methods must contain only verifiable IL
CA2138	Transparent methods must not call methods with the SuppressUnmanagedCodeSecurity attribute
CA2139	Transparent methods may not use the HandleProcessCorruptingExceptions attribute
CA2140	Transparent code must not reference security critical items
CA2141	Transparent methods must not satisfy LinkDemands
CA2142	Transparent code should not be protected with LinkDemands
CA2143	Transparent methods should not use security demands
CA2144	Transparent code should not load assemblies from byte arrays
CA2145	Transparent methods should not be decorated with the SuppressUnmanagedCodeSecurityAttribute
CA2146	Types must be at least as critical as their base types and interfaces
CA2147	Transparent methods may not use security asserts
CA2149	Transparent methods must not call into native code
CA2210	Assemblies should have valid strong names

Exercise: Reviewing Through Fortify Code Reports

- Spend some quality time to review through Scan Reports from the Fortify SCA Tool



Executive Summary

Company:	Ben Elliott - FOD
Project:	Webgoat Fixes
Version:	2
Static Analysis Date:	7/5/2013 8:54:34 AM
Dynamic Analysis Date:	

Fortify Security Rating

★☆☆☆☆

372 issues

Static: ✔ Dynamic: ✖

Application Details

Application Type:	N/A	Project Type:	N/A
Technology Stack:	JAVA/J2EE	Data Classification:	N/A
Interfaces:	N/A		

Risk Totals by Category

Risk Totals by Instance Count

Most Prevelant Issues (by Category)

Remediation Roadmap

To Achieve	Major Fixes	Minor Fixes
★☆☆☆☆	0	131
★★☆☆☆	0	237
★★★☆☆	0	0
★★★★☆	0	4
★★★★★	0	0

Issue Status

New	Existing	Reopened
5	367	0

GROUP
ENTERPRISE

NCS Code Quality Gates & Code Scan Statistics

Built in Quality

SwQA

Share Standards & Guidelines

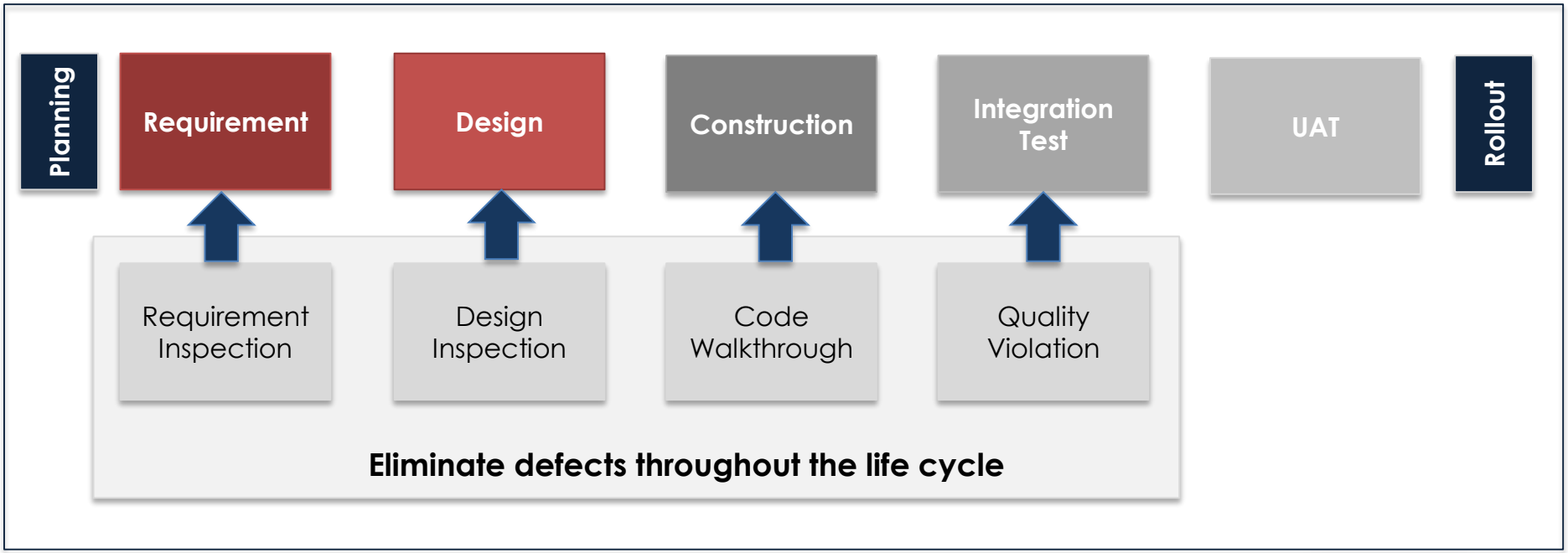
0 Design & DB Review

1 Static Code Scan

2 Static Code Scan SQL Performance

3 Performance & Vulnerability Test

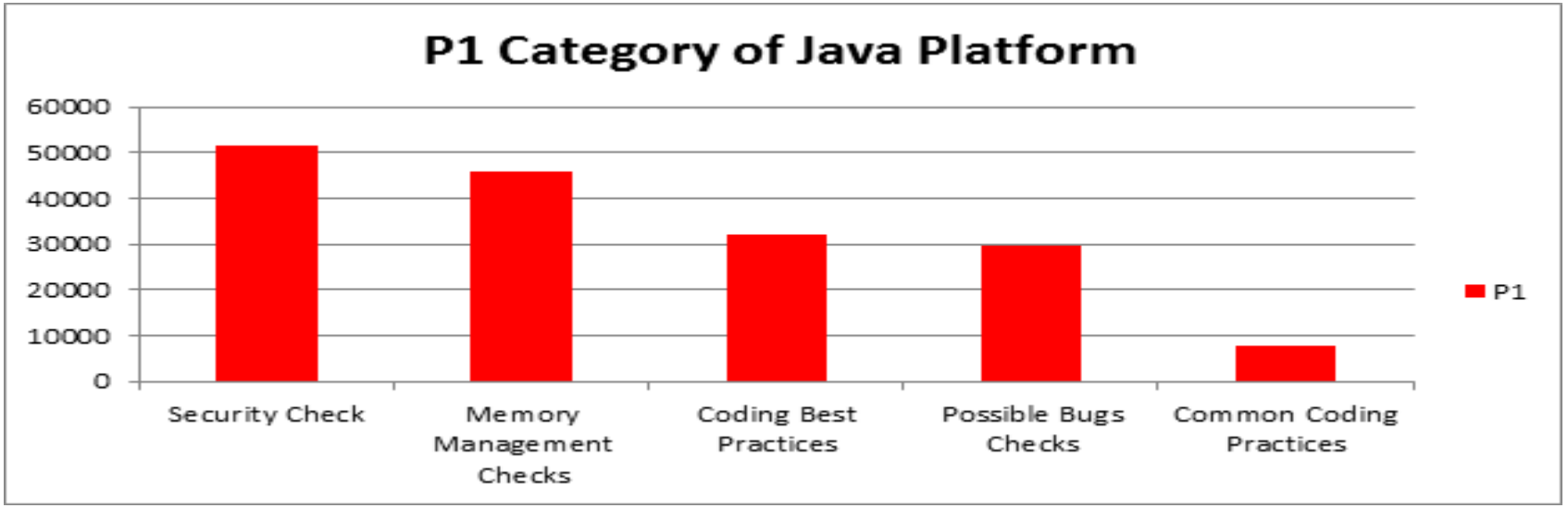
4 Static Code Scan

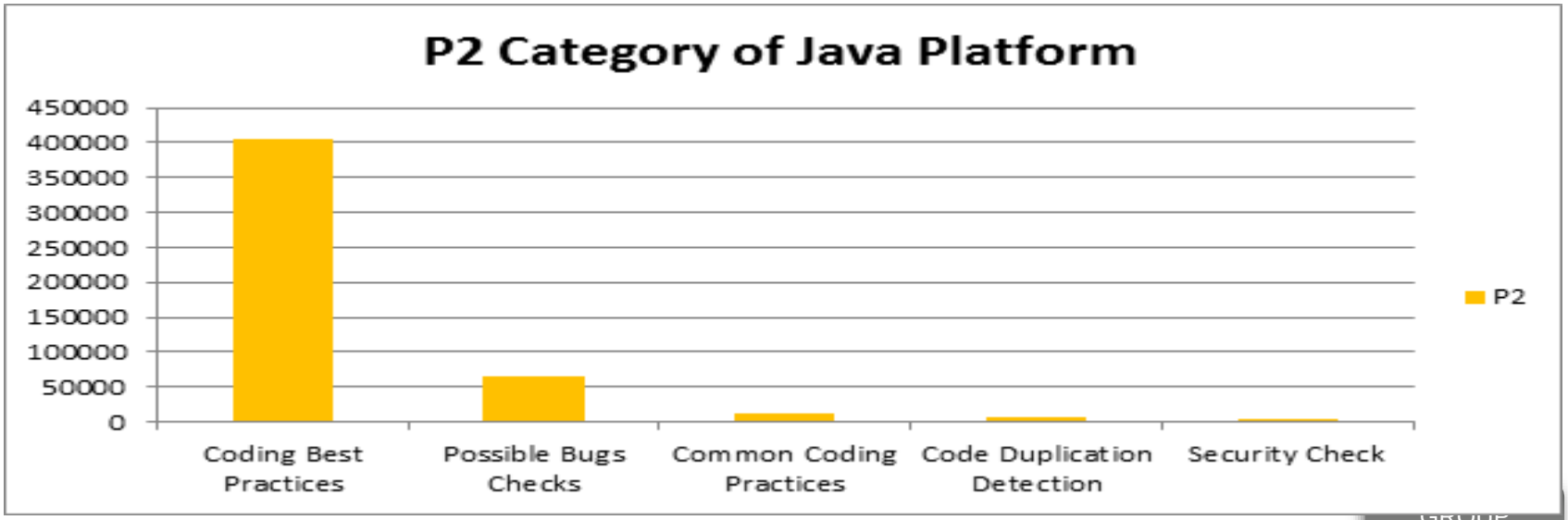


SwQA – Project Gates Governance

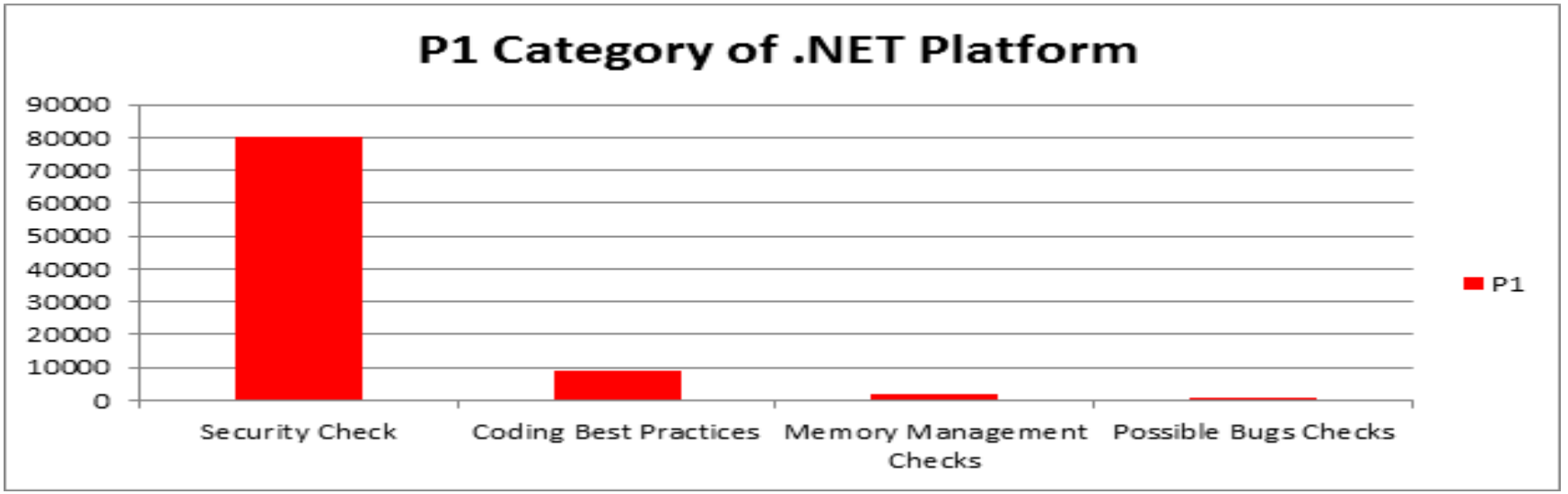
	Gate 0 Design & DB Review	Gate 1 Construction	Gate 2 SIT SQL Performance	Gate 3 Performance & Vulnerability Test	Gate 4 Before Go Live
Expected Project Status / Accomplishments	Ready for DB and Design Review	Ready for System Integration Testing (SIT) Completed all unit level testing. Developers uses client static tools such as PMD, FindBugs/MSDN to detect bugs and fix.	Ready for User Acceptance Testing	Ready for Load/Performance/Stress Testing , Vulnerability and Penetration Testing • Mandatory For Public Facing Application • TCC to provide Support	Ready for Go Live
Entry Criteria for Static Code Scan		Implemented critical recommendation made by CAO on Architecture, DB and Design.	Completed SIT testing and have resolved all agreed P1 and P2 defects reported at Gate 1.	Completed UAT Testing and have resolved all agreed P1 and P2 defects reported at Gate 2.	Completed agreed defects identified at Gate 1 and/or Gate 2.
Exit Criteria	Provide Plan to address concern raised at the review	Provide Plan to resolve P1 and P2 defects.	Provide Plan to resolve P1 and P2 defects Resolved all SQL performance issues identified with CAO	Meets project's SLA for Performance and Load testing Meets application's security requirements.	All critical defects cleared and all agreed defects identified at Gate 1 and 2 completed. Implemented critical recommendation made by CAO on Architecture, DB and Design Meet Performance & Security requirement
Escalation Parties if fails entry or exit criteria	SDD	SDD	SDD	SDD/ SwQA	Head SDC

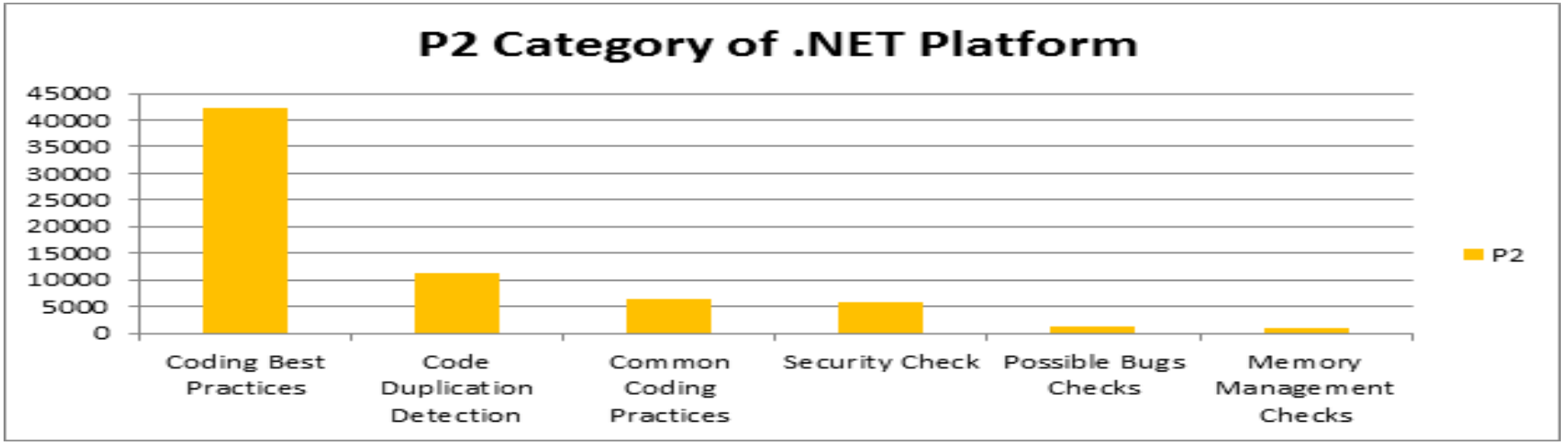
Java Defects by Category





.NET Defects by Category



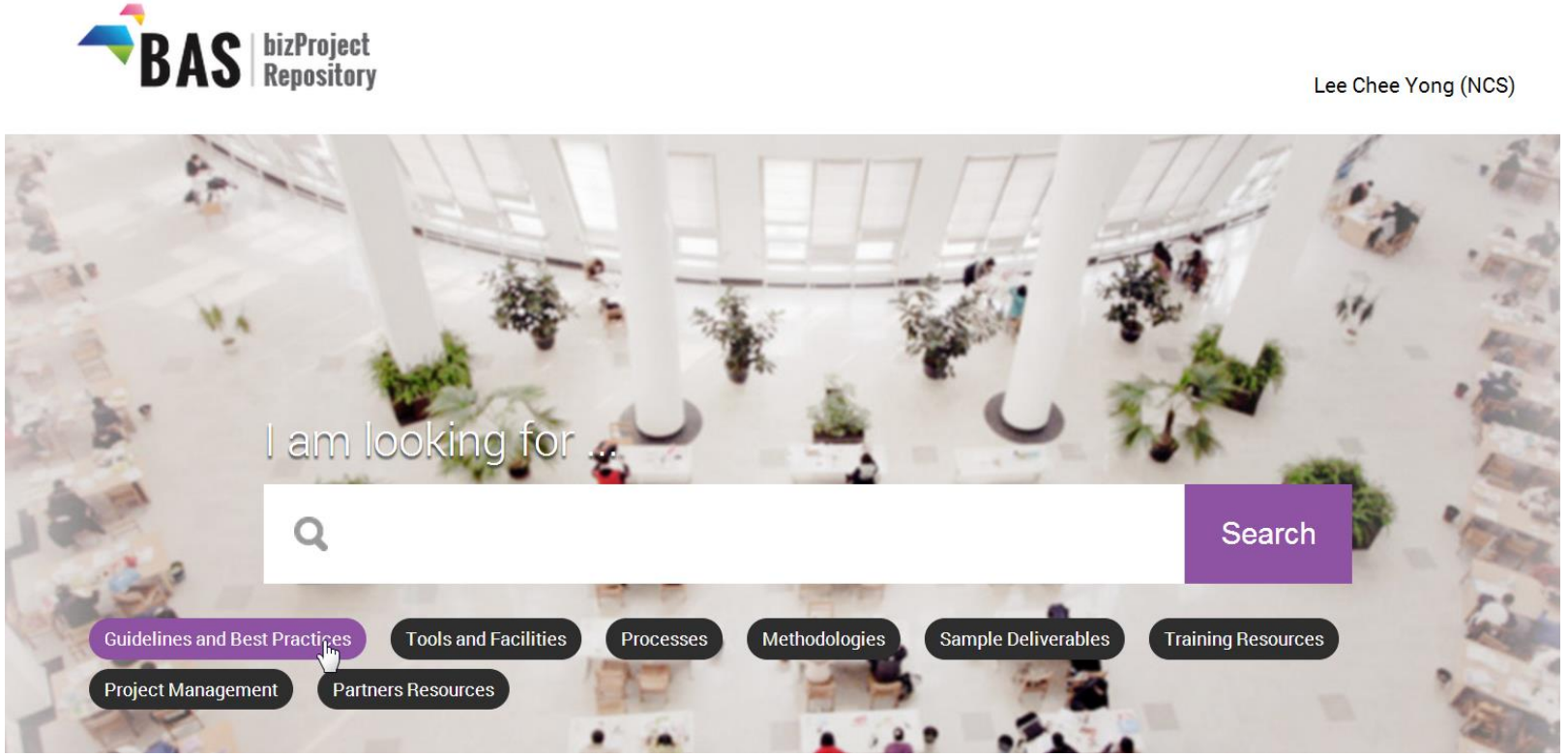


Walk Through of Top 25 Defects Commonly Found in NCS Source Codes

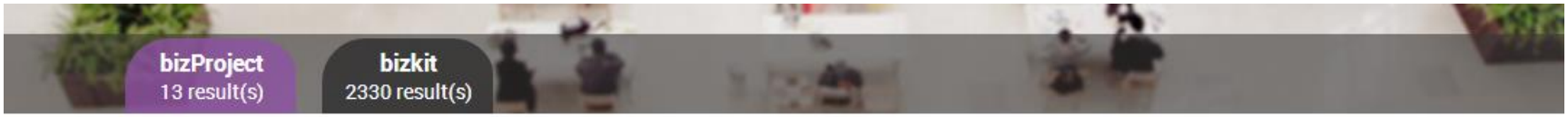
- To go through the Java Top 25 Defects document
- To go through the.NET Top 25 Defects document

BAS BizProject

- https://community.ncs.com.sg/project/bizProject/_layouts/bizProject/home.aspx



BAS BizProject



Internal Sharing

DB Naming Standards (Physical) v1.0.pdf

Not Reviewed

Guidelines and Best Practices
Industry:All
Author:Tan Boon Pieng (NCS)
Completed date:23 Jun 2014
Posted date:12 Jun 2014

Rating

Internal Sharing

Programming Guidelines (ASP NET) v4.1.pdf

Not Reviewed

Guidelines and Best Practices
Industry:All
Author:Tan Boon Pieng (NCS)
Completed date:29 Jun 2014
Posted date:4 Jun 2014

Rating

Internal Sharing

NCS Secure Web Application Programming Guide v1 0.pdf

Not Reviewed

Guidelines and Best Practices
Industry:All
Author:Tan Boon Pieng (NCS)
Completed date:29 May 2014
Posted date:2 Jun 2014

Rating


Project Alpha-
Secure
Security Tender
Specifications
GROUP
ENTERPRISE

Group Discussion – Part 2

- Stay in the previous grouping for this activity. Elect a new presenter. The group is expected to present their views based on this hypothetical situation.

- You are part of the project management team (PMT) of a new awarded project called Project “Alpha-Secure”. In the tender specifications, stringent security tender clauses have been specified. Project Alpha-Secure is an **Internet fronting web application** housed in G-Cloud with an backend system linking to our government agency intranet system. **Sensitive citizen data** such as income level, personal contact information are stored in this system. Customer has just experience a security breech in another project carried out by another vendor and is very anxious not to experience another similar embarrassing situation.

  – 1. Based on the tender specifications, what are the additional activities required in the project plans to address these security clauses in terms of:
    (a) Training and Staffing
    (b) System Design
    (c) Coding Practices
    (b) Testing Required
    (e) Any other areas

  – 2. What are the challenges and constraints you foresee carrying out these activities proposed in Qn 1?

alpha secure

GROUP
ENTERPRISE