Network Security Workshop

5-7 March 2018

Funafuti, Tuvalu

Sponsored by:

Issue Date:

Revision:

APNIC

Overview

- **Network Security Fundamentals**
- Threat Pragmatics
- Cryptography Basics
- SSH
- Network Infrastructure Filtering at the border
- PGP
- TLS/SSL
- IPSec
- IDS & Snort
- Wireshark

APNIC

2

Why Security?

- The Internet was initially designed for connectivity
  - Trust is assumed, no security
  - Security protocols added on top of the TCP/IP
- Fundamental aspects of information must be protected
  - Confidential data
  - Employee information
  - Business models
  - Protect identity and resources
- The Internet has become fundamental to our daily activities (business, work, and personal)

Internet Evolution

LAN connectivity

Application-specific
More online content

Application/data
hosted in the “cloud”

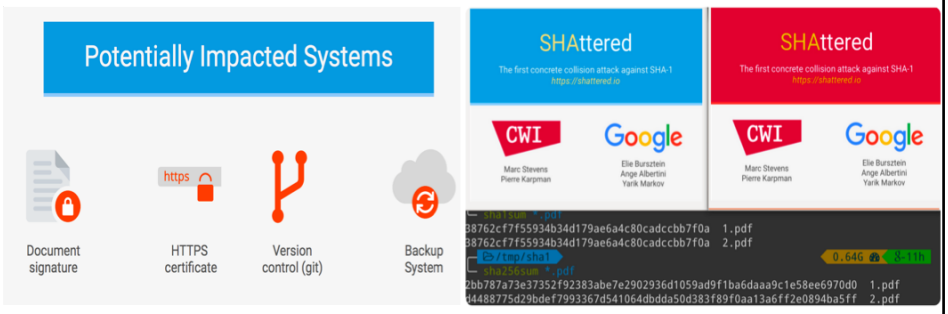
Different ways to handle security as the Internet evolves

Recent Incidents

- **WannaCry Ransomware (May 2017)**
  - As of 12 May, 45K attacks across 74 countries
  - Remote code execution in SMBv1 using EternalBlue exploit
    - TCP 445, or via NetBIOS (UDP/TCP 135-139)
  - Patch released on 14 March 2017 (MS17-010)
    - https://technet.microsoft.com/en-us/library/security/ms17-010.aspx
  - Exploit released on 14 April 2017

Recent Incidents

- **SHA-1 is broken (Feb 23, 2017)**
  - colliding PDF files: obtain same SHA-1 hash of two different pdf files, which can be *abused* as a valid signature on the second PDF file.
    - https://security.googleblog.com/2017/02/announcing-first-sha1-collision.html

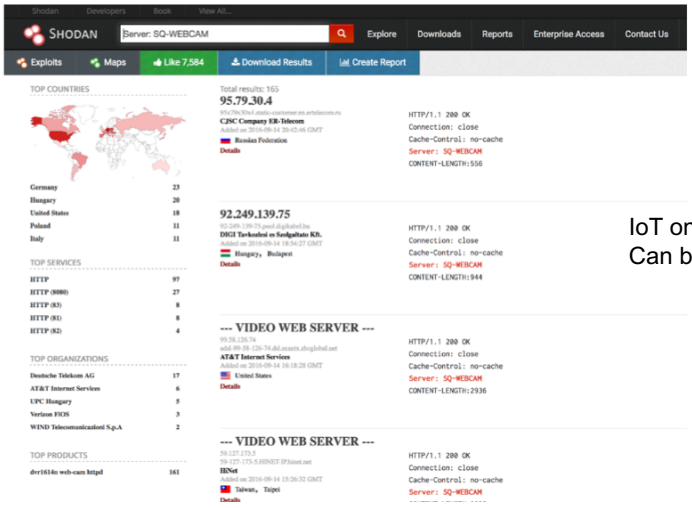


Recent Incidents

- **San Francisco Rail System Hacker Hacked (Nov 2016)**
  - Ransomware attack on San Francisco public transit gave everyone a free ride (cryptom27@yandex.com)
    - Encrypts boot sectors (ransom for decryption) - Mamba
  - Java vulnerability not patched (Security Alert CVE-2015-4852 since Nov 2015 from Oracle)

A copy of the ransom message left behind by the "Mamba" ransomware.

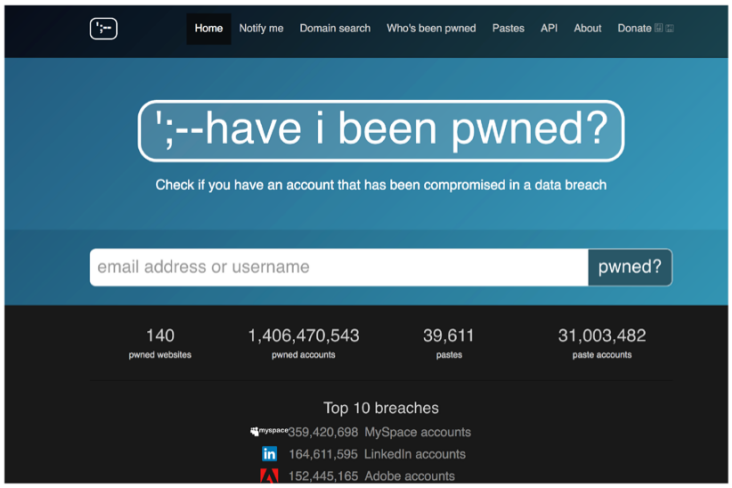
Shodan.io



IoT online
Can be searched!

haveibeenpwned.com

- Have you been compromised?



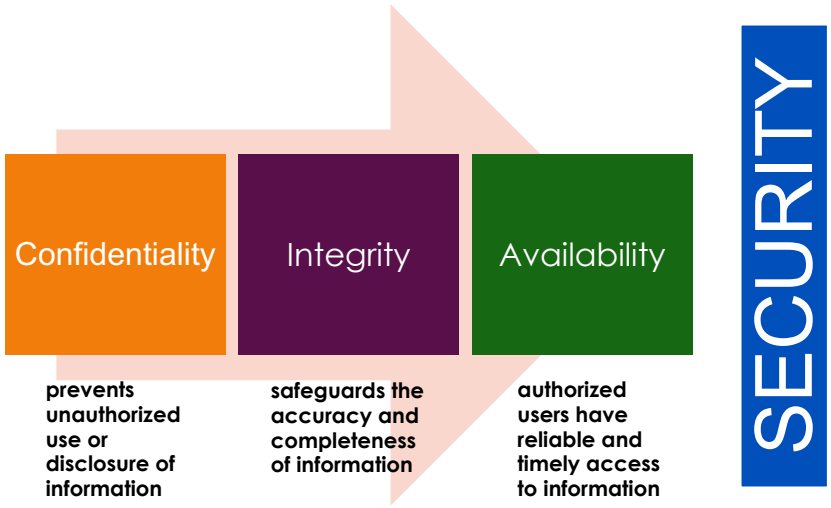
2 factor authentication --- https://www.turnon2fa.com/tutorials

Let's Encrypt

https://www.letsencrypt.org

HTTPS Everywhere

Goals of Information Security



APNIC 11

Threats, Vulnerability, and Risks

- Threat
  - circumstance or event with potential to cause harm to a networked system
- Vulnerability
  - A weakness that can be exploited
    - Software bugs
    - Design flaws
    - Configuration mistakes
    - Lack of encryption
- Risk
  - The likelihood that a particular vulnerability will be exploited

APNIC 12

Threat

- “a motivated, capable adversary”
- Examples:
  - Human Threats
    - Intentional or unintentional
    - Malicious or benign
  - Natural Threats
    - Earthquakes, tornadoes, floods, landslides
  - Environmental Threats
    - Long-term power failure, pollution, liquid leakage

Vulnerability

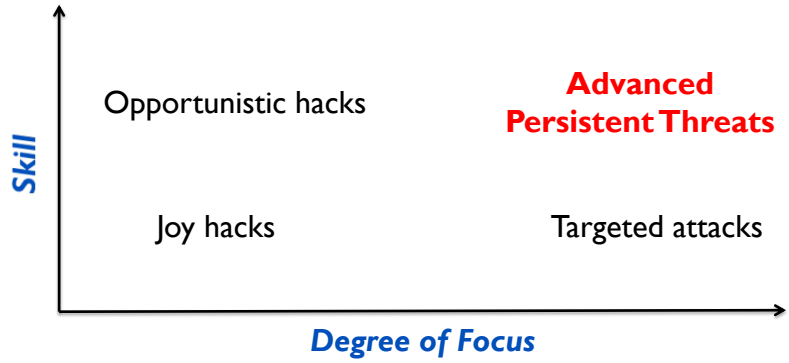
- A weakness in security procedures, network design, or implementation that can be exploited to violate a corporate security policy
  - Software bugs
  - Configuration mistakes
  - Network design flaw
  - Lack of encryption
- Where to check for vulnerabilities?
- Exploit
  - Taking advantage of a vulnerability

Risk

- Likelihood that a vulnerability will be exploited
- Some questions:
  - How likely is it to happen?
  - What is the level of risk if we decide to do nothing?
  - Will it result in data loss?
  - What is the impact on the reputation of the company?
- Categories:
  - High, medium or low risk

Risk = Threat * Vulnerability
(* Impact)

The Threat Matrix



Source: Thinking Security – Steve M. Bellovin

Joy Hacks

- For fun - with little skill using known exploits
- Minimal damage - especially unpatched machines
- Random targets – anyone they can hit
- Most hackers start this way – learning curve

Opportunistic Hacks

- Skilled (often very skilled) - also don't care whom they hit
  - Know many different vulnerabilities and techniques
- Profiting is the goal - bank account thefts, botnets, ransomwares….
  - WannaCry? Petya?
- Most phishers, virus writers, etc.

Targeted Attacks

- Have a specific target!
- Research the target and tailor attacks
  - physical reconnaissance
- At worst, an insider (behind all your defenses)
  - Not so happy
- Tools like “spear-phishing”
- May use 0-days

Advanced Persistent Threats

- Highly skilled (well funded) - specific targets
  - Mostly 0-days
- Sometimes (not always) working for a nation-state
  - Think Stuxnet (up to four 0-days were used)
- May use non-cyber means:
  - burglary, bribery, and blackmail
- ***Note:*** many lesser attacks blamed on APTs

Are you a Target?

- Biggest risk?
  - assuming you are not interesting enough!
- Vendors/System Integrators and their take on security:
  - Either underwhelming or Overwhelming ☹

Defense Strategies

- Depends on what you're trying to protect
- Tactics that keep out teenagers won't keep out a well-funded agency
- But stronger defenses are often much more expensive, and cause great inconvenience

What Are You Protecting?

- Identify your critical Assets
  - Both tangible and intangible (patents, methodologies) assets
    - Hardware, software, data, people, documents
  - Who would be interested?
- Place a Value on the asset
  - Different assets require different level of protection
  - Security measures must be in proportion with asset value
    - How much can you afford?
- Determine Likelihood of breaches
  - threats and vulnerabilities ?

Against Joy Hacks

- By definition, joy hackers use known exploits
- Patches exist for known holes:
  - Up to date system patches
  - Up to date antivirus database
- Ordinary enterprise-grade firewalls will also repel them

Opportunistic Hacks

- Sophisticated techniques used
- You need multiple layers of defense
  - Up to date patches and anti-virus
  - Firewalls
  - Intrusion detection
  - Lots of attention to log files

Targeted Attacks

- Targeted attacks exploit knowledge of target
  - Try to block or detect reconnaissance
  - Security policies and procedures matter a lot
    - How do you respond to phone callers?
    - What do people do with unexpected attachments?
    - USB sticks in the parking
- Hardest case: disgruntled employee or ex-employee
  - Already behind your defenses
  - Think Manning & Snowden

Advanced Persistent Threats

- ☹ very very hard defend against!
- Use all of the previous defenses
- There are no sure answers
- Pay special attention to policies and procedures
- Investigate all oddities

Putting CIA in Context

- **Scenario:** XYZ has a webmail for employees to access their email accounts.Sometimes they share reports and communicate with customers.
  - **Confidentiality:**
    - Username and password (or user credentials) to access webmail should only be known to the user. Contents of the email communication should only be available to the intended recipients only.
  - **Integrity:**
    - Emails that are received or sent out are not modified from their original form.
  - **Availability:**
    - Since email communication is critical to the company, this email service must be available all the time
- Question: Think about what we can put in place to make sure the CIA can be achieved

Causes of Security Related Issues

- Protocol error
  - No one gets it right the first time
- Software bugs
  - Is it a bug or feature ?
- Active attack
  - Target control/management plane
  - Target data plane
  - More probable than you think !
- Configuration mistakes
  - Most common form of problem

Threat & Threat Source Example

Vulnerability	Threat-Source	Threat Action
Critical vulnerability in a web server software was identified but software patches have not been applied	Unauthorized users (i.e. Internal employees, hackers, criminals)	Obtaining unauthorized access to information (files, sensitive information on the web server
Terminated employees credentials (username & password) are not removed from the system	Terminated Employees	Accessing companies systems and proprietary information

What Can Intruders Do?

- Eavesdrop - compromise routers, links, or DNS
- Send arbitrary messages (spoof IP headers and options)
- Replay recorded messages
- Modify messages in transit
- Write malicious code and trick people into running it
- Exploit bugs in software to ‘take over’ machines and use them as a base for future attacks

Attack Motivation

- Criminal
  - Criminal who use critical infrastructure as a tools to commit crime
  - Their motivation is money
- War Fighting/Espionage/Terrorist
  - What most people think of when talking about threats to critical infrastructure
- Patriotic/Principle
  - Large groups of people motivated by cause - be it national pride or a passion aka Anonymous

Attack Motivation

- Nation States want SECRETS
- Organized criminals want MONEY
- Protesters or activists want ATTENTION
- Hackers and researchers want KNOWLEDGE

Source: NANOG60 keynote presentation by Jeff Moss, Feb 2014

Goals are Determined by

- Services offered vs. security provided
  - Each service offers its own security risk
- Ease of use vs. security
  - Easiest system to use allows access to any user without password
- Cost of security vs. risk of loss
  - Cost to maintain

Goals must be communicated to all users, staff, managers, through a set of security rules called "security policy"

Example of Security Controls

Category	Example of Controls	Purpose
Policy & Procedure	Cyber Security Policy, Incident Handling Procedure	Make everyone aware of the importance of security, define role and responsibilities, scope of the problem
Technical	Firewall, Intrusion Detection System, Anti Virus Software	Prevent and detect potential attacks, mitigate risk of breach at the network or system layer
Physical	CCTV, Locks, Secure working space	Prevent physical theft information assets or unauthorized physical access

What Are You Protecting?

- Identify Critical Assets
  - Hardware, software, data, people, documentation
- Place a Value on the Asset
  - Intangible asset – importance or criticality
  - Tangible asset – replacement value, training costs and/or immediate impact of the loss
- Determine Likelihood of Security Breaches
  - What are threats and vulnerabilities ?

Impact and Consequences

- Data compromise
  - Stolen data
  - can be catastrophic for a financial institution
- Loss of data integrity
  - Negative press or loss of reputation (bank, public trust)
- Unavailability of resources
  - The average amount of downtime following a DDoS attack is 54 minutes
  - The average cost of one minute of downtime due to DDoS attack is $22,000*

* Based on a Ponemon Institute study (2012)

Security (What you Already Know)

- Security is a process
  - Design
  - Deployment / Implementation
  - Breach / Response
  - People, Process & Technology
- Business Implication of Security Incidents
  - Availability
  - Reputation
- Multi-stakeholders
  - Customers
  - LEAs (Law Enforcement Authorities)
  - Regulators
- Concerns
  - Cyber Crime (Take Down / Prevention/ Attribution & Investigation)
  - Cyber Hygiene (Cleaning up)
  - Resilience (Compliance / Best Practices) – Critical Information Protection
  - End-User Awareness*

Challenges in Implementing Security

- Many
  - Lack of Awareness
  - Not enough resources
  - We are moving too slow
- Root Cause?
- Challenges to Security Professionals
  - Don't know where to start
  - Too many things to learn & master
  - Too expensive to do training and certifications
  - No support from end-users & top management
  - 24x7 expectations - I have my own life!

Computer Security


What my parents think I do


What my friends think I do


What my boss thinks I do


What my girlfriend thinks I do


What the media thinks I do

What I actually do

Solutions

- Make security a priority (Sell it)
- Don't reinvent the wheel
- Keep on learning
- Keep sharing and contributing

Overview

- Network Security Fundamentals
- **Threat Pragmatics**
- Cryptography Basics
- SSH
- Network Infrastructure Filtering at the border
- PGP
- TLS/SSL
- IPSec
- IDS & Snort
- Wireshark

APNIC

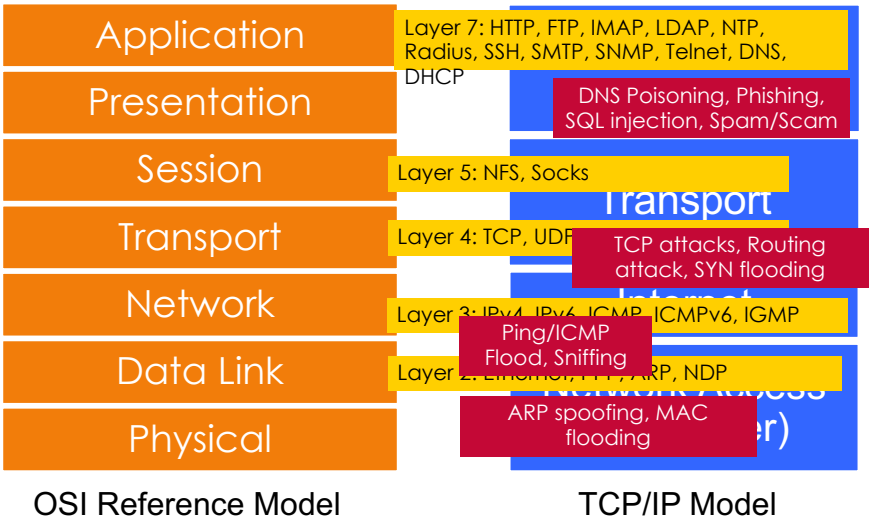
43

Target

- Many sorts of targets:
  - Network infrastructure
  - Network services
  - Application services
  - User machines
- What's at risk?

APNIC

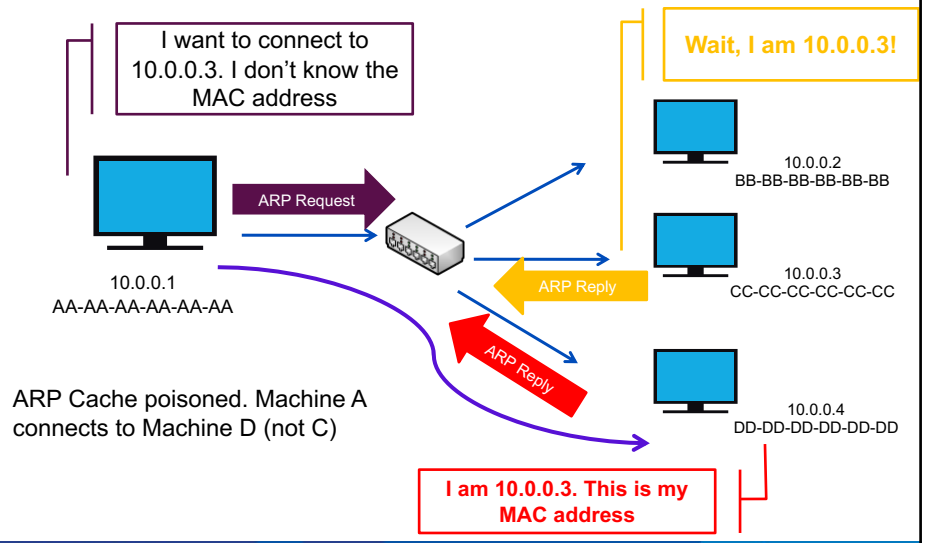
Attacks on Different Layers



Layer 2 Attacks

- ARP Spoofing
- MAC attacks
- DHCP attacks
- VLAN hopping

ARP Spoofing



MAC Flooding

- Exploits the limitation of all switches
- CAM = Content Addressable memory = stores info on the mapping of individual MAC addresses to physical ports on the switch.

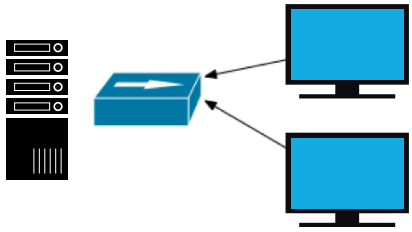
Attacker floods the switch interface with very large number of Ethernet frames with different fake source MAC address.

	Port 1	Port 2	Port 3	Port 4
00:01:23:45:67:A1	x			
00:01:23:45:67:B2	x			
00:01:23:45:67:C3	x			
00:01:23:45:67:D4	x			

DHCP Attacks

- DHCP Starvation Attack
  - Broadcasting vast number of DHCP requests with spoofed MAC address simultaneously.
  - DoS attack using DHCP leases
- Rogue DHCP Server Attacks

Server runs out of IP addresses to allocate to valid users

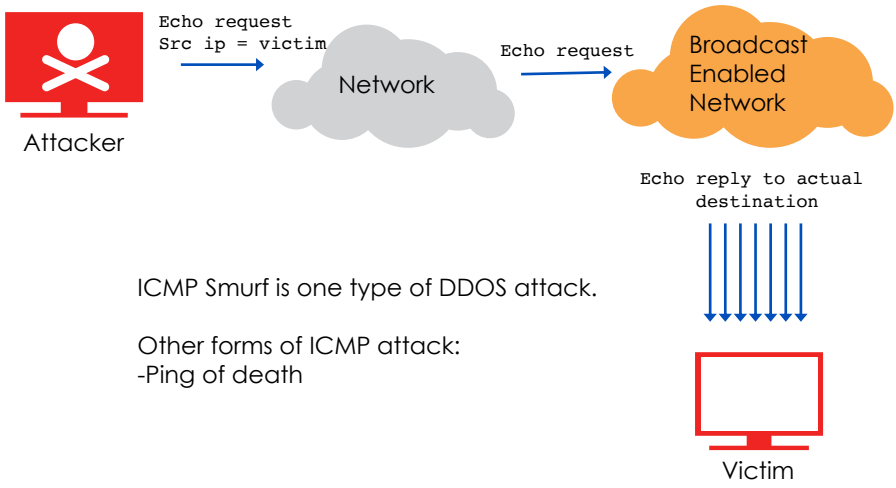


Attacker sends many different DHCP requests with many spoofed addresses.

Layer 3 Attacks

- ICMP Ping Flood
- ICMP Smurf
- Ping of death

ICMP Smurf



ICMP Smurf is one type of DDOS attack.

Other forms of ICMP attack:
-Ping of death

Routing Attacks

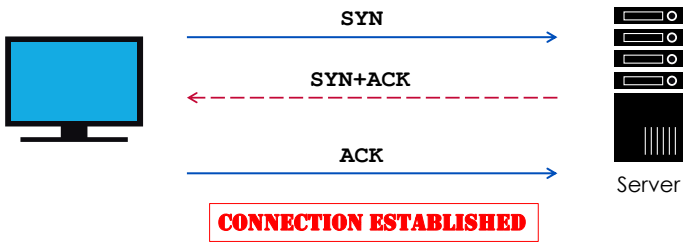
- Attempt to poison the routing information
- Distance Vector Routing
  - Announce 0 distance to all other nodes
    - Blackhole traffic
    - Eavesdrop
- Link State Routing
  - Can drop links randomly
  - Can claim direct link to any other routers
  - A bit harder to attack than DV
- BGP attacks
  - ASes can announce arbitrary prefix
  - ASes can alter path

TCP Attacks

- SYN Flood – occurs when an attacker sends SYN requests in succession to a target.
- Causes a host to retain enough state for bogus half-connections such that there are no resources left to establish new legitimate connections.

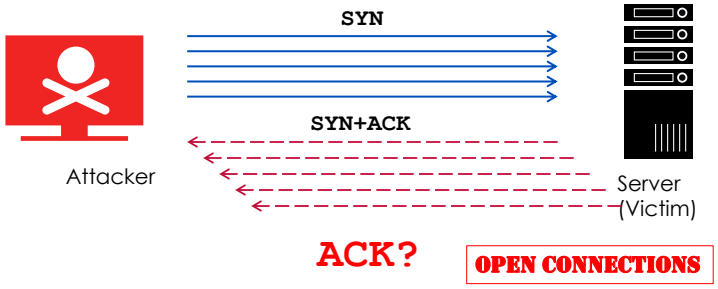
TCP Attacks

- Exploits the TCP 3-way handshake
- Attacker sends a series of SYN packets without replying with the ACK packet
- Finite queue size for incomplete connections



TCP Attacks

- Exploits the TCP 3-way handshake
- Attacker sends a series of SYN packets without replying with the ACK packet
- Finite queue size for incomplete connections



Application Layer Attacks

- Scripting vulnerabilities
- Cookie poisoning
- Buffer overflow
- Hidden field manipulation
- Parameter tampering
- Cross-site scripting
- SQL injection

DoS

- A Denial of Service attack aims to disrupt the availability of a service
- such as a machine or network resource by
  - Flooding
    - Bandwidth
    - number of connections
    - …
  - crashing the service

Nowadays also known as stress tests

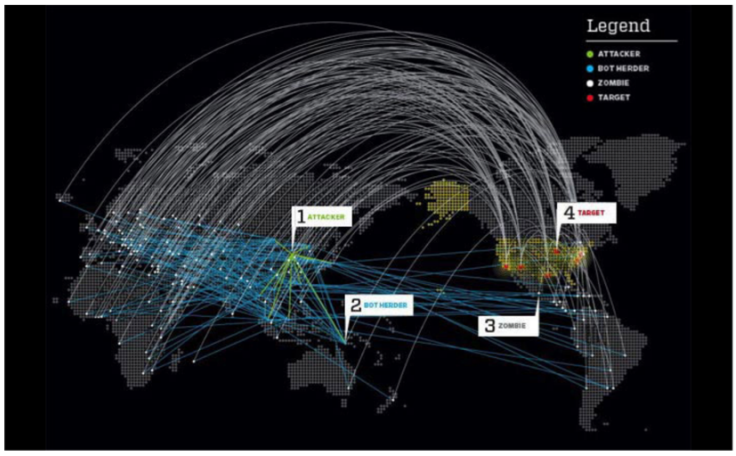
Layer 7 DDoS Attack

- Traditional DoS attacks focus on Layer 3 and Layer 4
- In Layer 7, a DoS attack is targeted towards the applications disguised as legitimate packets
- The aim is to exhaust application resources (bandwidth, ports, protocol weakness) rendering it unusable
- Includes:
  - HTTP GET
  - HTTP POST
  - Slowloris
  - LOIC / HOIC
  - RUDY (R-U-Dead Yet)

Layer 7 DDoS – Slowloris

- Incomplete HTTP requests
- Properties
  - Low bandwidth
  - Keep sockets alive
  - Only affects certain web servers
  - Doesn't work through load balancers
  - Managed to work around accf_http

Distributed Denial of Service attack

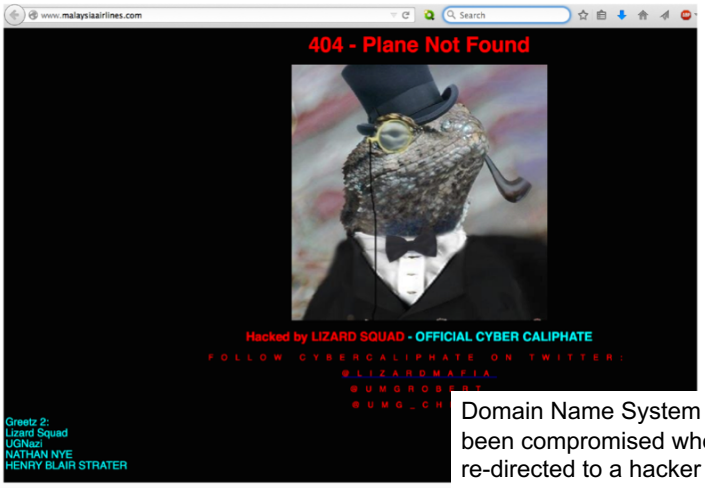


Botnet Mariposa 2009: 13 Mio *zombies*

Source: rivalhost 2013

DNS Attack Example

On 26th Jan 2015,



Domain Name System (DNS) has been compromised where users are re-directed to a hacker website

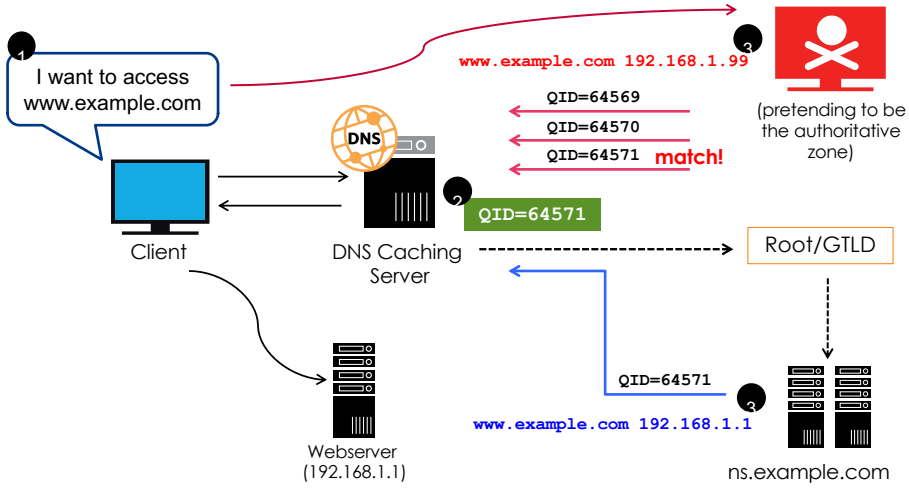
DNS Changer

- "Criminals have learned that if they can control a user's DNS servers, they can control what sites the user connects to the Internet."
- How: infect computers with a malicious software (malware)
- This malware changes the user's DNS settings with that of the attacker's DNS servers
- Points the DNS configuration to DNS resolvers in specific address blocks and use it for their criminal enterprise

DNS Cache Poisoning

- Caching incorrect resource record that did not originate from authoritative DNS sources.
- Result: connection (web, email, network) is redirected to another target (controlled by the attacker)

DNS Cache Poisoning



Best Practices

- Preventing Unauthorised changes / Transfer
  - Registry Lock Services
  - 2 Factor Authentication
- DNS Sec
  - Can be used to protect the communication between authoritative servers, and between authoritative servers and cache servers.

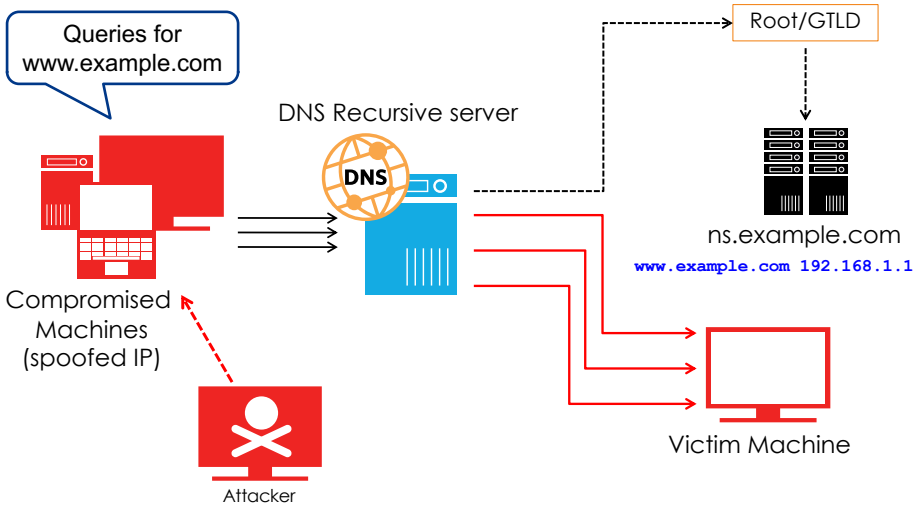
Amplification Attacks

- Distributed Reflection Denial of Service attack
  - No need for a botnet, just use existing servers with UDP services.
  - Some services can be misused because they amplify the request: DNS, NTP, SNMP, …
  - 1 small query in, 1 large answer out
  - This misuse can be avoided by disabling specific options or implementing firewall rules.
- Typical amplification factors
  - DNS: ~50-100
  - NTP: ~500-5000
  - SNMP: ~6-12

DNS Amplification Attack

- A type of reflection attack combined with amplification
  - Source of attack is reflected off another machine
  - Traffic received is bigger (amplified) than the traffic sent by the attacker
- UDP packet's source address is spoofed

DNS Amplification



NTP Amplification

- Network Time Protocol (NTP)
- Port 123/UDP
- Exploits NTP versions older than v4.2.7
  - monlist
- Several incidents in 2014

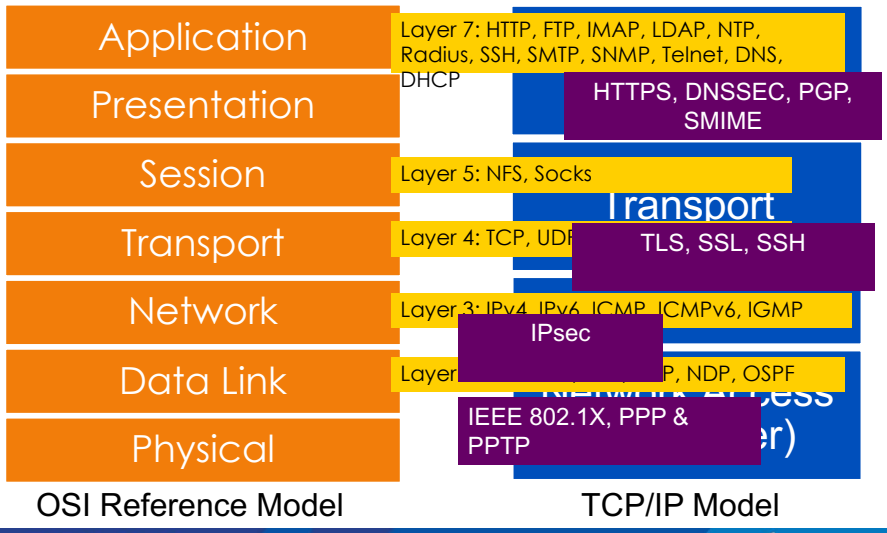
Wireless Attacks

- WEP – first security mechanism for 802.11 wireless networks
- Weaknesses in this protocol were discovered by Fluhrer, Mantin and Shamir, whose attacks became known as “FMS attacks”
- Tools were developed to automate WEP cracking
- Chopping attack were released to crack WEP more effectively and faster
- Cloud-based WPA crackers might speed it up

Man in the Middle Attacks (Wireless)

- Creates a fake access point and have clients authenticate to it instead of a legitimate one.
- Capture traffic to see usernames, passwords, etc that are sent in clear text.

Attacks on Different Layers



Link-Layer Security

- Layer 2 Forwarding (L2F)
- Point-to-Point Tunneling Protocol (PPTP)
- Layer 2 Tunneling Protocol (L2TP)

Transport Layer Security

- Secure Socket Layer (SSL)
- Secure Shell Protocol

Application Layer Security

- HTTPS
- PGP (Pretty Good Privacy)
- SMIME (Secure Multipurpose Internet Mail Extensions)
- TSIG and DNSSEC
- Wireless Encryption - WEP, WPA, WPA2

Overview

- Network Security Fundamentals
- Threat Pragmatics
- **Cryptography Basics**
- SSH
- Network Infrastructure Filtering at the border
- PGP
- TLS/SSL
- IPSec
- IDS & Snort
- Wireshark

Cryptography

- ?

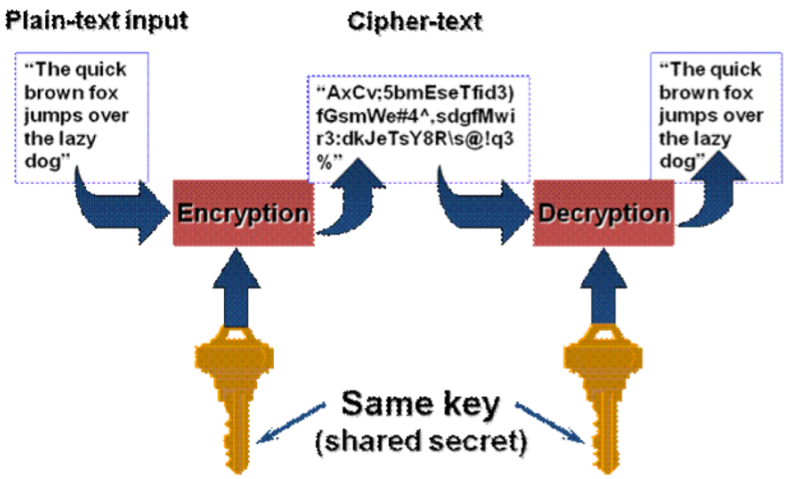
Cryptography

- Cryptography deals with creating documents that can be shared secretly over public communication channels
- Other terms closely associated
  - Cryptanalysis = code breaking
  - Cryptology
    - Kryptos (hidden or secret) and Logos (description) = secret speech / communication
    - combination of cryptography and cryptanalysis
- Cryptography is a function of plaintext and a cryptographic key

$$C = F(P,k)$$

Notation:
Plaintext (P)
Ciphertext (C)
Cryptographic Key (k)

Encryption & Decryption



Terminology

- Cryptography : the practice and study of hiding information
- Cryptanalysis : to find some weakness or insecurity in a cryptographic scheme
- Encryption : the method of transforming data (plain text) into an unreadable format
- Cipher text - the “scrambled” format of data after being encrypted

Cryptosystem Terminology

- Decryption : the method of turning cipher text back into plaintext
- Encryption Algorithm : a set of rules or procedures that dictates how to encrypt and decrypt data, also called encryption cipher
- Key : (cryptovariable) a value used in the encryption process to encrypt and decrypt

Cryptosystem Terminology

- Key Space : the range of possible values used to construct keys
- Example :
  - key can be 4 digits (0-9)
  - key space = 10,000
  - key can be 6 digits
  - key space = 1,000,000
- Key Clustering : when two different key generate the same cipher text from the same plaintext
- Work Factor : estimated time and resources to break a cryptosystem

Cryptosystem Development

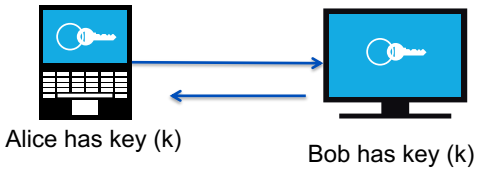
- Open algorithms to review
- Assume the attacker knows your encryption/decryption algorithm
- The only thing that should be secret in a cryptosystem is the key - Kirchhoff's Principle

Work Factor

- The amount of processing power and time to break a crypto system
- No system is unbreakable
- Make it too expensive to break

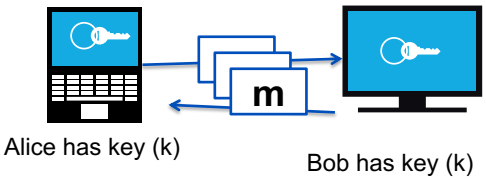
Crypto Core

- Secure key establishment



- Secure communication

Confidentiality and integrity



Source: Dan Boneh, Stanford

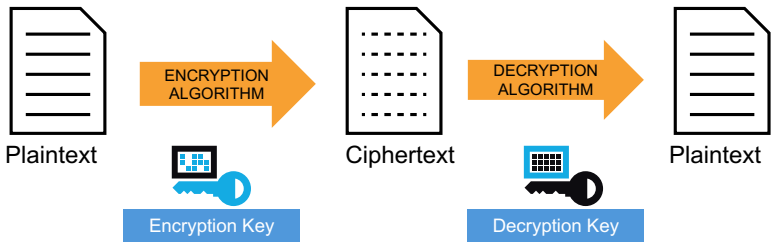
Kerckhoff's Law (1883)

- The system must not be required to be secret, and it must be able to fall into the hands of the enemy without inconvenience.
- In other words, the security of the system must rest entirely on the secrecy of the key.

Encryption

- process of transforming plaintext to ciphertext using a cryptographic key
- Used all around us
  - In Application Layer – used in secure email, database sessions, and messaging
  - In session layer – using Secure Socket Layer (SSL) or Transport Layer Security (TLS)
  - In the Network Layer – using protocols such as IPsec

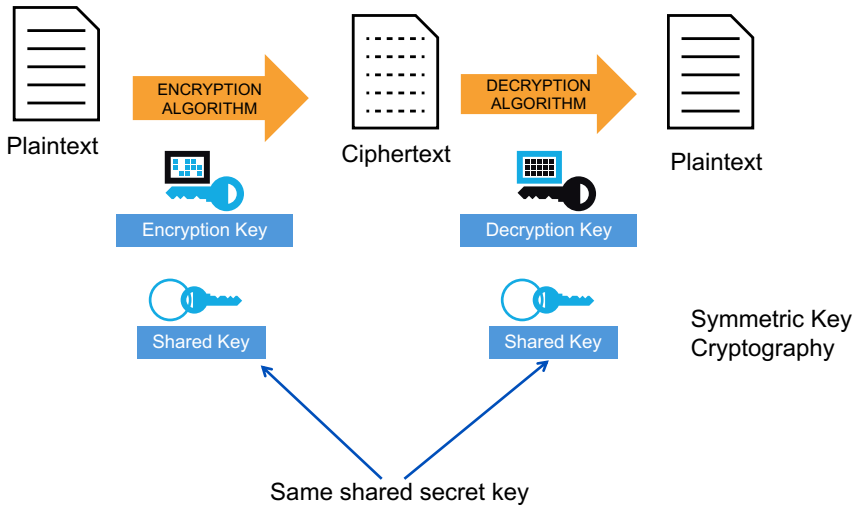
Encryption and Decryption



Symmetric Key Algorithm

- Uses a single key to both encrypt and decrypt information
- Also known as a secret-key algorithm
  - The key must be kept a “secret” to maintain security
  - This key is also known as a private key
- Follows the more traditional form of cryptography with key lengths ranging from 40 to 256 bits.
- Examples:
  - DES, 3DES, AES, RC4, RC6, Blowfish

Symmetric Encryption



Symmetric Key Algorithm

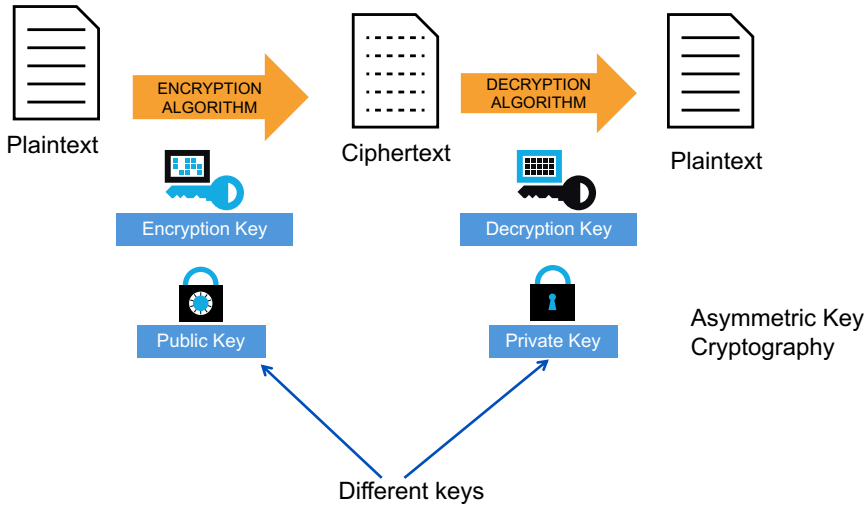
Symmetric Algorithm	Key Size
DES	56-bit keys
Triple DES (3DES)	112-bit and 168-bit keys
AES	128, 192, and 256-bit keys
IDEA	128-bit keys
RC2	40 and 64-bit keys
RC4	1 to 256-bit keys
RC5	0 to 2040-bit keys
RC6	128, 192, and 256-bit keys
Blowfish	32 to 448-bit keys

Note:
Longer keys are more difficult to crack, but more computationally expensive.

Asymmetric Key Algorithm

- Also called public-key cryptography
  - Keep private key private
  - Anyone can see public key
- separate keys for encryption and decryption (public and private key pairs)
- Examples:
  - RSA, DSA, Diffie-Hellman, ElGamal, PKCS

Asymmetric Encryption



Asymmetric Key Algorithm

- RSA – the first and still most common implementation
- DSA – specified in NIST's Digital Signature Standard (DSS), provides digital signature capability for authentication of messages
- Diffie-Hellman – used for secret key exchange only, and not for authentication or digital signature
- ElGamal – similar to Diffie-Hellman and used for key exchange
- PKCS – set of interoperable standards and guidelines

Digital Signature

- a message appended to a packet
- used to prove the identity of the sender and the integrity of the packet
- how it works:
  - sender signs the message with own private key
  - receiver uses the sender's public key to verify the signature

PKI / PGP Primer

- Public Key
- Private Key
- Message

- + = Encrypted
- + = Decrypted
- + = Signed
- + = Authenticated

Use Case

- email
  - encrypting: to send confidential information
  - signing: to prove the message actually comes from you and is not modified during delivery
- file distribution
  - signing: to prove the contents is distributed by you and not modified since signed
  - you can generate separate signature file if needed
    - you have the original file and signature file for it

Hash Functions

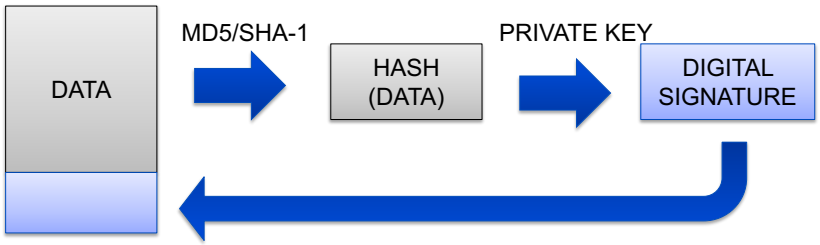
- produces a condensed representation of a message
- takes an input message of arbitrary length and outputs fixed-length code
  - The fixed-length output is called the hash or message digest
- A form of signature that uniquely represents the data
- Uses:
  - Verifying file integrity
  - Digitally signing documents
  - Hashing passwords

Hash Functions

- Message Digest (MD) Algorithm
  - Outputs a 128-bit fingerprint of an arbitrary-length input
  - MD4 is obsolete, MD5 is widely-used
- Secure Hash Algorithm (SHA)
  - SHA-1 produces a 160-bit message digest similar to MD5 (Please stop using SHA-1)
  - Widely-used on security applications (TLS, SSL, PGP, SSH, S/MIME, IPsec)
  - SHA-256, SHA-384, SHA-512 can produce hash values that are 256, 384, and 512-bits respectively

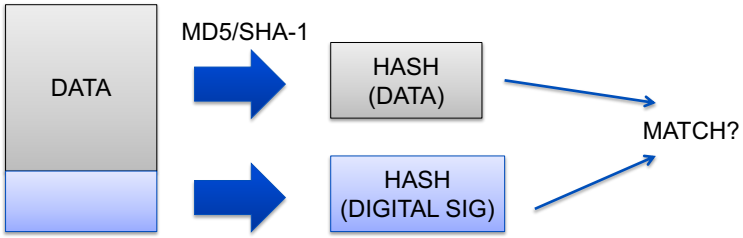
Digital Signature Process

- Hash the data using one of the supported hashing algorithms (MD5, SHA-1, SHA-256)
- Encrypt the hashed data using the sender's private key
- Append the signature (and a copy of the sender's public key) to the end of the data that was signed)



Signature Verification Process

- Hash the original data using the same hashing algorithm
- Decrypt the digital signature using the sender's public key. All digital signatures contain a copy of the signer's public key
- Compare the results of the hashing and the decryption. If the values match then the signature is verified. If the values do not match, then the data or signature was probably modified.



How to Attach the Signature

- appends it to the original file
- clearsign

```
-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA1

hello
-----BEGIN PGP SIGNATURE-----
Version: GnuPG v1

iQCVAwUBVsDzeDoo7EGKtqONAQKLFQQApJs+32OcZFZUdzQAO6GT8px6+F20CmO/
hInDACZnM2mvKP34J+fdslYyZWaivlhcaYeQsel+yvVjlO5NOLkdpjyOHw0ce99a
kAOP5cvSzw+fxGLkegrM3IhVCHlinKzLswpRCGOWP4xyAi1qaNPoykUzuInvnZ3u
3dVssbaXSN0=
=za1/
-----END PGP SIGNATURE-----
```

- https://www.gpg4win.org/ for windows
- https://www.gpgtools.org/ for OS X

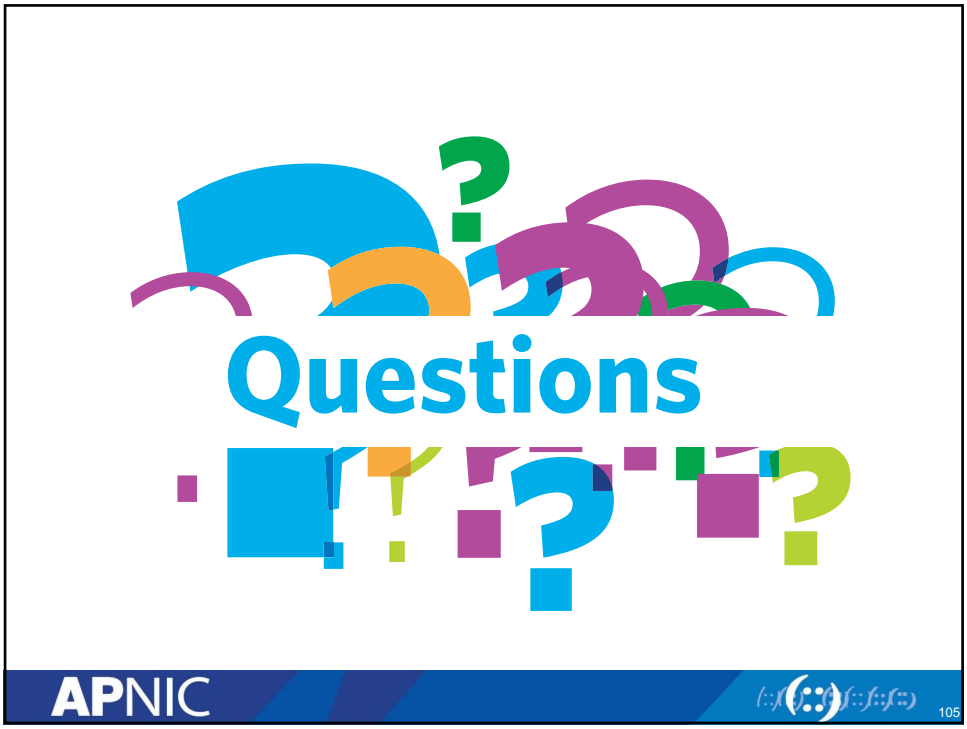
SSLLabs – Handson

- Do People Configure TLS/SSL (HTTPS) Correctly?

- 1. Go to https://www.ssllabs.com/ssltest/
- 2. Insert URL
- 3. What grade did you get and why?
- 4. How do we fix it?

Recommended References

- 1. Better Crypto
  - https://www.bettercrypto.org
- 2. Bulletproof SSL & TLS
  - https://www.feistyduck.com/books/bulletproof-ssl-and-tls/



Overview

- Network Security Fundamentals
- Threat Pragmatics
- Cryptography Basics
- **SSH**
- Network Infrastructure
  Filtering at the border
- PGP
- TLS/SSL
- IPSec
- IDS & Snort
- Wireshark

APNIC

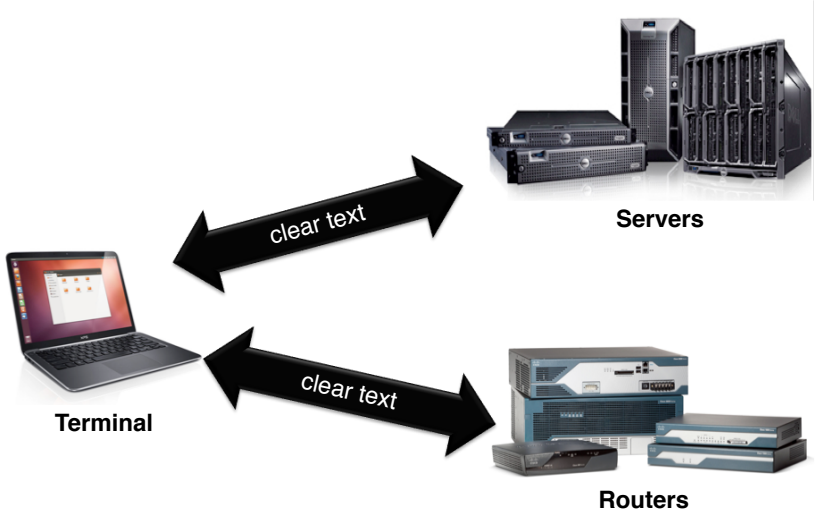
106

What is “Safely”

- **Authentication** – I am Assured of Which Host I am Talking With
- **Authentication** - The Host Knows Who I Am
- The Traffic is **Encrypted**

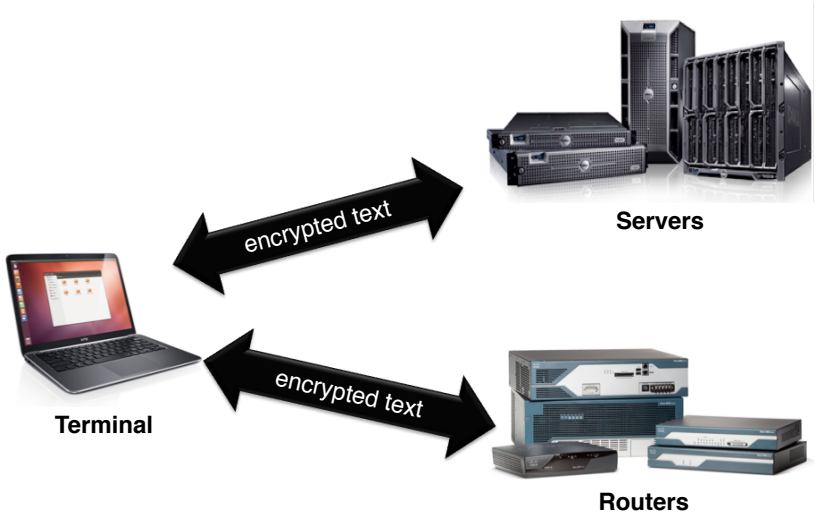
APNIC

Traditional (Telnet)



APNIC

Encrypted (SSH)



Secure Shell (SSH)

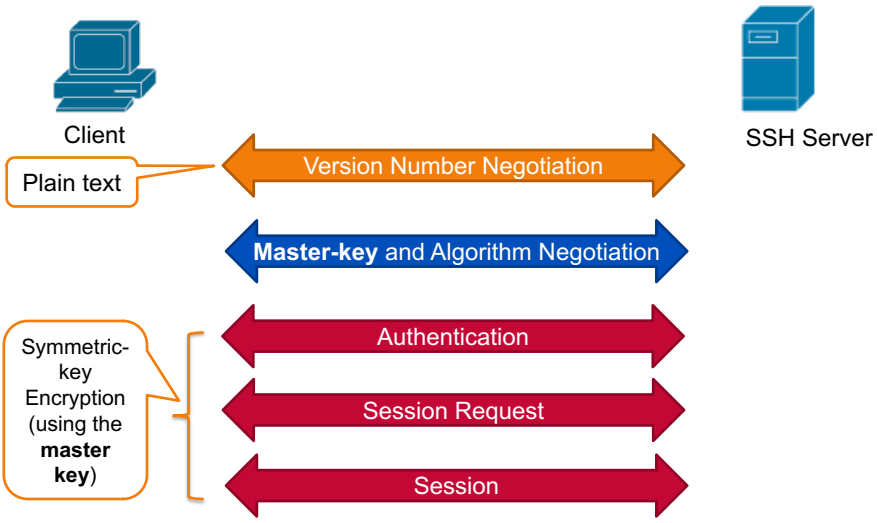
- Provides authenticated and encrypted shell access to a remote host
- Client-server model
- TCP 22

- It's not only a secure shell; it is much more
  - Transport protocol (eg. SCP, SFTP, SVN)
  - Connection forwarder.
    - You can use it to build custom tunnels

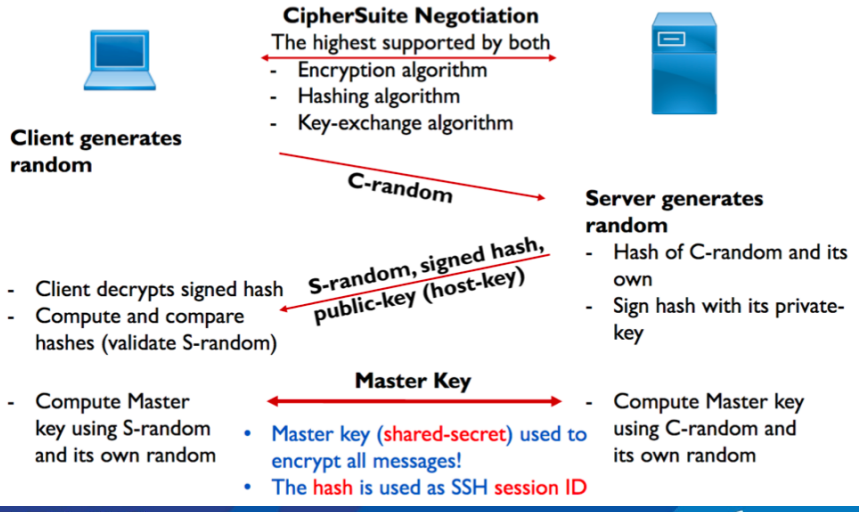
Secure Shell (SSH)

- Client-server crypto handshake
- Generate a symmetric key to secure the transport
- Client authenticates to server securely
- Secure communication

SSH – How it works



SSH- How it works



SSH Authentication

- Client sends its username to server over the secure channel
  - master key
- Server checks if the username exists in the local database
  - If username not valid, tear down the SSH session!
  - If valid, the server sends back authentication method
    - Password
    - Public key

SSH (Password Authentication) (1)

Terminal

Access Request

Server

Server's Public Key

```
$ ssh user@host
    The authenticity of host 'host (192.168.29.1)' can't be established.
    RSA key fingerprint is
98:2e:d7:e0:de:9f:ac:67:28:c2:42:2d:37:16:58:4d.
    Are you sure you want to continue connecting (yes/no)?
```

APNIC

SSH (Password Authentication)

Terminal

Password encrypted by Server's Public Key

Server

Server uses its Private Key to decrypt the message. Compare with the password stored on server.

Access granted

APNIC

Password Authentication

- Password Authentication is that it's simple to set up - usually the default - and is easy to understand.
- Allows brute-force password guessing.
- Passwords must be remembered and entered separately upon every login.

Public Key Access

- User creates a pair of public and private keys.
- The **public key** - nonsensitive information.
- The **private key** - is protected on the local machine by a strong passphrase.
- Installs the public key in his **`$HOME/.ssh/authorized_keys`** file on the target server.
- This key must be installed on the target system - one time.

Public Key Access

1. The user makes an initial connection and sends a username along with a request to use a key.

2. The ssh daemon on the server looks in the user's authorized_keys file, constructs a challenge based on the public key found there, and sends this challenge back to the user's ssh client.

3. The ssh client receives the key challenge. It finds the user's private key on the local system, but it's protected by an encrypting passphrase.

4. The user is prompted for the passphrase to unlock the private key.

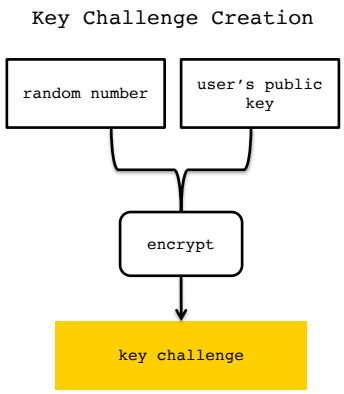
5. ssh uses the private key to construct a key response, and sends it to the waiting sshd on the other end of the connection. **It does not send the private key itself!**

6. sshd validates the key response, and if valid, grants access to the system.

How key challenge work (Under the hood)

1. User ssh to server, he presents his username to the server with a request to set up a key session.

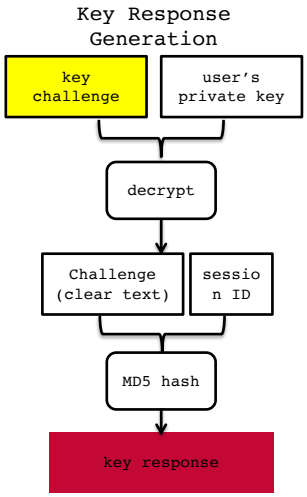
2. The server creates a "challenge". It creates and remembers a large random number, then encrypts it with the user's public key.



How key challenge work (Under the hood)

3. Agent decrypts it with the private key and get the random number generated by the server.

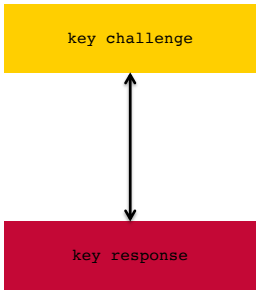
4. The agent takes this random number, appends the previously negotiated SSH session ID and creates an MD5 hash value of the resultant string: this result is sent back to the server as the key response.



How key challenge work (Under the hood)

5. The server computes the same MD5 hash (random number + session ID) and compares it with the key response from the agent.

6. If they match, the user must have been in possession of the private key, and access is granted.



Public Key Access

- Public keys cannot be easily brute-forced.
- The same private key (with passphrase) can be used to access multiple systems: no need to remember many passwords.
- Requires one-time setup of public key on target system.
- Requires unlocking private key with secret passphrase upon each connection.

Public Key Access

- Never store Private Key on a multi-user host.
- Store Private Key ONLY on your laptop and protect your laptop (Encrypt Disk!).
- It is OK to use SSH_AGENT to remember your key ONLY if your laptop/computer locks very quickly.

Private Key on Unix / MacOSX

- SSH is Built In
  - UNIX
  - Linux
  - MacOS X

Generate Key (Unix / MacOSX)

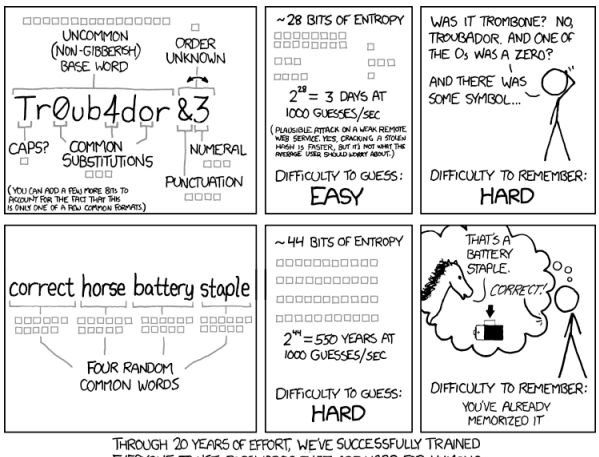
```
$/usr/home/foo> ssh-keygen -t rsa -b 4096 -C your_email@example.com
Generating public/private rsa key pair.
Enter file in which to save the key (/usr/home/foo/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /usr/home/foo/.ssh/id_rsa.
Your public key has been saved in /usr/home/foo/.ssh/id_rsa.pub.
The key fingerprint is:
27:99:35:e4:ab:9b:d8:50:6a:8b:27:08:2f:44:d4:20 foo@bdnog.org
```

Generate Key (Unix / MacOSX)

~/.ssh/id_rsa: The private key. DO NOT SHARE THIS FILE!

~/.ssh/id_rsa.pub: The associated public key. This can be shared freely without consequence.

Password vs Passphrase



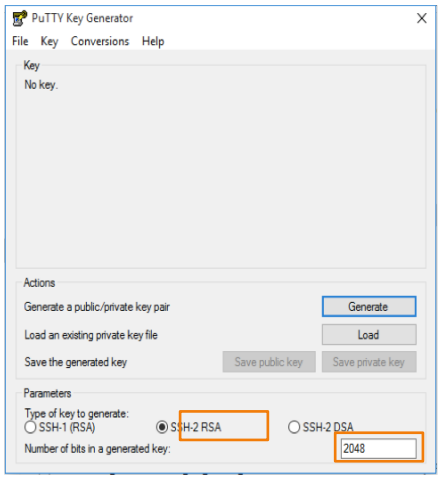
source : http://xkcd.com/936/

Private Key on Windows

- http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html
  - PuTTY (the Telnet and SSH client itself)
  - PuTTYgen (an RSA and DSA key generation utility).
  - Pageant (an SSH authentication agent for PuTTY, PSCP, PSFTP, and Plink)

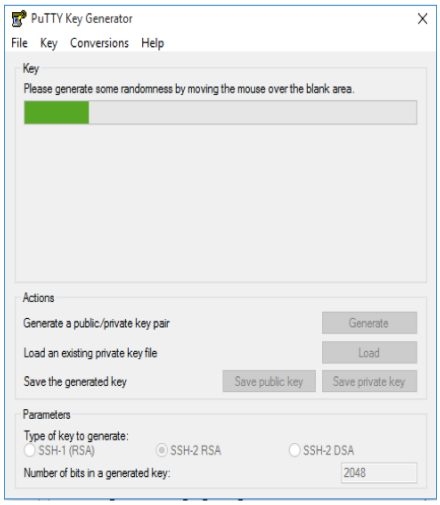
Generate Key (Windows)

1. Run PuttyGen



Generate Key (Windows)

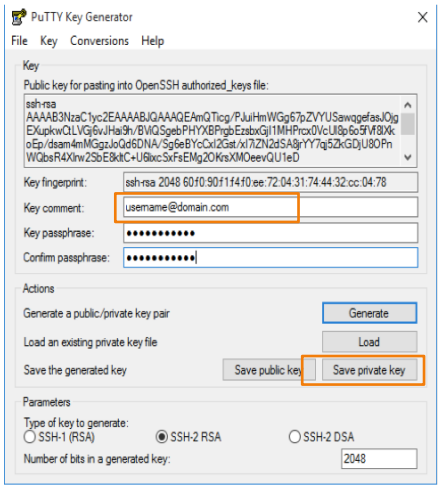
2. Generate Key



Generate Key (Windows)

3. Enter Passphrase & Save Private Key

4. Right-click in the text field labeled Public key for pasting into OpenSSH authorized_keys file and choose Select All and copy the key

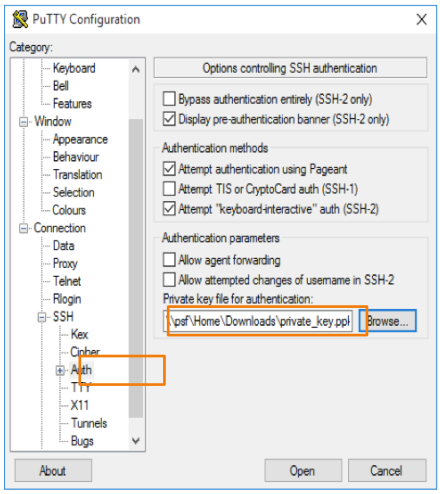


Putting the Key on the Target Host

- You can copy the public key into the new machine's authorized_keys file with the ssh-copy-id command
  `ssh-copy-id user@serverip`
- Alternatively, you can paste in the keys using SSH:
  `cat ~/.ssh/id_rsa.pub | ssh user@serverip "mkdir -p ~/.ssh && cat >> ~/.ssh/authorized_keys"`

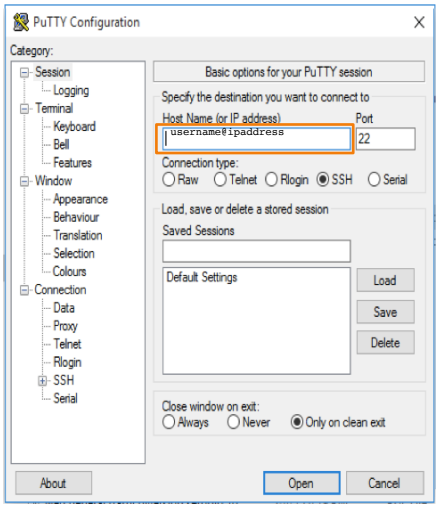
Generate Key (Windows)

4. Load Key in Putty



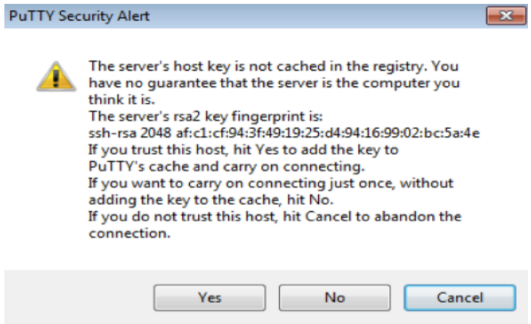
Generate Key (Windows)

5. SSH to host



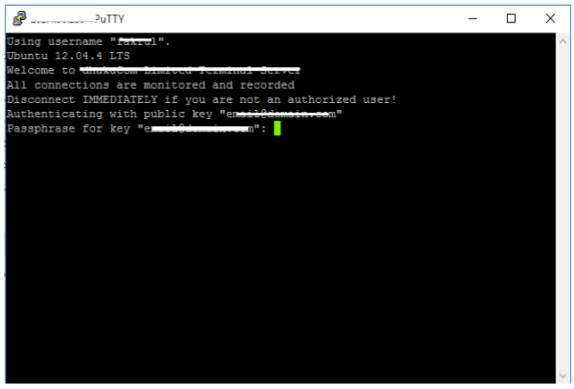
Generate Key (Windows)

6. Accept Host's Key



Generate Key (Windows)

7. passphrase for Key



PuTTY Agent: Pageant

- Select Add Key, browse to your key, select, enter passphrase
- Enter passphrase again. Eventually you'll get it right.
- SSH to your server
- PuTTY enable/disable agent: `Connection -> SSH -> Auth, "Attempt Authentication using Pageant"` checkbox

Exercise

- Create your key
- Follow the lab manual ssh-lab.pdf

APNIC

Overview

- Network Security Fundamentals
- Threat Pragmatics
- Cryptography Basics
- SSH
- **Network Infrastructure Filtering at the border**
- PGP
- TLS/SSL
- IPSec
- IDS & Snort
- Wireshark

APNIC

140

What we have in network?

- Router
- Switch
- CPE (ADSL Router / WiFi Router)
- Servers
- PC/Laptop
- Smart Phone

APNIC

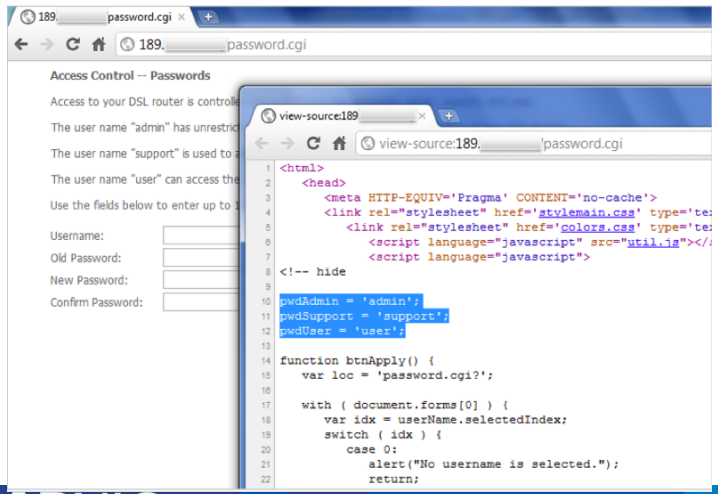
- **Securing The Device**

APNIC

Think of ALL Devices

- The following problem was recently reported and affects low-end CPEs (ADSL connections only)
  - Admin password exposed via web interface
  - Allow WAN management (this means anyone on Internet)
  - Bug fixed and reintroduced depending on the firmware version
- The bug is quite a number of years old

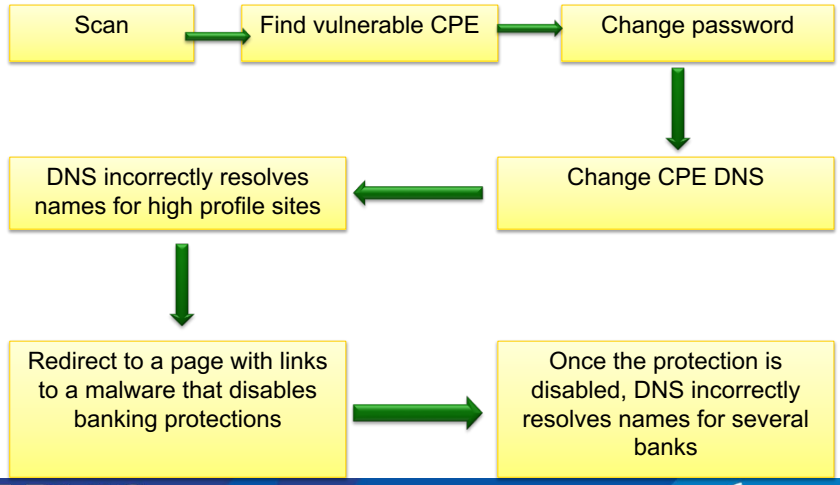
Password Visible via Web Interface



Magnitude of Problem

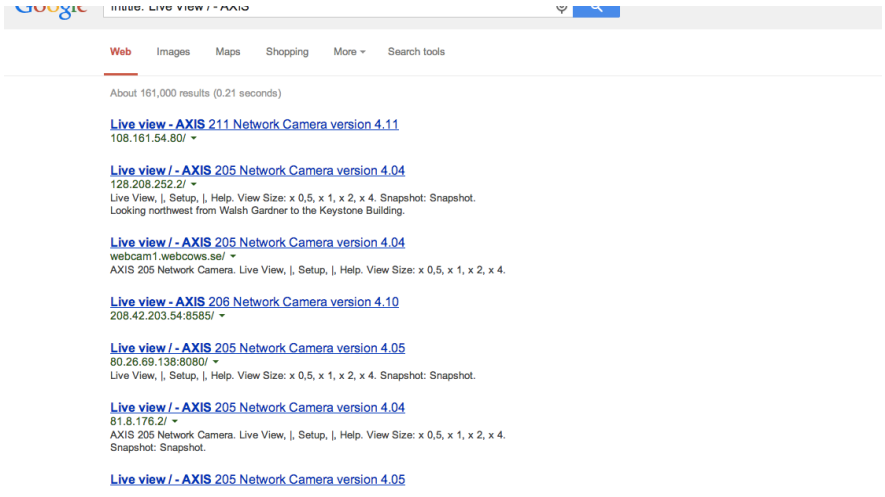
- 4.5 Million CPEs (ADSL Modems) using a unique malicious DNS
- In early 2012 more than 300,000 CPEs still infected
- 40 malicious DNS servers found

Implication of CPEs Exploited



Allow remote access

Finding out open IPcam!!!



And more…..

IOActive Lights Up Vulnerabilities for Over Half a Million Belkin WeMo Users
Popular home automation devices are wide open to attackers

Seattle, US — February 18, 2014 — IOActive, Inc., the leading global provider of specialist information security services, announced today that it has uncovered multiple vulnerabilities in Belkin WeMo Home Automation devices that could affect over half a million[1] users. Belkin's WeMo uses Wi-Fi and the mobile Internet to control home electronics anywhere in the world directly from the user's smartphone.

Mike Davis, IOActive's principal research scientist, uncovered multiple vulnerabilities in the WeMo product set that gives attackers the ability to:

- Remotely control WeMo Home Automation attached devices over the Internet
- Perform malicious firmware updates
- Remotely monitor the devices (in some cases)
- Access an internal home network

- **Could device hardening have made a difference?**

Device Access Control (Physical)

- Lock up the server room. Equipment kept in highly restrictive environments
- Set up surveillance
- Make sure the most vulnerable devices are in that locked room
- Keep intruders from opening the case
- Protect the portables
- Pack up the backups
- Disable the drives
- Social engineering training and awareness
- Console access
  - password protected
  - access via OOB (Out-of-band) management
  - configure timeouts

Device Access Control (Logical)

- Set passwords to something not easily guessed
- Use single-user passwords (avoid group passwords)
- Encrypt the passwords in the configuration files
- Use different passwords for different privilege levels
- Use different passwords for different modes of access
- IF AVAILABLE – use digital certificate based authentication mechanisms instead of passwords

Management Plane Filters

- Authenticate Access
- Define Explicit Access To/From Management Stations
  - SNMP
  - Syslog
  - TFTP
  - NTP
  - AAA Protocols
  - SSH, Telnet, etc.

Securing SNMP

```
access-list 99 permit 192.168.1.250
access-list 99 permit 192.168.1.240

snmp-server community N3T-manag3m3nt ro 99
```

Securing SSH

```
ipv6 access-list AUTHORIZED_IPV6_HOST
 permit ipv6 host 2405:7600:0:6::250 any
 deny ipv6 any any log
!
ip access-list extended AUTHORIZED_IPV4_HOST
 permit tcp host 103.21.75.5 any eq 22
 deny   tcp any any log
!
line vty 0 4
 access-class AUTHORIZED_IPV4_HOST in
 ipv6 access-class AUTHORIZED_IPV6_HOST in
```


Secure Access with Passwords and Logout Timers

```
line console 0
    login
    password console-pw
    exec-timeout 1 30
!
line vty 0 4
    login
    password vty-pw
    exec-timeout 5 00
!
enable secret enable-secret
username bob secret bob-secret
```

Never Leave Passwords in Clear-Text

- ***service password-encryption*** command
- ~~***password*** command~~
  - Will encrypt all passwords on the Cisco IOS with Cisco-defined encryption type “7”
  - Use “*command* password 7 <password>” for cut/paste operations
  - Cisco proprietary encryption method
- ***secret*** command
  - Uses MD5 to produce a one-way hash
  - Cannot be decrypted
  - Use “*command* secret 5 <password>” to cut/paste another “enable secret” password

Authenticate Individual Users

```
username mike secret mike-secret
username john secret john-secret
username chris secret chris-secret
!
~~username staff secret group-secret~~
```

Radius Authentication (AAA)

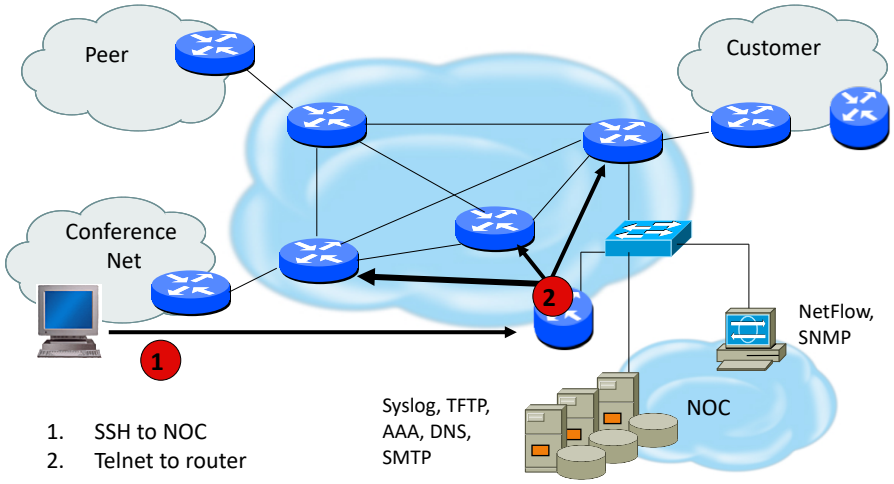
```
aaa new-model
!
aaa authentication login default group radius
local
aaa authorization exec default group radius
local
!
radius-server host 192.168.1.250 auth-port
1812 acct-port 1813
radius-server key 7 0130310759262E000B69560F
```


Restrict Access To Trusted Hosts

- Use filters to specifically permit hosts to access an infrastructure device
- Example

```
access-list 103 permit tcp host 192.168.200.7
  192.168.1.0 0.0.0.255 eq 22 log-input
access-list 103 permit tcp host 192.168.200.8
  192.168.1.0 0.0.0.255 eq 22 log-input
access-list 103 permit tcp host 192.168.100.6
  192.168.1.0 0.0.0.255 eq 23 log-input
access-list 103 deny ip any any log-input
!
line vty 0 4
access-class 103 in
transport input ssh
```

Telnet using SSH 'Jumphost'



1. SSH to NOC
2. Telnet to router

APNIC

Banner – What Is Wrong ?

```
banner login ^C

     You should not be on this device.

     Please Get Off My Router!!
^C
```

APNIC

More Appropriate Banner

```
!!!!  WARNING !!!!
You have accessed a restricted device.
All access is being logged and any
unauthorized access will be prosecuted to the
full extent of the law.
```


Centralized Log (syslog)

```
Router(config)# logging 192.168.0.30
Router(config)# logging trap 3
Router(config)# logging facility local3
```

Trap:	**Facility:**
Emergency: 0	local0
Alert: 1	Local1
Critical: 2	Local2
Error: 3	Local3
Warning: 4	Local4
Notice: 5	Local5
Informational: 6	Local6
Debug: 7	and local7

Configuration change logging

```
Router# configure terminal
Router(config)# archive
Router(config-archive)# log config
Router(config-archive-log-config)# logging enable
Router(config-archive-log-config)# logging size 200
Router(config-archive-log-config)# hidekeys
Router(config-archive-log-config)# notify syslog
```

768962: Feb 1 20:59:45.081 UTC: %PARSER-5-CFGLOG_LOGGEDCMD: User:fakrul logged command:!exec: enable

768963: Feb 1 21:03:17.160 UTC: %PARSER-5-CFGLOG_LOGGEDCMD: User:fakrul logged command:no ipv6 prefix-list dhakacom_AS23956_IN_IPv6 description

768965: Feb 1 21:03:19.182 UTC: %SYS-5-CONFIG_I: Configured from console by fakrul on vty0 (2405:7600:0:6::250)

Turn Off Unused Services

Feature	Description	Default	Recommendation	Command
CDP	Proprietary layer 2 protocol between Cisco devices	Enabled		`no cdp run`
TCP small servers	Standard TCP network services: echo, chargen, etc	11.3: disabled 11.2: enabled	This is a legacy feature, disable it explicitly	`no service tcp-small-servers`
UDP small servers	Standard UDP network services: echo, discard, etc	11.3: disabled 11.2: enabled	This is a legacy feature, disable it explicitly	`no service udp-small-servers`
Finger	Unix user lookup service, allows remote listing of logged in users.	Enabled	Unauthorized persons don't need to know this, disable it.	`no service finger`
HTTP server	Some Cisco IOS devices offer web-based configuration	Varies by device	If not in use, explicitly disable, otherwise restrict access	`no ip http server`
Bootp server	Service to allow other routers to boot from this one	Enabled	This is rarely needed and may open a security hole, disable it	`no ip bootp server`

Turn Off Unused Services

Feature	Description	Default	Recommendation	Command
PAD Service	Router will support X.25 packet assembler service	Enabled	Disable if not explicitly needed	`no service pad`
IP source routing	Feature that allows a packet to specify its own route	Enabled	Can be helpful in attacks, disable it	`no ip source-route`
Proxy ARP	Router will act as a proxy for layer 2 address resolution	Enabled	Disable this service unless the router is serving as a LAN bridge	`no ip proxy-arp`
IP directed broadcast	Packets can identify a target LAN for broadcasts	Enabled (11.3 & earlier)	Directed broadcast can be used for attacks, disable it	`no ip directed-broadcast`

Configuration (Templates)

```
!configure timezone
service timestamps debug uptime
service timestamps log datetime localtime
service password-encryption
clock timezone UTC +6

! turn off unnecessary services (global)
no ip domain-lookup
no cdp run
no ip http server
no ip source-route
no service finger
no ip bootp server
```

```
! turn off unnecessary services (interface)
Interface GigabitEthernet0/0
no ip redirects
no ip directed-broadcast
no ip proxy arp
no cdp enable

! turn on logging and snmp
logging 192.168.253.56
snmp-server communityTxo~QbW3XM ro 98
!
access-list 99 permit 192.168.253.0 0.0.0.255
access-list 99 deny any log
access-list 98 permit host 192.168.253.51
access-list 98 deny any log
!
```

Configuration (Templates)

```
line vty 0 4
access-class 99 in
exec-timeout 2 0
transport input ssh
!
line con 0
access-class 99 in
exec-timeout 2 0
!
banner motd #
!!!!  WARNING !!!!
You have accessed a restricted device.
All access is being logged and any
unauthorized access will be prosecuted to
```

```
!Turn on NTP
ntp authenticate
ntp authentication-key 1 md5 -
UN&/6[oh6
ntp trusted-key 1
ntp access-group peer 96
ntp server 192.168.254.57 key 1
access-list 96 permit host
192.168.254.57
access-list 96 deny any log
```

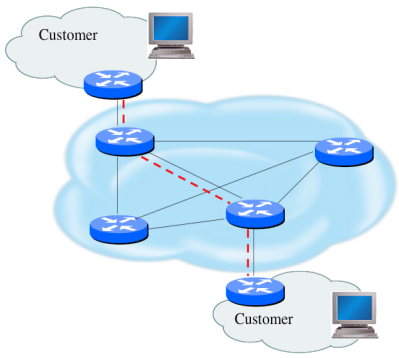

Fundamental Device Protection Summary

- Secure logical access to routers with passwords and timeouts
- Never leave passwords in clear-text
- Authenticate individual users
- Restrict logical access to specified trusted hosts
- Allow remote vty access only through ssh
- Disable device access methods that are not used
- Protect SNMP if used
- Shut down unused interfaces
- Shut down unneeded services
- Ensure accurate timestamps for all logging
- Create appropriate banners
- Test device integrity on a regular basis

- Securing The Data Path

APNIC

Securing The Data Path



- Filtering and rate limiting are primary mitigation techniques
- Edge filter guidelines for ingress filtering (BCP38/BCP84)
- Null-route and black-hole any detected malicious traffic
- Netflow is primary method used for tracking traffic flows
- Logging of Exceptions

APNIC

Data Plane (Packet) Filters

- Most common problems
  - Poorly-constructed filters
  - Ordering matters in some devices
- Scaling and maintainability issues with filters are commonplace
- Make your filters as modular and simple as possible
- Take into consideration alternate routes
  - Backdoor paths due to network failures

Filtering Deployment Considerations

- How does the filter load into the router?
- Does it interrupt packet flow?
- How many filters can be supported in hardware?
- How many filters can be supported in software?
- How does filter depth impact performance?
- How do multiple concurrent features affect performance?
- Do I need a standalone firewall?

General Filtering Best Practices

- Explicitly deny all traffic and only allow what you need
- The default policy should be that if the firewall doesn't know what to do with the packet, deny/drop it
- Don't rely only on your firewall for all protection of your network
- Implement multiple layers of network protection
- Make sure all of the network traffic passes through the firewall
- Log all firewall exceptions (if possible)

Filtering Recommendations

- Log filter port messages properly
- Allow only internal addresses to enter the router from the internal interface
- Block packets from outside (untrusted) that are obviously fake or commonly used for attacks
- Block packets that claim to have a source address of any internal (trusted) network.

Filtering Recommendations

- Block incoming loopback packets and RFC 1918 networks
  - 127.0.0.0
  - 10.0.0.0 – 10.255.255.255
  - 172.16.0.0 – 172.31.0.0
  - 192.168.0.0 – 192.168.255.255
- Block multicast packets (if NOT using multicast)
- Block broadcast packets (careful of DHCP & BOOTP users)
- Block incoming packets that claim to have same destination and source address

DoS Filtering

(* these networks were reallocated and are actually used)

Description	Network
default	0.0.0.0 /8
loopback	127.0.0.0 /8
RFC 1918	10.0.0.0 /8
RFC 1918	172.16.0.0 /12
RFC 1918	192.168.0.0 /16
Net Test	192.0.2.0 /24
Testing devices *	192.18.0.0 /15
IPv6 to IPv4 relay *	192.88.99.0 /24
RFC 1918 nameservers *	192.175.48.0 /24
End-node auto configuration *	169.254.0.0 /16

Example Incoming IPv4 Bogon Packet Filter

```
ip access-list extended DSL-Incoming
 deny   ip 127.0.0.0 0.255.255.255 any log
 deny   ip 10.0.0.0 0.255.255.255 any log
 deny   ip 169.254.0.0 0.0.255.255 any log
 deny   ip 172.16.0.0 0.15.255.255 any log
 deny   ip 192.168.0.0 0.0.255.255 any log
 deny   ip 224.0.0.0 15.255.255.255 any log
 permit icmp any any ttl-exceeded
 permit icmp any any echo-reply
 permit icmp any any echo
 permit tcp any any eq 22 log
 permit udp host <ip address> eq domain <subnet range>
 permit udp host <ip address> eq domain <subnet range>
 permit udp host <ip address>  <subnet range>  eq ntp
 permit udp host <ip address>  <subnet range>  eq ntp
 permit tcp any <my sybnet>  established
 deny   ip any any log
```


Example Incoming IPv4 Bogon Packet Filter

- Bogon and fullbogon peering use different ASNs
- Advertise all fullbogons (IPv4 and IPv6) over a single BGP peering session
- For details: http://www.team-cymru.org/Services/Bogons/bgp.html

RFC2827 (BCP38) – Ingress Filtering

- If an ISP is aggregating routing announcements for multiple downstream networks, strict traffic filtering should be used to prohibit traffic which claims to have originated from outside of these aggregated announcements.
- The ONLY valid source IP address for packets originating from a customer network is the one assigned by the ISP (whether statically or dynamically assigned).
- An edge router could check every packet on ingress to ensure the user is not spoofing the source address on the packets which he is originating.

Guideline for BCP38

- Networks connecting to the Internet
  - Must use inbound and outbound packet filters to protect network
- Configuration example
  - Outbound—only allow my network source addresses out
  - Inbound—only allow specific ports to specific destinations in

Techniques for BCP 38

- Static ACLs on the edge of the network
- Unicast RPF strict mode
- IP source guard

Example Inbound Packet Filter

```
access-list 121 permit ip 192.168.1.250
0.0.0.255 any
access-list 121 deny ip any any log
!
interface serial 1/1/1.3
    Description Link to XYZ
    ip access-group 121 in
```

Infrastructure Filters

- Permit only required protocols and deny ALL others to infrastructure space
  - Filters now need to be IPv4 and IPv6!
  - Applied inbound on ingress interfaces
- Basic premise: filter traffic destined TO your core routers
- Develop list of required protocols that are sourced from outside your AS and access core routers
  - Example: eBGP peering, GRE, IPSec, etc.
  - Use classification filters as required
- Identify core address block(s)
  - This is the protected address space
  - Summarization is critical for simpler and shorter filters

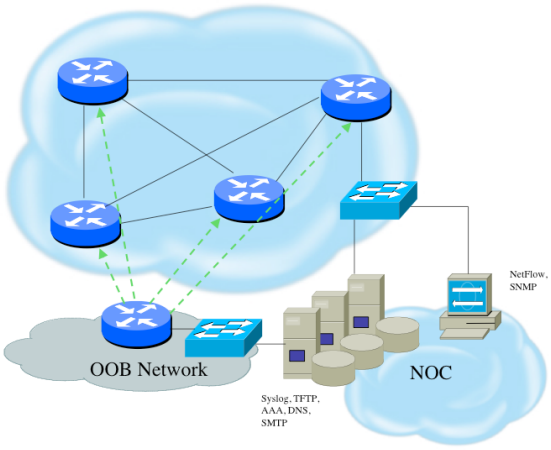
References

- Articles, documents and templates from Team CYMRU http://www.team-cymru.org/ReadingRoom/
- Google for the information specifics from the vendors you use: “<vendor> security template”

- Configuration and Archiving

APNIC

Device OOB Management



- Out-of-band device management should be used to ensure DoS attacks do not hinder getting access to critical infrastructure devices
- Dial-back encrypted modems are sometimes still used as backup

APNIC

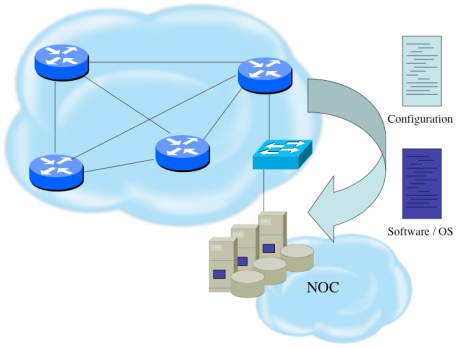
Device Management Common Practice

- SSH primarily used; Telnet only from jumphosts
- HTTP access explicitly disabled
- All access authenticated
  - Varying password mechanisms
  - AAA usually used
    - Different servers for in-band vs OOB
    - Different servers for device authentication vs other
    - Static username pw or one-time pw
  - Single local database entry for backup
- Each individual has specific authorization
- Strict access control via filtering
- Access is audited with triggered pager/email notifications
- SNMP is read-only
  - Restricted to specific hosts
  - View restricted if capability exists
  - Community strings updated every 30-90 days

System Images and Configuration Files

- Careful of sending configurations where people can snoop the wire
  - CRC or MD5 validation
  - Sanitize configuration files
- SCP(Secured Copy) should be used to copy files
  - TFTP and FTP should be avoided
- Use tools like ‘rancid’ to periodically check them against modified configuration files

Software and Configuration Upgrade / Integrity



- Files stored on specific systems with limited access
- All access to these systems are authenticated and audited
- SCP is used where possible; FTP is NEVER used; TFTP still used
- Configuration files are polled and compared on an hourly basis (RANCID)
- Filters limit uploading / downloading of files to specific systems
- Many system binaries use MD-5 checks for integrity
- Configuration files are stored with obfuscated passwords

Overview

- Network Security Fundamentals
- Threat Pragmatics
- Cryptography Basics
- SSH
- Network Infrastructure Filtering at the border
- **PGP**
- TLS/SSL
- IPSec
- IDS & Snort
- Wireshark

APNIC

193

PGP vs GPG vs OpenPGP

- **Pretty Good Privacy (PGP)** is proprietary software written by Phil Zimmerman and released in 1991
- **Gnu Privacy Guard (GPG)** is similar software released in 1999 under the GPL open source license.
- **OpenPGP** is an IETF standard with which both pieces of software are compliant.

APNIC

194

Security issues for E-mail

- Confidentiality
  - Network admin can read your e-mail.
  - Webmail provider can read your e-mail.
  - LAN user may read your e-mail by monitoring tool.
  - Even in some hotel, I could have chance to read other rooms internet traffic.
- Integrity
  - E-mail contents may be changed by some attacker on the network.
- Authenticity
  - Easy to set any e-mail headers like “From”.
  - Any other e-mail headers can be set anything you want.
  - Difficult to know it is true.

Targeted Attack

- Attacks on information security which seek to affect a specific organization or group, rather than indiscriminately. Some may be customized for a specific target organization or group.
  - An e-mail with suspicious file attached
  - Executable binary
  - Word document file
  - Database application file

Cryptography

- Symmetric and Asymmetric (public-key)
- The latter is widely accepted
- PGP is based on Asymmetric (Public-Key) Encryption

Symmetric Encryption

- Involves only one key, which is used by both the sender for encrypting and the recipient for decrypting
- Symmetric algorithms: blowfish, Triple-DES, AES (Advanced Encryption Standard), CAST (Carlisle Adams and Stafford Tavares) , IDEA (International Data Encryption Algorithm, legally restricted, but the other algorithms may be freely used)
- Problem: the means of distributing the key

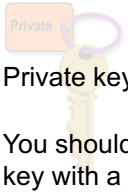
Asymmetric (Public-Key) Encryption

- Solves the problem of distributing keys by using one pair of complimentary keys, one public and the other private.
- Public: freely exchanged to others without fear of compromising security.
- Private: only you have access, should be carefully protected.
- A message is encrypted to a recipient using the recipient's public key, and it can only be decrypted using the corresponding private key.

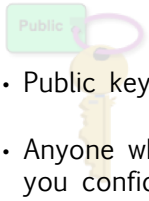
Asymmetric Encryption Refresher

- One key mathematically related to the other.
- Public key can be generated from private key. But NOT vice versa.
- If you encrypt data with the public key, you need private key to decrypt
- You can sign data with the private key and verify the signature using the public key

Keys



- Private key is kept SECRET.
- You should encrypt your private key with a symmetric passphrase.

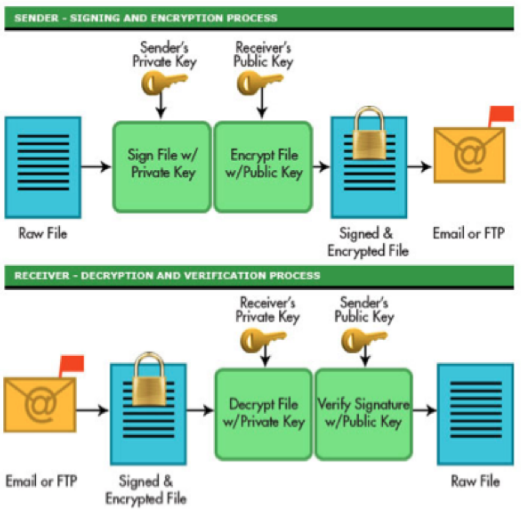


- Public key is distributed.
- Anyone who needs to send you confidential data can use your public key

Signing & Encrypting

- Data is encrypted with a public key to be decrypted with the corresponding private key.
- Data can be signed with the private key to be verified by anyone who has the corresponding public key.
- Since public keys are data they can be signed too.

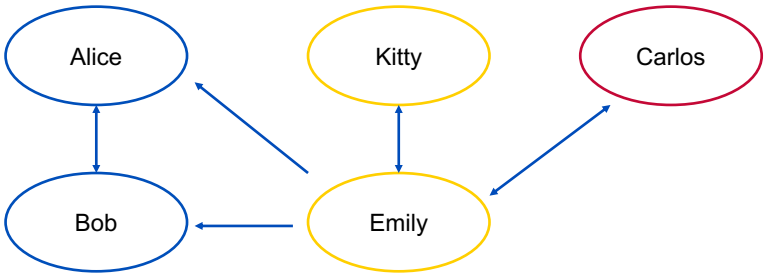
How PGP Works



Trust

- Centralized / hierarchal trust – where certain globally trusted bodies sign keys for every one else.
- Decentralized webs of trust – where you pick who you trust yourself, and decide if you trust who those people trust in turn.
- Which works better for what reasons?

Sample Web of Trust



You can share your “trust information” by publishing others’ public keys with your pgp sign

PGP by GnuPG

- **Create your keys**
  - Public key
  - Private key (secret key)
- **Identify key by**
  - Key ID (like 0x23AD8EF6)
- **Verify others’ public key by**
  - Key fingerprint
- **Find keys on PGP key servers**
  - Like http://pgp.mit.edu

Key Management

- Using graphical tools based on what you installed above:
  - GPG Keychain Access for OS X
  - Kleopatra or GPA for windows
- Using the command line:
  - gpg --list-keys

Key Management

- On printed media: published book or business cards:
- Digitally in email or using sneaker-net
- Online using the openpgp key servers.
- Still does not tell you if you trust the key.

Key Management

- Expiry dates ensure that if your private key is compromised they can only be used till they expire.
- Can be changed after creating the key.
- Before expiry, you need to create a new key, sign it with the old one, send the signed new one to everyone in your web of trust asking them to sign your new key.

Key Management - Revocation

- Used to mark a key as invalid before its expiry date.
- Always generate a revocation certificate as soon as you create your key.
- Do not keep your revocation certificate with your private key.
- gpg --gen-revoke IDENTITY

Key Management – Revocation (Windows)

Step 1 `C:\Users\User>gpg --gen-revoke C3ED244E`

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v2Comment: A revocation certificate should
followiQEnBCABCAARBQJYx150Ch0DZmluaXNoZWQACgkQYHMJecPtJE5ELAf+Pdh0DiHoE9D6bKDu49i
UJh5h6G9Yyw6jWZmHgVIoMY+Ae29kqbJ2IrELE+gcTwe6Ri6UrusrX9hz4xuRlUWNINE46b0pzkr5Zy5eMjo
mcj4gB41XAlJ7kgX2yQTYjOpA0IkRgenIHIlJUbXQovcF9w7QJzkhOi+1k7DZczmmPzvGDlntbX0rgj2hD+W4
71wHMWiVHdkPDP0H3gL3uDwikmqxqcyHKWZIX/SaTSZZETv0xO0R+Lr/li1yfJA6ihoxieOjy2SpB8xPKnHs3
JL8Gyj94jDuFRongD3yId3xvA2uaEWlvCyp+GJKuOWYAgEaQTmysihSwdSM4fIhHK8DFA===teYJ
-----END PGP PUBLIC KEY BLOCK-----
```

Step 2 `C:\Users\User>gpg --import my_revocation.txt`

Step 3

```
C:\Users\User>gpg --keyserver pgp.mit.edu --send-keys C3ED244E
gpg: sending key C3ED244E to hkp server pgp.mit.edu
```


Key Management - Partying

- Key signing parties are ways to build webs of trust.
- Each participant carries identification, as well as a copy of their key fingerprint. (maybe some $ as well ☺)
- Each participant decides if they're going to sign another key based on their personal policy.
- Keys are easiest kept in a keyring on an openpgp keyserver in the aftermath of the party.

Installing GnuPG Software

- Core software either commercial from pgp or opensource from gnupg.
  - https://www.gpg4win.org/ for windows
  - https://www.gpgtools.org/ for OS X
- Your package manager for Linux/UNIX
  - Source code from https://www.gnupg.org/

How PGP Works

- Check your GnuPG version

```
fakrul@rnd:~$
fakrul@rnd:~$ gpg --version
gpg (GnuPG) 1.4.12
Copyright (C) 2012 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Home: ~/.gnupg
Supported algorithms:
Pubkey: RSA, RSA-E, RSA-S, ELG-E, DSA
Cipher: 3DES, CAST5, BLOWFISH, AES, AES192, AES256, TWOFISH, CAMELLIA128,
        CAMELLIA192, CAMELLIA256
Hash: MD5, SHA1, RIPEMD160, SHA256, SHA384, SHA512, SHA224
Compression: Uncompressed, ZIP, ZLIB, BZIP2
fakrul@rnd:~$
```

How PGP Works

- Use “gpg --help” or “man gpg” for manuals.

```
Commands:

 -s, --sign [file]           make a signature
     --clearsign [file]      make a clear text signature
 -b, --detach-sign           make a detached signature
 -e, --encrypt               encrypt data
 -c, --symmetric             encryption only with symmetric cipher
 -d, --decrypt               decrypt data (default)
     --verify                verify a signature
     --list-keys             list keys
     --list-sigs             list keys and signatures
     --check-sigs            list and check key signatures
     --fingerprint           list keys and fingerprints
 -K, --list-secret-keys      list secret keys
     --gen-key               generate a new key pair
     --delete-keys           remove keys from the public keyring
     --delete-secret-keys    remove keys from the secret keyring
     --sign-key              sign a key
     --lsign-key             sign a key locally
     --edit-key              sign or edit a key
     --gen-revoke            generate a revocation certificate
     --export                export keys
     --send-keys             export keys to a key server
     --recv-keys             import keys from a key server
     --search-keys           search for keys on a key server
     --refresh-keys          update all keys from a keyserver
     --import                import/merge keys
     --card-status           print the card status
     --card-edit             change data on a card
     --change-pin            change a card's PIN
     --update-trustdb        update the trust database
```


Create Public & Private key pairs for GnuPG.

- Create Public & Private key pairs for GnuPG.

```
fakrul@rnd:~$ gpg --gen-key
gpg (GnuPG) 1.4.12; Copyright (C) 2012 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

gpg: directory `/home/fakrul/.gnupg' created
gpg: new configuration file `/home/fakrul/.gnupg/gpg.conf' created
gpg: WARNING: options in `/home/fakrul/.gnupg/gpg.conf' are not yet active during this run
gpg: keyring `/home/fakrul/.gnupg/secring.gpg' created
gpg: keyring `/home/fakrul/.gnupg/pubring.gpg' created
Please select what kind of key you want:
   (1) RSA and RSA (default)
   (2) DSA and Elgamal
   (3) DSA (sign only)
   (4) RSA (sign only)
Your selection? 
```

Find the above screen and choose “algorithm” of the encryption. At this time, we’ll choose “RSA and RSA” as a default.

Create public & private key pair

- Some people say that 1024 bit not strong enough anymore. So we'll choose 2048 bit for this time. After that we'll have to think about the expire date of the key pairs.

```
Please select what kind of key you want:
   (1) RSA and RSA (default)
   (2) DSA and Elgamal
   (3) DSA (sign only)
   (4) RSA (sign only)
Your selection? 1
RSA keys may be between 1024 and 4096 bits long.
What keysize do you want? (2048) 2048
Requested keysize is 2048 bits
Please specify how long the key should be valid.
         0 = key does not expire
      <n>  = key expires in n days
      <n>w = key expires in n weeks
      <n>m = key expires in n months
      <n>y = key expires in n years
Key is valid for? (0)
```

Note: It is important to select expire period. It is basically up to your security policy to decide this one. Several organization operate with 1 year. If you choose one year for this, you have to notify to users about the changing of the keys.

Create public & private key pair

- Type your "Real name" and "e-mail address" for this.

```
Key is valid for? (0) 1y
Key expires at Wed 30 Apr 2014 05:45:23 PM BDT
Is this correct? (y/N) y

You need a user ID to identify your key; the software constructs the user ID
from the Real Name, Comment and Email Address in this form:
    "Heinrich Heine (Der Dichter) <heinrichh@duesseldorf.de>"

Real name: Fakrul Alam
Email address: fakrul@dhakacom.com
Comment: Fakrul Alam / PGP Key
You selected this USER-ID:
    "Fakrul Alam (Fakrul Alam / PGP Key) <fakrul@dhakacom.com>"

Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit?
```

Note: Please keep in mind that anyone can make your keys of e-mail address. So what is the way that you can make sure that your key belongs your key ? The answer is "fingerprint".

Create public & private key pair

- Enter passphrase for 1st time & repeat it.

```
Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? O
You need a Passphrase to protect your secret key.

Repeat passphrase:
```

Note: Please do not forget this password and make sure the password is strong enough for brute forcing.

Create public & private key pair

- GnuPG automatically creates the keys

```
Not enough random bytes available.  Please do some other work to give
the OS a chance to collect more entropy! (Need 284 more bytes)
.+++++
.....+++++
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.

Not enough random bytes available.  Please do some other work to give
the OS a chance to collect more entropy! (Need 92 more bytes)
.+++++

Not enough random bytes available.  Please do some other work to give
the OS a chance to collect more entropy! (Need 111 more bytes)
..+++++
gpg: /home/fakrul/.gnupg/trustdb.gpg: trustdb created
gpg: key B2CF94E5 marked as ultimately trusted
public and secret key created and signed.

gpg: checking the trustdb
gpg: 3 marginal(s) needed, 1 complete(s) needed, PGP trust model
gpg: depth: 0  valid:   1  signed:   0  trust: 0-, 0q, 0n, 0m, 0f, 1u
gpg: next trustdb check due at 2014-04-30
pub   2048R/B2CF94E5 2013-04-30 [expires: 2014-04-30]
      Key fingerprint = 0302 768A C6F3 8EB3 3ED2  C511 FE72 5A7A B2CF 94E5
uid                  Fakrul Alam (Fakrul Alam / PGP Key) <fakrul@dhakacom.com>
sub   2048R/33D42A92 2013-04-30 [expires: 2014-04-30]
```

Note 1: When generating the key pairs, the operating system needs many random numbers. It is recommended to do something on the system for that.

Note 2: Read these messages carefully and should know the contents below
- Key ID
- What is the "trust"
- Key Length
- Expires date
- Key fingerprint

Create public & private key pair

- List your keys

```
fakrul@rnd:~$ gpg --list-keys B2CF94E5
pub   2048R/B2CF94E5 2013-04-30 [expires: 2014-04-30]
uid                  Fakrul Alam (Fakrul Alam / PGP Key) <fakrul@dhakacom.com>
sub   2048R/33D42A92 2013-04-30 [expires: 2014-04-30]

fakrul@rnd:~$ gpg --list-keys fakrul@dhakacom.com
pub   2048R/B2CF94E5 2013-04-30 [expires: 2014-04-30]
uid                  Fakrul Alam (Fakrul Alam / PGP Key) <fakrul@dhakacom.com>
sub   2048R/33D42A92 2013-04-30 [expires: 2014-04-30]
```

Note: Please remember the option "gpg --list-keys" you can list keys in your keyrings. And you can use both Key ID and e-mail address.

Create public & private key pair

- Where is the key files

```
fakrul@rnd:~$ cd .gnupg/
fakrul@rnd:~/.gnupg$ ls -lah
total 40K
drwx------  2 fakrul fakrul 4.0K Apr 30 18:00 .
drwxr-xr-x 34 fakrul fakrul 4.0K Apr 30 17:43 ..
-rw-------  1 fakrul fakrul 9.0K Apr 30 17:43 gpg.conf
-rw-------  1 fakrul fakrul 1.2K Apr 30 17:57 pubring.gpg
-rw-------  1 fakrul fakrul 1.2K Apr 30 17:57 pubring.gpg~
-rw-------  1 fakrul fakrul  600 Apr 30 17:57 random_seed
-rw-------  1 fakrul fakrul 2.6K Apr 30 17:57 secring.gpg
-rw-------  1 fakrul fakrul 1.3K Apr 30 17:57 trustdb.gpg
fakrul@rnd:~/.gnupg$
```

Just under the ".gnupg" directory of your home directory.
Public keys stored in : pubring.gpg. Private keys are stored in : secring.gpg
You can choose your favorite option in : gpg.conf

Sign messages & verify it

- Create file for encryption

```
fakrul@rnd:~/.gnupg$
fakrul@rnd:~/.gnupg$ echo "This is a test message." > test_sign
fakrul@rnd:~/.gnupg$ echo "Hope we can sign it." >> test_sign
fakrul@rnd:~/.gnupg$ cat test_sign
This is a test message.
Hope we can sign it.
fakrul@rnd:~/.gnupg$
```

- Sign the file

```
fakrul@rnd:~/.gnupg$ gpg --clearsign test_sign

You need a passphrase to unlock the secret key for
user: "Fakrul Alam (Fakrul Alam / PGP Key) <fakrul@dhakacom.com>"
2048-bit RSA key, ID B2CF94E5, created 2013-04-30

Enter passphrase:
```


Sign messages & verify it

- After typing your passphrase correctly, please try the "ls –l" and find the file "test_sign.asc". That is a signed file. Let's see the inside of file.

```
fakrul@rnd:~/.gnupg$ ls
gpg.conf  pubring.gpg  pubring.gpg~  random_seed  secring.gpg  test_sign  test_sign.asc  trustdb.gpg
fakrul@rnd:~/.gnupg$ cat test_sign.asc
-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA1

This is a test message.
Hope we can sign it.
-----BEGIN PGP SIGNATURE-----
Version: GnuPG v1.4.12 (GNU/Linux)

iQEcBAEBAgAGBQJRf7QcAAoJEP5yWnqyz5TlrY4H/i4eft0bBu310tUvG+4cAvGl
OVj7vLkB/Ty8jkaCFIpzP1lYrhaqcjTVSwCwXQ77SEa5hrRN7Wa/sfDbLsXBLJpK
OHqzDSgTErUbT2tjhFmrVtvmfqzuE52RqZkF4YjjSJX+cysdgY/WydnVWakLFBhs
4wqcXU5lV2pPJ08HGpSwalaF21VbnyLrseYdTXAwuqn6OIybh+7gSDOVCeT9/YPu
jbZniQEhBA7fdi18juTcwP61GZ2A/gLAyPaBKrHgsyABqN/7YnFbKnAXVPUiTZc5
gF/nkqFPR5wQ9kuPCsK2Uy24WrVU+gyDdBzFtBQPFDKZR2pCXG7HIbcxcuhXG7I=
=lV5Q
-----END PGP SIGNATURE-----
```

Sign messages & verify it

- Verification Process

```
fakrul@rnd:~/.gnupg$ gpg --verify test_sign.asc
gpg: Signature made Tue 30 Apr 2013 06:07:56 PM BDT using RSA key ID B2CF94E5
gpg: Good signature from "Fakrul Alam (Fakrul Alam / PGP Key) <fakrul@dhakacom.com>"
fakrul@rnd:~/.gnupg$
```

Note: Please find the message "Good signature from" and that is a message that gpg command can successfully verify the message. That means the file is surely signed by your private keys.

```
fakrul@rnd:~/.gnupg$
fakrul@rnd:~/.gnupg$ gpg --verify test_sign.asc
gpg: Signature made Tue 30 Apr 2013 06:07:56 PM BDT using RSA key ID B2CF94E5
gpg: BAD signature from "Fakrul Alam (Fakrul Alam / PGP Key) <fakrul@dhakacom.com>"
fakrul@rnd:~/.gnupg$
```

Note: You may find the message "BAD signature from" that means the file may be altered by someone. Do you want to see the inside ?

Export / Import Public Key

- Export your public key

```
fakrul@rnd:~/.gnupg$ gpg -a --export fakrul@dhakacom.com
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v1.4.12 (GNU/Linux)

mQENBFF/sO8BCAC827VLM+lPbztyPPWKTSIlOMpq45glakdBAccZZQe9GX4YuUPi
epP3VxKAMgTKk2lAu6kRiA9YmNGhXJ4waB44YzGhyjigfurtL3RH8Olu+CJRraGj
s+76KajST4gythJCuSUwO3+GJNLajVHUxPmSu6/v3LzVcxuQnhcqyD85xPgacjI
jgFVu76j6DEjrhxjd2U1f5dNoBhltfasDo5Mr6loyekXNForRljI3X7for26aKuN
UueUICRH60CvOn4xVaLU71R76aTxZigaCQUgUCoBwdyfgavBhQh2GgxD6mj9pGIs
/nROw6PCbaBllrxLOTHTdDlDVuvJWTsirGGhABEBAAG0OUZha3JlbCBbGFtIChG
YWtydWwgQWxhbSAvIFBHUCBLZXkpIDxmYWtydWxAZGhha2Fjb20uY29tPokBPgQT
AQIAKAUCUX+w7wIbAwUJAeEzgAYLCQgHAwIGFQgCCQoLBBYCAwECHgECF4AACgkQ
/nJaerLPlOW9qgf/X9vT9vzfX59zd4isY0xoGEzsaXvNtLila+Gu7kMUXNAUGxwz
gLlKJLoKN5Y9/uxCRjm+EdiEkPTJwxiKq/gVqR5fRzHkhmI4vZ4GfAHmkHlJ6BxL
6GfCOjwiLcNdfNP0eONSg9io6QlRAAVu4PSYBxHg7i5Ohx1FzR+/Ecl7J/Pr408
cbqjeHC3LTQWZgWlBPtQbrvtfPYUcQtcqlBal69FlywKz7Cf3jhnTpSxZJMQL+aK
LcTDRHNfQl20rpJb81rNtvolfdANB5JjgPOGDC+szxzOSyPwCttr5sanJy3DHVaT
lwqT7d9IQCi8fVcMdD0cjbp4Om/JO/AUYlgchLkBDQRRf7DvAQgAs44En2oAcLAM
IuQuhp83D7uX4Zsp4EJGnDtCX3o+2QU4staVmVeGAZIBeOd+iIvRlsLmKmvSdDCa
4ZTU3Yrk+lVcJjUbBldFIdf3n0X1JbrJh/e0a+gWglSJVJ3OlTK0olZ6Y/tOkxkh
OU/HIbUtNrTrqPrqmXwiD08JnwaG1N3FppUQ/9mLCmHJ5vGjF0wg8otota2acHH8
y9OgaopPsi3AEYdfgdHyHON0gxzK4BQ324wbiDppUoFR/0/OsqgYKURaWmiPr6Y0
T1WwJ3djCHb84JuoVPvsy4XtvOJSUtfuazZ13JKR2qQo4qKkaZxYG6++3DGAujk
Nm2sa8+0QwARAQABiQElBBgBAgAPBQJRf7DvAhsMBQkB4TOAAAoJEP5yWnqyz5T1
Q38H/0OlOX7liyXONoFyn/fSUjXNHxhnEMtlA7MKHla4EdgsbpW92/hTPWEfI7qz
VRGZFbJUjXgWOFLrd5I0BeDxTpsdhbKELGX3S1MimlogEilM4i/zlzCpKvMTGlrf
lvkUYs+oD0WxT2jsvavzE+fYM229r4bxk2ulxVgoSb6/7PwBWsUXVRrvjHRPqHG
H/d6fWJGpIRzofOSeKm1WPvrOz/rN9/1JkEpJAmws1pTiC6C1qXAVGZX3E5oHTCL
ay8DwOWRUD+tgHwTFR6F+FpHC/4cBoI65npaXnRUeHlhggJ2NKLybDdmL9L2N/tX
2ADI8ibfMZYWKnLvgRIJmAHBpic=
=tZiU
-----END PGP PUBLIC KEY BLOCK-----
```

Note: You can export key to a file using:
`gpg -a --export fakrul@dhakacom.com > fakrul_public.key`

Export / Import Public Key

- Import Key

```
fakrul@rnd:~/.gnupg$ gpg --import fakrul_bdhub.key
gpg: key 109C56FC: public key "Fakrul Alam (bdHUB pgp key) <fakrul@bdhub.com>" imported
gpg: Total number processed: 1
gpg:               imported: 1  (RSA: 1)
fakrul@rnd:~/.gnupg$
```

- Find the imported key

```
fakrul@rnd:~/.gnupg$ gpg --list-key 109C56FC
pub   2048R/109C56FC 2013-02-05 [expires: 2020-02-05]
uid                  Fakrul Alam (bdHUB pgp key) <fakrul@bdhub.com>
uid                  [jpeg image of size 10334]
sub   2048R/F66ACECA 2013-02-05 [expires: 2020-02-05]
```

APNIC

Export / Import Public Key

- Make sure fingerprint is right

```
fakrul@rnd:~/.gnupg$ gpg --fingerprint 109C56FC
pub   2048R/109C56FC 2013-02-05 [expires: 2020-02-05]
      Key fingerprint = 94EA 86AD 428C 4072 7995  9150 E338 712B 109C 56FC
uid                  Fakrul Alam (bdHUB pgp key) <fakrul@bdhub.com>
uid                  [jpeg image of size 10334]
sub   2048R/F66ACECA 2013-02-05 [expires: 2020-02-05]
```

APNIC

Encrypt Message

- Make some file to encrypt

```
fakrul@rnd:~/.gnupg$ echo "This is a file for encryption" > test_encrypt
fakrul@rnd:~/.gnupg$ echo "Can you read me" >> test_encrypt
fakrul@rnd:~/.gnupg$ cat test_encrypt
This is a file for encryption
Can you read me
fakrul@rnd:~/.gnupg$
```

- Encrypt the file `# gpg --encrypt --armor -r RECEIVER_EMAIL_ID -u SENDER_EMAIL_ID test_encrypt`

```
fakrul@rnd:~/.gnupg$ gpg --encrypt --armor -r fakrul@bdhub.com -u fakrul@dhakacom.com test_encrypt
gpg: F66ACECA: There is no assurance this key belongs to the named user

pub  2048R/F66ACECA 2013-02-05 Fakrul Alam (bdHUB pgp key) <fakrul@bdhub.com>
 Primary key fingerprint: 94EA 86AD 428C 4072 7995  9150 E338 712B 109C 56FC
      Subkey fingerprint: E2FB 4B8C E12A C043 A578  DB66 CAA0 09C8 F66A CECA

It is NOT certain that the key belongs to the person named
in the user ID.  If you *really* know what you are doing,
you may answer the next question with yes.

Use this key anyway? (y/N) y
```


Encrypt Message

- Try to read encrypted message

```
fakrul@rnd:~/.gnupg$ cat test_encrypt.asc
-----BEGIN PGP MESSAGE-----
Version: GnuPG v1.4.12 (GNU/Linux)

hQEMA8qgCcj2as7KAQf/bMc79wwCaEl1UbdW13Dz6YEOUDaaMG9dBbNY8iK+ijfA
usow8AZJBH/L94HY83t+OzmWbMMhyXwCn3DN6VtqhfAtuNk1QFiqiTQ+njHg33cW
TT3pcwsDmqVhJ1D+WuKqezY59HSWy1kNJLS7t4Tyw3ROLyj1Eyg2Og3Bv4VE2sBM
Pr3nHub6TweVHdmp7kQeW6LrLe93pjnXWtShVfvuvRuhfoV3XPfUqIX+XH679ZdU
1vZYaX1hg1rVJoV6rOgWA/IYPUon/e/n4CcEETu2TqPoTwvbs96qmSwB8FeF0dHC
QeaEHddd1zO4IO112xnGfJ3BmXuJ4s3s/dHmNDepL9JuAboumgGemMckLA1b1RIX
ClITHIX+wLA8zWj9u0Z8t9sGOS1uPNnj1IZWUH3Cc1ptT+jtEd15oPMrfx+I0bac
6gzFEbOa2OaG/hq2sUXPz+CD0FSR4xREaxAy1cNctnuCKZSyOLMGnVBMy6O9qNY=
=fB9B
-----END PGP MESSAGE-----
```

Decrypt Message

- Decrypt the file.

```
# gpg --output OUTPUT_FILE_NAME --
decrypt ENCRYPTED_FILE_NAME
```

```
FakrulMac:Downloads rapappu$ gpg --output test1.txt --decrypt test.txt.asc

You need a passphrase to unlock the secret key for
user: "Fakrul Alam (bdHUB pgp key) <fakrul@bdhub.com>"
2048-bit RSA key, ID F66ACECA, created 2013-02-05 (main key ID 109C56FC)

gpg: encrypted with 2048-bit RSA key, ID F66ACECA, created 2013-02-05
      "Fakrul Alam (bdHUB pgp key) <fakrul@bdhub.com>"
```

- Read the file

```
FakrulMac:Downloads rapappu$ cat test1.txt
This is an encrypted message.
Let see if you can decrypt it.
FakrulMac:Downloads rapappu$
```

APNIC

LAB

APNIC

Overview

- Network Security Fundamentals
- Threat Pragmatics
- Cryptography Basics
- SSH
- Network Infrastructure Filtering at the border
- PGP
- TLS/SSL
- **IPSec**
- IDS & Snort
- Wireshark

APNIC

Overview

- Virtual Private Networks
- What is IPsec?
- Benefits of IPsec
- IPsec Architecture and Components
- Setting up an IPsec VPN tunnel
- Tunnel and Transport Mode

APNIC

Virtual Private Network

- Creates a secure tunnel over a public network
  - Client to firewall
  - Router to router
  - Firewall to firewall
- Uses the Internet as the public backbone to access a secure private network
  - Remote employees can access their office network
- VPN Protocols
  - PPTP (Point-to-Point tunneling Protocol)
  - L2F (Layer 2 Forwarding Protocol)
  - L2TP (Layer 2 Tunneling Protocol)
  - IPSec (Internet Protocol Security)

IPsec

- Provides Layer 3 security (RFC 2401)
  - Transparent to applications (no need for integrated IPSec support)
- A set of protocols and algorithms used to secure IP data at the network layer
- Combines different components:
  - Security associations (SA)
  - Authentication headers (AH)
  - Encapsulating security payload (ESP)
  - Internet Key Exchange (IKE)
- A security context for the VPN tunnel is established via the ISAKMP

IPSec Standards

- RFC 4301 “The IP Security Architecture”
  - Defines the original IPsec architecture and elements common to both AH and ESP
- RFC 4302
  - Defines authentication headers (AH)
- RFC 4303
  - Defines the Encapsulating Security Payload (ESP)
- RFC 2408
  - ISAKMP
- RFC 5996
  - IKE v2 (Sept 2010)
- RFC 4835
  - Cryptographic algorithm implementation for ESP and AH

Benefits of IPsec

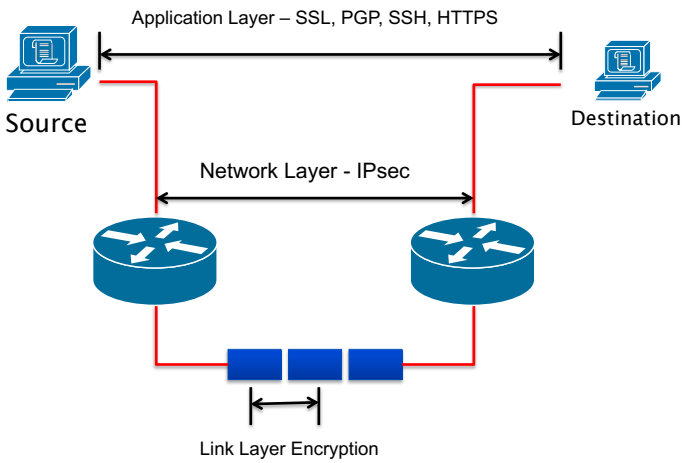
- Confidentiality
  - By encrypting data
- Integrity
  - Routers at each end of a tunnel calculates the checksum or hash value of the data
- Authentication
  - Signatures and certificates
  - All these while still maintaining the ability to route through existing IP networks

“IPsec is designed to provide interoperable, high quality, cryptographically-based security for IPv4 and IPv6” - (RFC 2401)

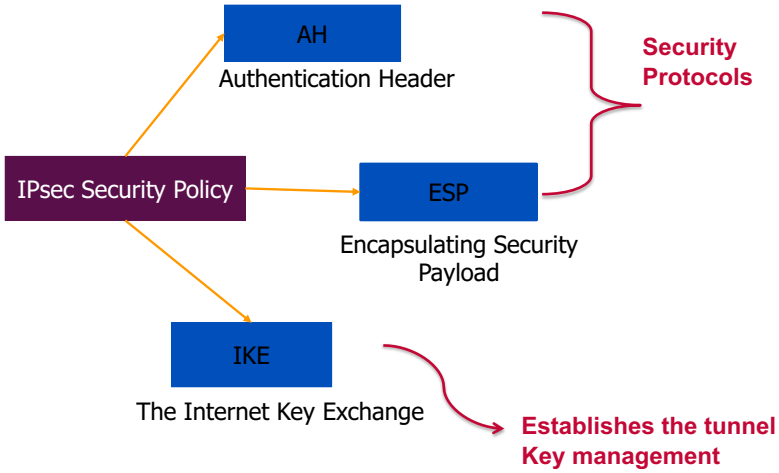
Benefits of IPsec

- Data integrity and source authentication
  - Data “signed” by sender and “signature” is verified by the recipient
  - Modification of data can be detected by signature “verification”
  - Because “signature” is based on a shared secret, it gives source authentication
- Anti-replay protection
  - Optional; the sender must provide it but the recipient may ignore
- Key management
  - IKE – session negotiation and establishment
  - Sessions are rekeyed or deleted automatically
  - Secret keys are securely established and authenticated
  - Remote peer is authenticated through varying options

Different Layers of Encryption



IPsec Architecture



Security Associations (SA)

- A collection of parameters required to establish a secure session
- Uniquely identified by three parameters consisting of
  - Security Parameter Index (SPI)
  - IP destination address
  - Security protocol (AH or ESP) identifier
- An SA is either uni- or bidirectional
  - IKE SAs are bidirectional
  - IPsec SAs are unidirectional
    - Two SAs required for a bidirectional communication
- A single SA can be used for AH or ESP, but not both
  - must create two (or more) SAs for each direction if using both AH and ESP

Security Parameter Index (SPI)

- A unique 32-bit identification number that is part of the Security Association (SA)
- It enables the receiving system to select the SA under which a received packet will be processed.
- Has only local significance, defined by the creator of the SA.
- Carried in the ESP or AH header
- When an ESP/AH packet is received, the SPI is used to look up all of the crypto parameters

How to Set Up SA

- Manually
  - Sometimes referred to as “manual keying”
  - You configure on each node:
    - Participating nodes (I.e. traffic selectors)
    - AH and/or ESP [tunnel or transport]
    - Cryptographic algorithm and key
- Automatically
  - Using IKE (Internet Key Exchange)

ISAKMP

- Internet Security Association and Key Management Protocol
- Used for establishing Security Associations (SA) and cryptographic keys
- Only provides the framework for authentication and key exchange, but key exchange independent
- Key exchange protocols
  - Internet Key Exchange (IKE)
  - Kerberized Internet Negotiation of Keys (KINK)

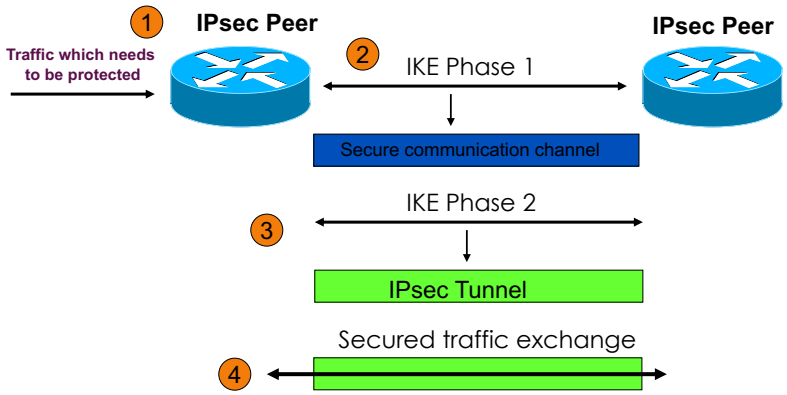
Authentication Header (AH)

- Provides source authentication and data integrity
  - Protection against source spoofing and replay attacks
- Authentication is applied to the entire packet, with the mutable fields in the IP header zeroed out
- If both AH and ESP are applied to a packet, AH follows ESP
- Operates on top of IP using protocol 51
- In IPv4, AH protects the payload and all header fields except mutable fields and IP options (such as IPSec option)

Encapsulating Security Payload (ESP)

- Uses IP protocol 50
- Provides all that is offered by AH, plus data confidentiality
  - uses symmetric key encryption
- Must encrypt and/or authenticate in each packet
  - Encryption occurs before authentication
- Authentication is applied to data in the IPsec header as well as the data contained as payload

Overview of IPSec



Internet Key Exchange (IKE)

- "An IPsec component used for performing mutual authentication and establishing and maintaining Security Associations." (RFC 5996)
- Typically used for establishing IPsec sessions
- A key exchange mechanism
- Five variations of an IKE negotiation:
  - Two modes (aggressive and main modes)
  - Three authentication methods (pre-shared, public key encryption, and public key signature)
- Uses UDP port 500

IKE Modes

Mode	Description
Main mode	Three exchanges of information between IPsec peers. Initiator sends one or more proposals to the other peer (responder) Responder selects a proposal
Aggressive Mode	Achieves same result as main mode using only 3 packets First packet sent by initiator containing all info to establish SA Second packet by responder with all security parameters selected Third packet finalizes authentication of the ISAKMP session
Quick Mode	Negotiates the parameters for the IPsec session. Entire negotiation occurs within the protection of ISAKMP session

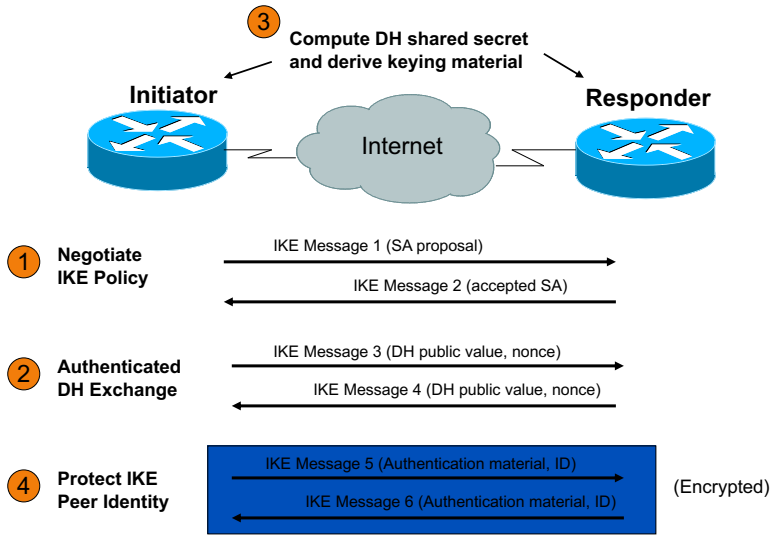
Internet Key Exchange (IKE)

- Phase I
  - Establish a secure channel (ISAKMP SA)
  - Using either main mode or aggressive mode
  - Authenticate computer identity using certificates or pre-shared secret
- Phase II
  - Establishes a secure channel between computers intended for the transmission of data (IPsec SA)
  - Using quick mode

IKE Phase 1 (Main Mode)

- Main mode negotiates an ISAKMP SA which will be used to create IPsec SAs
- Three steps
  - SA negotiation (encryption algorithm, hash algorithm, authentication method, which DH group to use)
  - Diffie-Hellman exchange
  - Provide authentication information
  - Authenticate the peer

IKE Phase 1 (Main Mode)



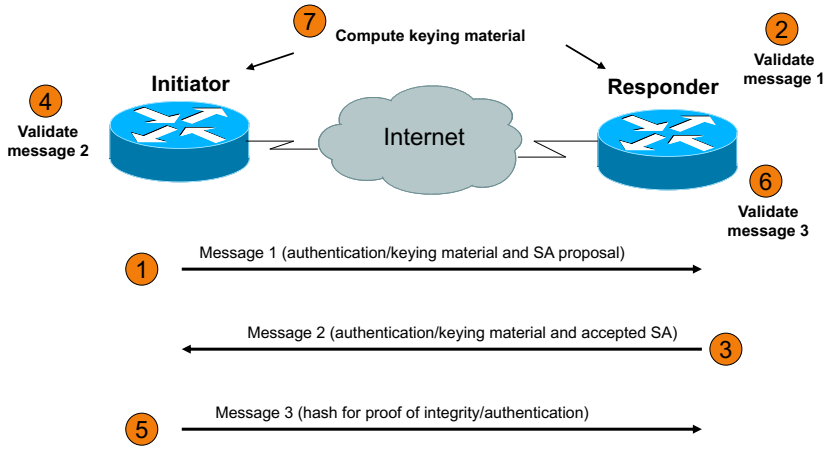
IKE Phase 1 (Aggressive Mode)

- Uses 3 (vs 6) messages to establish IKE SA
- No denial of service protection
- Does not have identity protection
- Optional exchange and not widely implemented

IKE Phase 2 (Quick Mode)

- In phase 2, all traffic is encrypted using the ISAKMP Security Association
- Each quick mode negotiation results in two IPsec Security Associations (one inbound, one outbound)
- Creates/refreshes keys

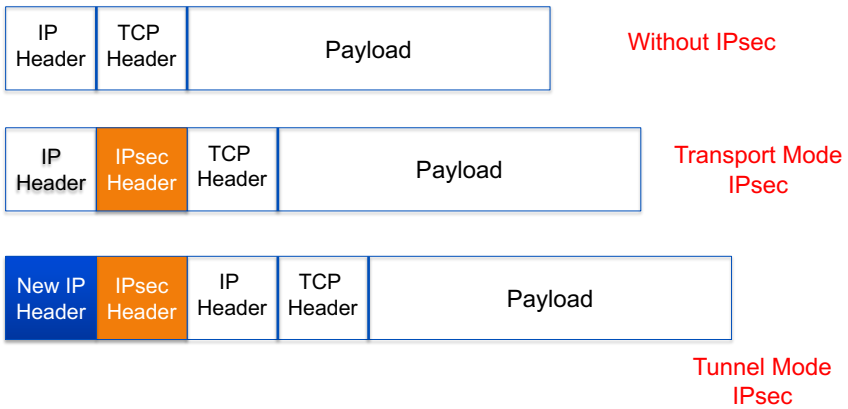
IKE Phase 2 (Quick Mode)



IPsec Modes

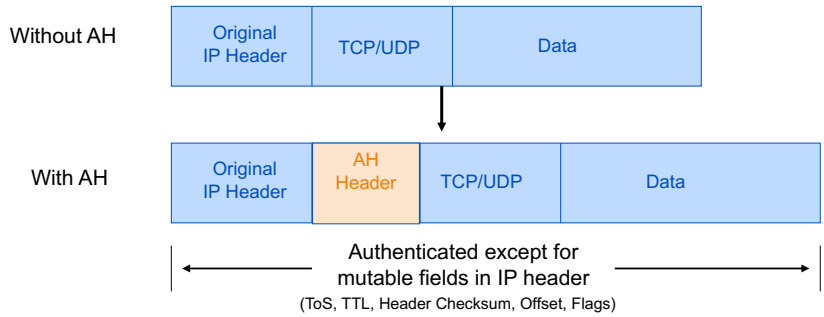
- Tunnel Mode
  - Entire IP packet is encrypted and becomes the data component of a new (and larger) IP packet.
  - Frequently used in an IPsec site-to-site VPN
- Transport Mode
  - IPsec header is inserted into the IP packet
  - No new packet is created
  - Works well in networks where increasing a packet's size could cause an issue
  - Frequently used for remote-access VPNs

Tunnel vs. Transport Mode IPsec



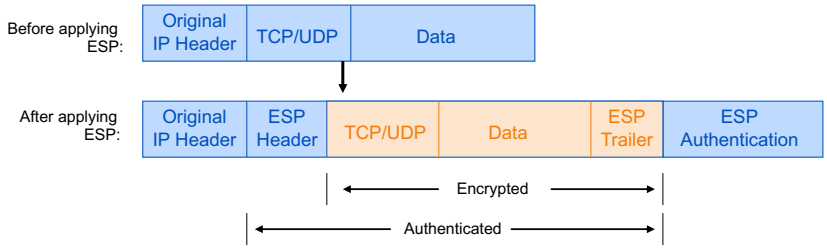
Packet Format Alteration for AH Transport Mode

Authentication Header

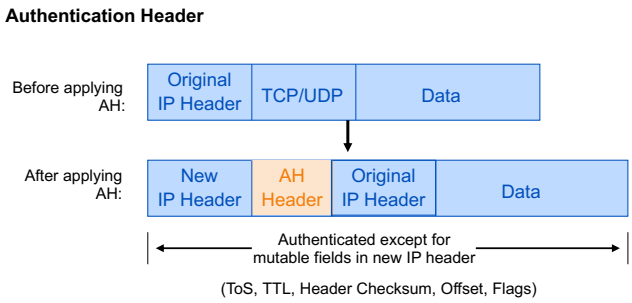


Packet Format Alteration for ESP Transport Mode

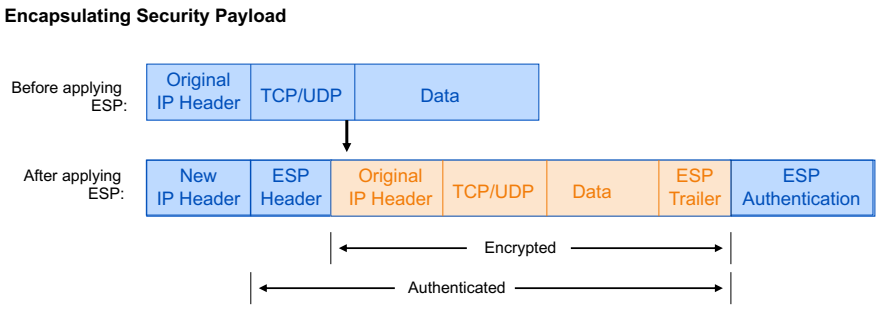
Encapsulating Security Payload



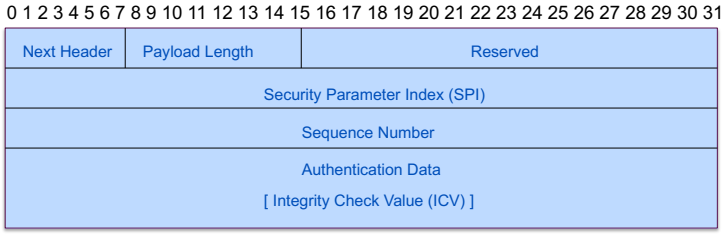
Packet Format Alteration for AH Tunnel Mode



Packet Format Alteration for ESP Tunnel Mode

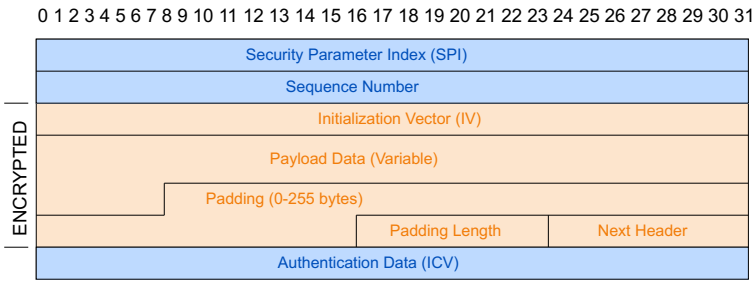


AH Header Format



- Next Header (8 bits): indicates which upper layer protocol is protected (UDP, TCP, ESP)
- Payload Length (8 bits): size of AH in 32-bit longwords, minus 2
- Reserved (16 bits): for future use; must be set to all zeroes for now
- SPI (32 bits): arbitrary 32-bit number that specifies to the receiving device which security association is being used (security protocols, algorithms, keys, times, addresses, etc)
- Sequence Number (32 bits): start at 1 and must never repeat. It is always set but receiver may choose to ignore this field
- Authentication Data: ICV is a digital signature over the packet and it varies in length depending on the algorithm used (SHA-1, MD5)

ESP Header Format



- SPI: arbitrary 32-bit number that specifies SA to the receiving device
- Seq #: start at 1 and must never repeat; receiver may choose to ignore
- IV: used to initialize CBC mode of an encryption algorithm
- Payload Data: encrypted IP header, TCP or UDP header and data
- Padding: used for encryption algorithms which operate in CBC mode
- Padding Length: number of bytes added to the data stream (may be 0)
- Next Header: the type of protocol from the original header which appears in the encrypted part of the packet
- Authentication Header: ICV is a digital signature over the packet and it varies in length depending on the algorithm used (SHA-1, MD5)

Considerations For Using IPsec

- Security Services
  - Data origin authentication
  - Data integrity
  - Replay protection
  - Confidentiality
- Size of network
- How trusted are end hosts – can apriori communication policies be created?
- Vendor support
- What other mechanisms can accomplish similar attack risk mitigation

IPsec Best Practices

- Use IPsec to provide integrity in addition to encryption.
  - Use ESP option
- Use strong encryption algorithms
  - 3DES and AES instead of DES
- Use a good hashing algorithm
  - SHA instead of MD5
- Reduce the lifetime of the Security Association (SA) by enabling Perfect Forward Secrecy (PFS)
  - Increases processor burden so do this only if data is highly sensitive

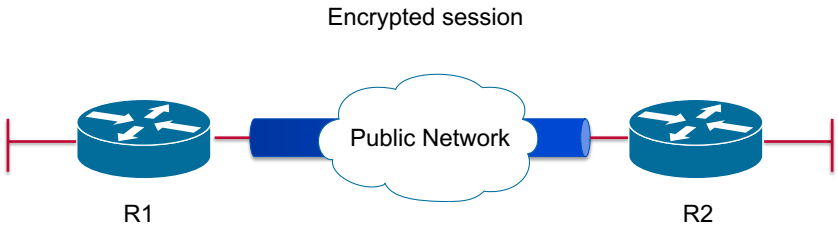
Configuring IPsec

- Step 1: Configure the IKE Phase 1 Policy (ISAKMP Policy)
  - ***crypto isakmp policy [priority]***
- Step 2: Set the ISAKMP Identity
  - **crypto isakmp identity {ipaddress|hostname}**
- Step 3: Configure the IPsec transfer set
  - **crypto ipsec transform-set transform-set-name <transform1> <transform2> mode [tunnel|transport]**
  - **crypto ipsec security-association lifetime seconds seconds**

Configuring IPsec

- Step 5: Creating map with name
  - `Crypto map crypto-map-name seq-num ipsec-isakmp`
  - `Match address access-list-id`
  - `Set peer [ipaddress|hostname]`
  - `Set transform-set transform-set-name`
  - `Set security-association lifetime seconds seconds`
  - `Set pfs [group1|group2]`
- Step 6: Apply the IPsec Policy to an Interface
  - `Crypto map crypto-map-name local-address interface-id`

IPsec Layout



Router Configuration

```
crypto isakmp policy 1
  authentication pre-share
  encryption aes
  hash sha
  group 5
crypto isakmp key Training123 address 172.16.11.66
!
crypto ipsec transform-set ESP-AES-SHA esp-aes esp-sha-hmac
!
crypto map LAB-VPN 10 ipsec-isakmp
  match address 101
  set transform-set ESP-AES-SHA
  set peer 172.16.11.66
```

Phase 1 SA

Encryption and authentication

Phase 2 SA

Router Configuration

```
int fa 0/1
crypto map LAB-VPN
Exit
!
access-list 101 permit ip 172.16.16.0
0.0.0.255 172.16.20.0 0.0.0.255
```

Apply to an outbound interface

Define interesting VPN traffic

IPsec Debug Commands

- **sh crypto ipsec sa**
- **sh crypto isakmp peers**
- **sh crypto isakmp sa**
- **sh crypto map**

Pretty Good IPsec Policy

- IKE Phase 1 (aka ISAKMP SA or IKE SA or Main Mode)
  - 3DES (AES-192 if both ends support it)
  - Lifetime (8 hours = 480 min = 28800 sec)
  - SHA-2 (256 bit keys)
  - DH Group 14 (aka MODP# 14)
- IKE Phase 2 (aka IPsec SA or Quick Mode)
  - 3DES (AES-192 if both ends support it)
  - Lifetime (1 hour = 60 min = 3600 sec)
  - SHA-2 (256 bit keys)
  - PFS 2
  - DH Group 14 (aka MODP# 14)

Cryptography Application TLS / SSL

<TITLE>

<DATE>

<LOCATION>

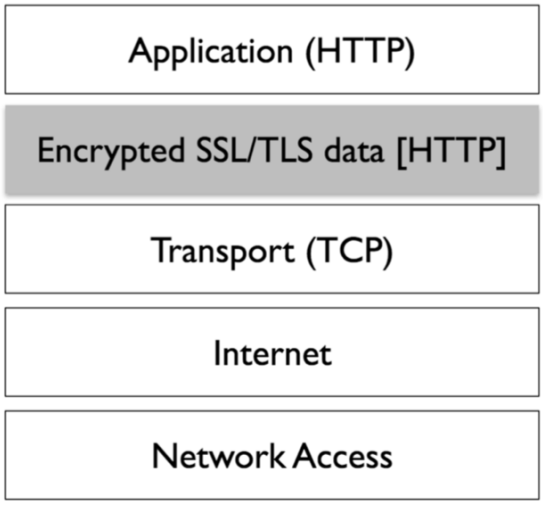
Issue Date: [31-12-2015]
Revision: [V.1]

APNIC

History

- Secure Sockets Layer was developed by Netscape in 1994 as a protocol which permitted persistent and secure transactions.
- In 1997 an Open Source version of Netscape's patented version was created, which is now OpenSSL.
- In 1999 the existing protocol was extended by a version now known as Transport Layer Security (TLS).
- By convention, the term "SSL" is used even when technically the TLS protocol is being used.

Location of SSL Protocol & TCP Ports



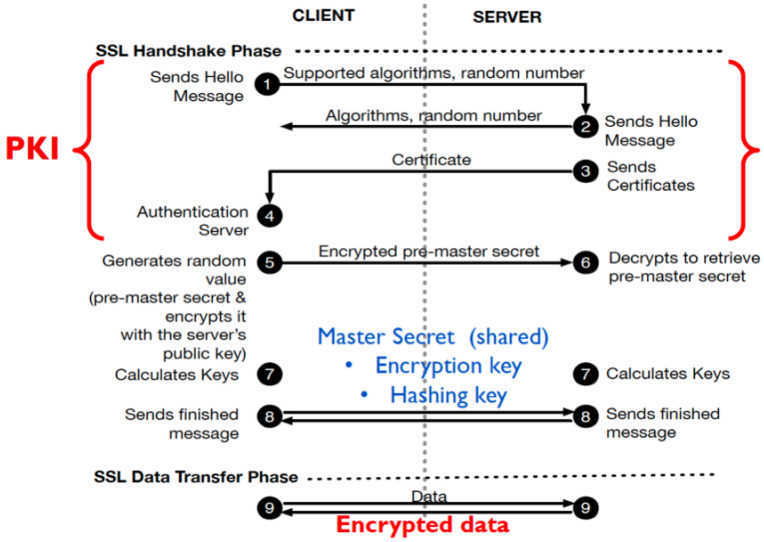
TLS/SSL: What it does

- Confidentiality
  - Encryption
- Integrity
  - Keyed hash (HMAC): TLS (authentication!)
  - Hash (MAC): SSL
- Authentication
  - Certificates

SSL/TLS Operations

- Client connects to the server
  - To access a resource
- Public Key cryptography is used during handshake to authenticate parties and exchange session key.
  - PKI (X.509 Certificates)
- Symmetric Key cryptography to encrypt and hash data.
  - Master secret (shared secret) generated
  - Separate **Encryption** and **Hashing** keys from the master secret

How SSL/TLS Works – Part 1



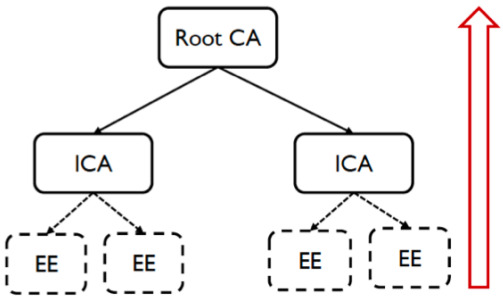
PKI – public key infra

- Digital (X.509) certificates
  - associates a public key with an individual or organization
  - public key of the subject!

Version
Serial Number
Signature Algorithm
Issuer Name
Validity Period
Subject Name
Subject Public Key
Issuer ID
Subject ID
Extensions (CRL)

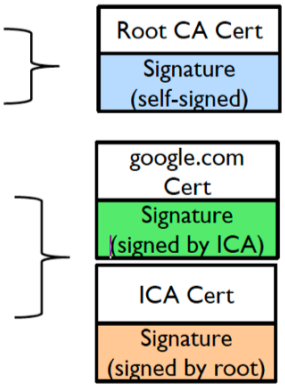
PKI – Chain of Trust

- Root CA
  - Self-signed
  - Issue and sign ICA's certificate
- Intermediate CA
  - Issue and sign EE certificate
- End Entity

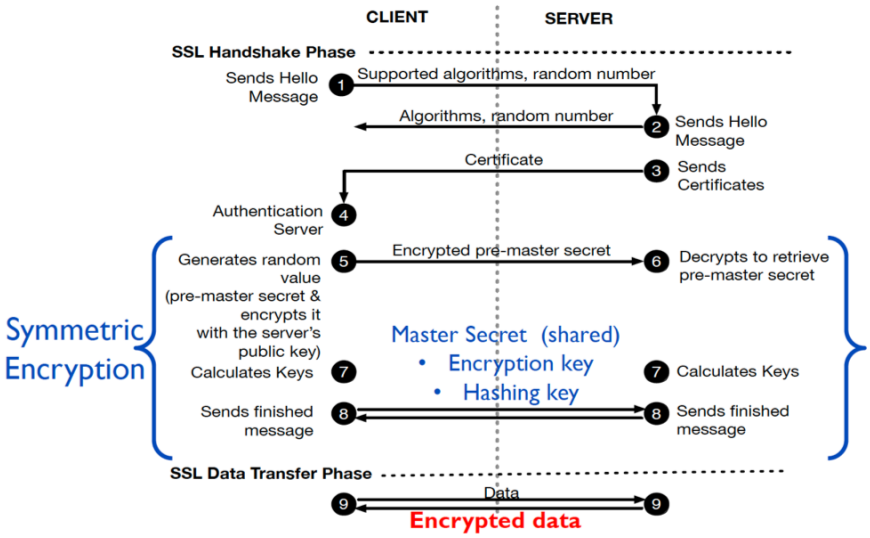


PKI - Example

- Client (browser) sends https request to google.com
  - browsers have trusted CA certificates stored
- Web server sends back google.com's certificate
  - Signed by Google ICA, plus
  - Google ICA's certificate signed by root CA(GeoTrust)
- Verify the certificates up the chain of trust
  - Once successfully verified, use the public key

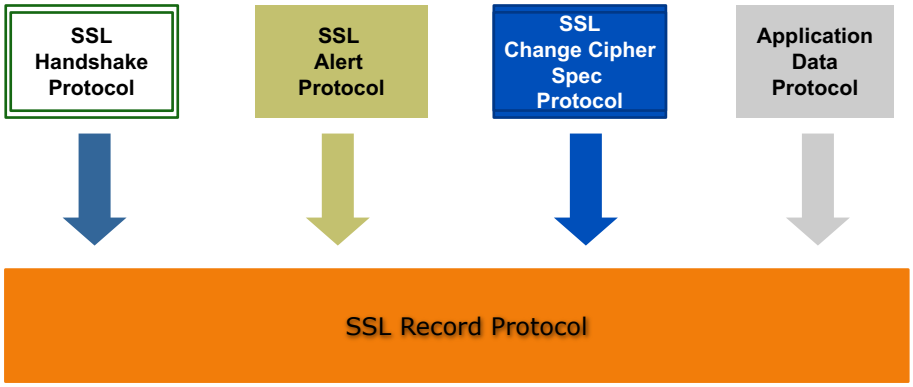


How SSL/TLS Works – Part 2

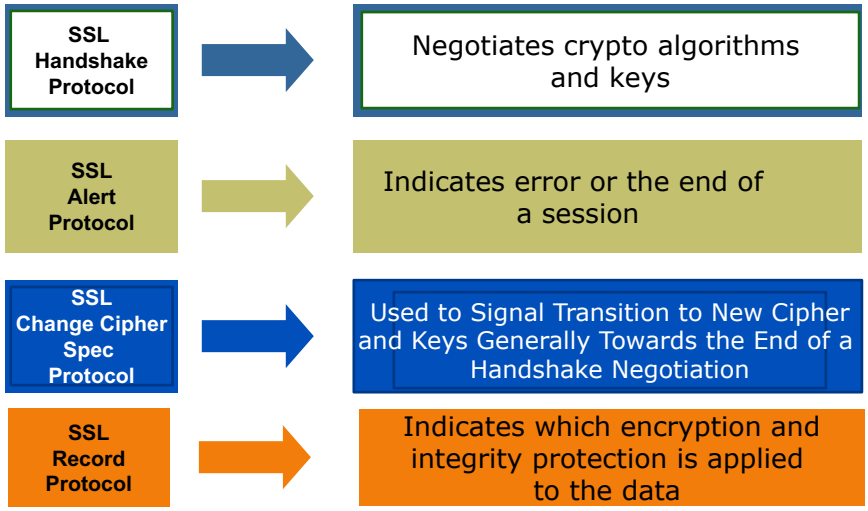


SSL Protocol Building Blocks

SSL is a Combination of a Primary Record Protocol with Four 'Client' Protocols



SSL Protocol Building Block Functions

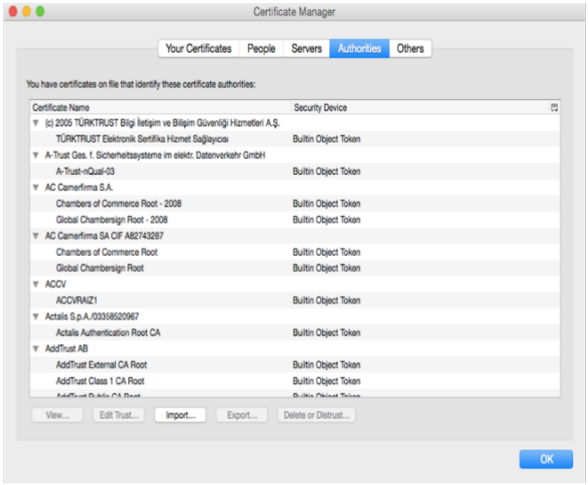


Symmetric Encryption

- Once the server's public key is verified up the chain of trust
  - The client generates a pre-master secret
    - C-random & S-random
  - Sends to the server encrypted with server's public key
- Both client and server generates the Master Secret
  - Uses the pre-master secret, C-random, and S-random with the agreed key exchange cipher (eg: DH)
- Separate Encryption and Hashing keys generated from the Master secret
  - All future communication hashed and encrypted using the symmetric keys

Trusted vs Non Trusted Certificate

Certificate Authority



Chinese CA WoSign faces revocation after issuing fake certificates of Github, Microsoft and Alibaba

MONDAY, AUGUST 29, 2016

Chinese CA WoSign faces revocation after issuing fake certificates of Github, Microsoft and Alibaba

One of the largest Chinese root certificate authority WoSign issued many fake certificates due to an vulnerability. WoSign's free certificate service allowed its users to get a certificate for the base domain if they were able to prove control of a subdomain. This means that if you can control a subdomain of a major website, say percy.github.io, you're able to obtain a certificate by WoSign for github.io, taking control over the entire domain.

In deed, this has been seen in the wild in multiple instances as reported in the thread, aggregated here. I've notified related parties about the possible fake certs.

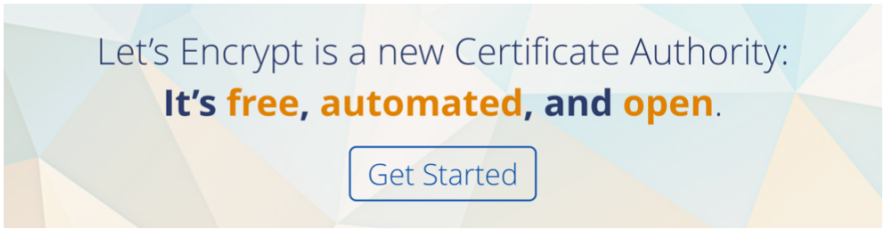
Possible fake cert for Github -- confirmed fake
https://crt.sh/?id=29647048
https://crt.sh/?id=29805567

Update: crt.sh is down after my post. Google's CT log here https://www.google.com/transparencyreport/https/ct/#domain=github.io&incl_exp=false&incl_sub=false&issuer=IPrsb9Gbn4s%3D

Possible fake cert for Alibaba, the largest commercial site in China -- confirmed fake
https://crt.sh/?id=29884704

https://groups.google.com/forum/m/#!topic/mozilla.dev.security.policy/k9PBmyLCi8I/discussion

Introducing Let'sEncrypt



https://letsencrypt.org/

292

Introducing Let'sEncrypt

- Which browsers and operating systems support Let's Encrypt
  - https://community.letsencrypt.org/t/which-browsers-and-operating-systems-support-lets-encrypt/4394
- Check your browser
  - https://wiki.apnictraining.net

LAB

Overview

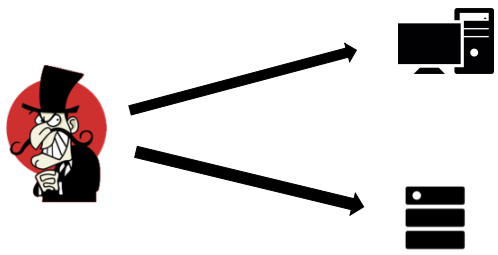
- Network Security Fundamentals
- Threat Pragmatics
- Cryptography Basics
- SSH
- Network Infrastructure Filtering at the border
- PGP
- TLS/SSL
- IPSec
- **IDS & Snort**
- Wireshark

Sometimes, Defenses Fail

- Our defenses aren't perfect
  - –Patches weren't applied promptly enough
  - –Antivirus signatures not up to date
  - –0-days get through
  - –Someone brings in an infected USB drive
  - –An insider misbehaves
- Now what?
- Most penetrations are never detected
  - –This allows continuing abuse, and helps the attackers spread elsewhere

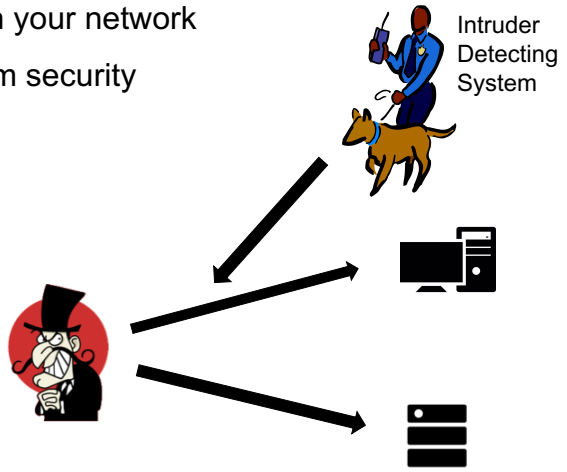
Unexpected Activity

- There could be an intruder even if you have security practice in place



Additional Monitoring

- Activity in your network
- To confirm security



APNIC

What can IDS realistically do

- Detect successful attacks
- Look for various things that shouldn't be there
- Infected files
- Attacks on other machines
- Packets that shouldn't exist
- Strange patterns of behavior
- Contain attacks before they spread further
- Clean up penetrated machines—because you'll know they're infected
- Recognition of pattern reflecting known attacks
- Statistical analysis for abnormal activities

APNIC 298

What IDS can't do

•Compensate for weak authentication & identification mechanisms

•Investigate attacks without human intervention

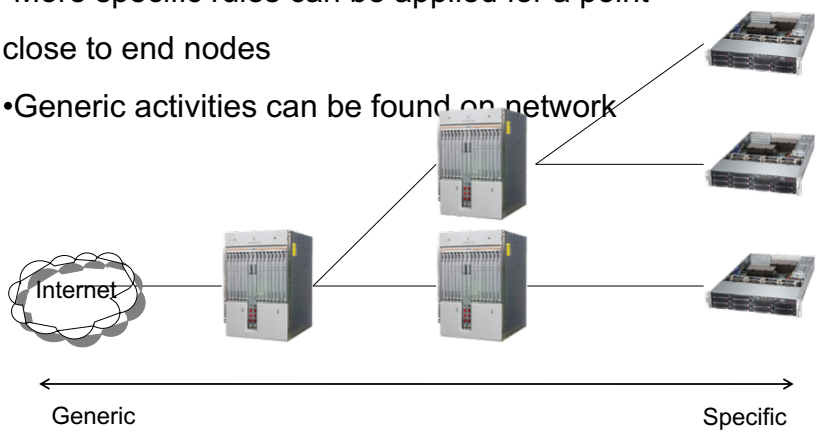
•Guess the content of your organization security policy

•Compensate for weakness in networking protocols, for example IP Spoofing

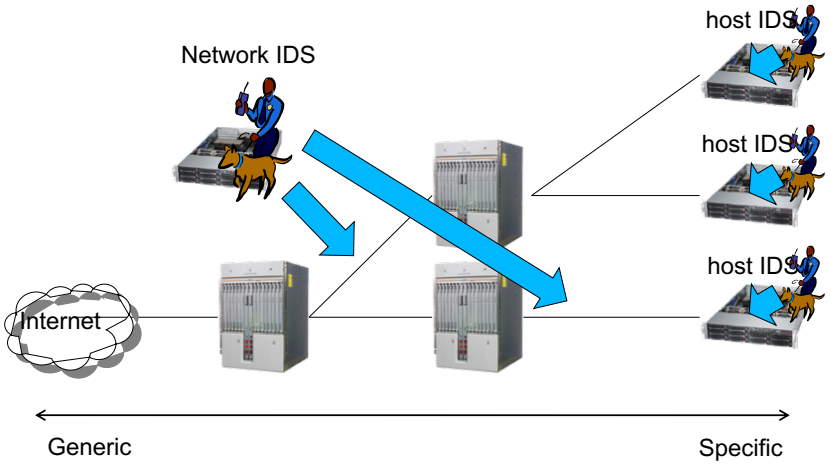
Monitoring Point

•More specific rules can be applied for a point close to end nodes

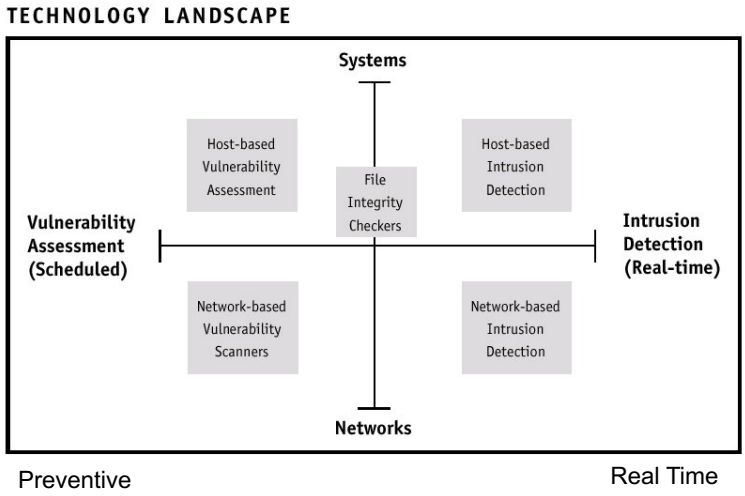
•Generic activities can be found on network



Network and Host IDS



IDS Technology landscape



Alert

- You may receive tons of millions of alerts
  - Depending on your detection rules
  - There are many suspicious activities in the Internet today
- You should notice a critical one at least
  - Detection rule is important!

Alert

- False Positive / Type I Error:
  - is the incorrect rejection of a true null hypothesis
  - is when a system raises an incorrect alert
- False Negative / Type II Error:
  - is the failure to reject a false null hypothesis
  - is when an attack pass undetected

Types of Detection

- Signature Based
  - –Match patterns against known attacks
  - –Catch the intrusions in terms of the characteristics of known attacks or system vulnerabilities
- Anomaly Based
  - –Look for unusual behavior
  - –Detect any action that significantly deviates from the normal behavior

Intrusion Detection for ISPs

- Monitor your own network—but that's no different than any other enterprise
- Monitor your customers
  - –Good: you can help them by detecting problems
  - –Good: you can prevent them from clogging your infrastructure
  - –Bad: it can be privacy-invasive

SNORT

•Snort is an open source IDS, and one of the oldest ones

•Hundreds of thousands of users

•Active development of rules by the community make Snort up to date, and often more so than commercial alternatives

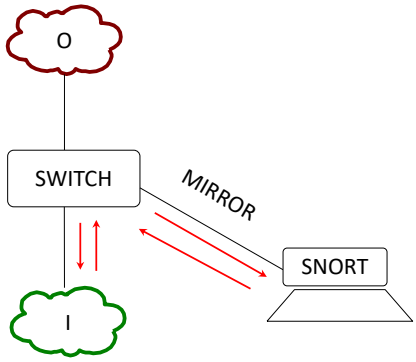
•Snort is fast! It can run at Gbit/s rates with the right hardware and proper tuning

Getting Snort to see the network

•You could run Snort in multiple ways

- –As a device “in line” behind or after the firewall/router
  - •But this adds one more element that can fail in your connectivity
- –Or you could use a span/mirror port to send traffic to Snort
- –Or you can use an “optical splitter” to “mirror” or “tap into” traffic from a fiber optic link
  - •This method and the previous are the most recommended

Getting Snort to see the network



Getting Snort to see the network

•Be careful not to overload your switch port – If you mirror a gigabit port to another gigabit port, the monitoring port (the receiving port) can drop packets if the total traffic exceeds 1 Gbit/s

Monitoring Port…

•On Cisco Catalyst, this is a “SPAN” port

•You can SPAN one port to another, a group of ports to one port, or an entire VLAN to a port

•Sample config:

```
interface FastEthernet 0/1
# port monitor FastEthernet 0/2
```

•This would copy any packet received on F0/2 to F0/1

Snort configuration file

•By default, /etc/snort/snort.conf

•It’s a long file – 900+ lines

•If you browse it, you will notice many “preprocessor” entries

•Snort has a number of “preprocessors” which will analyze the network traffic and possibly clean it up before passing it to the rules

SNORT Rules

- Snort rules are plain text files
- Adding new rules to snort is as simple as dropping the files into /etc/snort/rules/
- Groups of rules can be loaded from snort.conf using the “include” statement
- Rules can match anything
- Technical – web attacks, buffer overflow, portscan, etc…
- Policy/user oriented – URL filtering, keyword, forbidden applications, etc…

Tailoring the rules

•Not all rules will make sense in your network

•You will want to customize which rules you want to run

•Otherwise you will get many false positives, which will lead you to ignore Snort, or simply turn it off…

•It doesn’t help to have logs full of junk alerts you don’t want

•To avoid this, rules can be suppressed (disabled)

Updating Snort rules

•The commercially maintained snort rules are available for free with a 30 day delay from http://www.snort.org/start/rules

•Other rules are maintained by some volunteers at emerging threats: http://rules.emergingthreats.net/open/

•The updating of rules can be automated with a tool called “Pulled Pork”, which is located at http://code.google.com/p/pulledpork/

Snort rules

•Snort rules are divided into two logical sections:

–**Rule Header** : The rule header contains the rule's action, protocol, source and destination IP addresses and netmasks, and the source and destination ports information.

–**Rule Options** : The rule option section contains alert messages and information on which parts of the packet should be inspected to determine if the rule action should be taken.

Snort rules

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 22
(msg: "SSH Detected"; sid:10; rev:1;)
```

The text up to the first parenthesis is the rule header and the section enclosed in parenthesis contains the rule options. The words before the colons in the rule options section are called option *keywords*.

Snort rules header

•alert - generate an alert using the selected alert method, and then log the packet

•log - log the packet

•pass - ignore the packet

•activate - alert and then turn on another dynamic rule

•dynamic - remain idle until activated by an activate rule , then act as a log rule

•drop - block and log the packet

•reject - block the packet, log it, and then send a TCP reset if the protocol is TCP or an ICMP port unreachable message if the protocol is UDP.

•sdrop - block the packet but do not log it.

Snort rules : The Direction Operator

•The direction operator **->** indicates the orientation, or direction, of the traffic that the rule applies to.

•There is no **<-** operator.

•Bidirectional operator **<>**

Snort rules : sid

•The sid keyword is used to add a “Snort ID” to rules

- –Range 0-99 is reserved for future use
- –Range 100-1,000,000 is reserved for rules that come with Snort distribution
- –All numbers above 1,000,000 can be used for local rules

Snort rules : classtype

•Rules can be assigned classifications and priority numbers to group and distinguish them

–/etc/snort/classification.config

```
config classification: DoS,Denial of Service Attack,2
```

Name Description Priority

•You can distinguish between high- and low-risk alerts

Sample rules

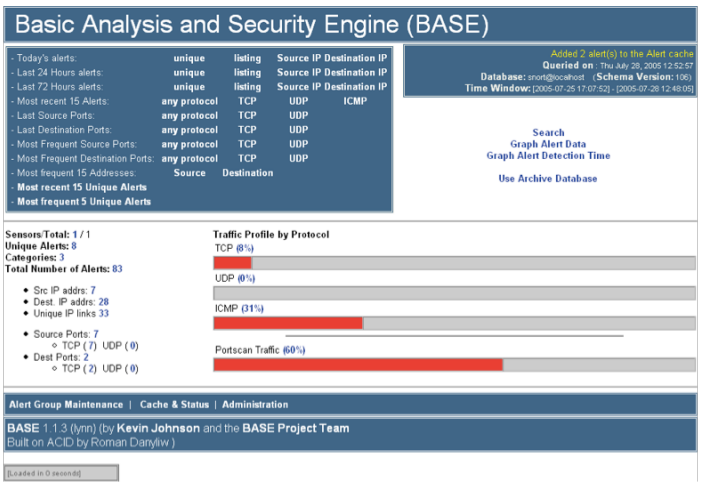
```
alert tcp msg:"MYSQL root login attempt";
flow:to_server,established; content:"|0A 00 00 01 85 04 00 00
80|root|00|"; classtype:protocol-command-decode; sid:1775;
rev:2;)

alert tcp $EXTERNAL_NET any -> $SQL_SERVERS 3306 (msg:"MYSQL
show databases attempt"; flow:to_server,established;
content:"|0F 00 00 00 03|show databases"; classtype:protocol-
command-decode; sid:1776; rev:2;)

alert tcp $EXTERNAL_NET any -> $SQL_SERVERS 3306 (msg:"MYSQL
4.0 root login attempt"; flow:to_server,established;
content:"|01|"; within:1; distance:3; content:"root|00|";
within:5; distance:5; nocase; classtype:protocol-command-
decode; sid:3456; rev:2;)
```

Reporting and logging

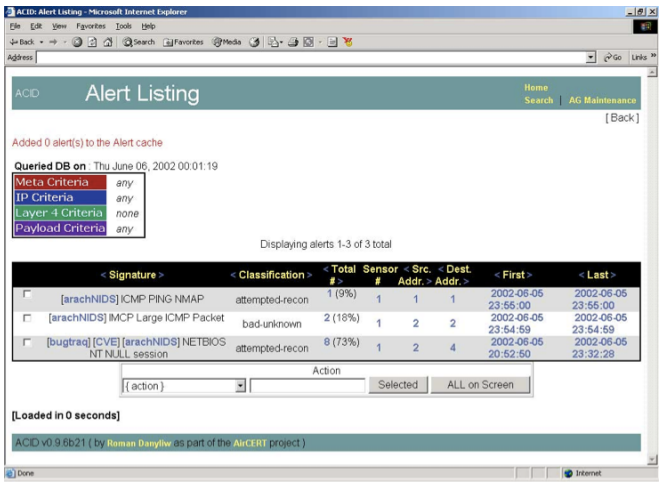
•Snort can be made to log alerts to an SQL database, for easier searching

•A web front-end for Snort, BASE, allows one to browse security alerts graphically

BASE (Basic Analysis and Security Engine)



BASE (Basic Analysis and Security Engine)



References and documentation

- Snort preprocessors:
  - –http://www.informit.com/articles/article.aspx?p=101148&seqNum=2
- Snort documentation
  - –http://www.snort.org/docs
- An install guide for Ubuntu 10.04:
  - –http://www.snort.org/assets/158/014-snortinstallguide292.pdf
- Writing SNORT Rules
  - –http://manual.snort.org/node27.html

Lab

APNIC

SNORT Setup

•Follow lab manual to install SNORT and check the basic SNORT rules.

APNIC

Exercise : 1

- Write a rules to check XMAS scan on your server
  - Clue XMAS scan sets the FIN, PSH, and URG flags
  - Check the rules with nmap
    - `nmap -sX SERVER_IP`

Exercise : 2

- Write a rules to check any external network access your webserver /admin pages
  - Match content

Exercise : 3

•Write a rules to check SSH brute force attack and log IP trying to connect more than 3 times in 60 seconds.

–threshold:type threshold, track by_src, count 3, seconds 60;

APNIC

Overview

- Network Security Fundamentals
- Threat Pragmatics
- Cryptography Basics
- SSH
- Network Infrastructure Filtering at the border
- PGP
- TLS/SSL
- IPSec
- IDS & Snort
- **Wireshark**

APNIC

Why we need to capture packet & how it's related to security?

tcpdump Definition

tcpdump is a utility used to capture and analyze packets on network interfaces. Details about these packets can either be displayed to the screen or they can be saved to a file for later analysis. tcpdump utilizes the libpcap library for packet capturing.

tcpdump command example

```
# tcpdump –nni eth0
# tcpdump –nni eth0 host 10.10.10.10
# tcpdump –nni eth0 dst host 10.10.10.10 and proto tcp
# tcpdump –nni eth0 src net 10.10.10.0/24 and port tcp
and portrange 1-1024
```

-nn = don't use DNS to resolve IPs and display port no
-i = interface to watch
dst = watch only traffic destined to a net, host or port
src = watch only traffic whose src is a net, host or port
net = specifies network
host = specifies host
port = specifies a port
proto = protocol ie tcp or udp

tcpdump command example

```
# tcpdump –nni eth0 –s0
# tcpdump –nni eth0 not port 22 –s0 –c 1000
# tcpdump –nni eth0 not port 22 and dst host 10.10.10.10
and not src net 10.20.30.0/24
```

-s0 = setting samples length to 0 means use the required length to catch whole packet
-c = no to packets

tcpdump pcaps

```
# tcpdump –nni eth0 -w capture.pcap –vv –c 1000
# tcpdump –nni eth0 –r capture.pcap and port 80
```

-w capture.pcap = save capture packet to capture.pcap
–vv = display number of packet captured
-r caputre.pcap = read capture file
-c = no to packets

tcpdump Output

```
IP 199.59.148.139.443 > 192.168.1.8.54343: Flags [P.],
seq 53:106, ack 1, win 67, options [nop,nop,TS val
854797891 ecr 376933204], length 53

IP 192.168.1.8.54343 > 199.59.148.139.443: Flags [.], ack
106, win 4092, options [nop,nop,TS val 376934736 ecr
854797891], length 0

IP 199.59.148.139.443 > 192.168.1.8.54343: Flags [P.],
seq 106:159, ack 1, win 67, options [nop,nop,TS val
854797891 ecr 376933204], length 53

IP 192.168.1.8.54343 > 199.59.148.139.443: Flags [.], ack
159, win 4091, options [nop,nop,TS val 376934736 ecr
854797891], length 0
```

What is Wireshark?

- Wireshark is a network packet/protocol analyzer.
  - A network packet analyzer will try to capture network packets and tries to display that packet data as detailed as possible.
- Wireshark is perhaps one of the best open source packet analyzers available today for **UNIX** and **Windows**.

About Wireshark

- Formerly known as “Ethereal”
  - Author, Gerald Combs quit Network Integration Services
  - Free
- Requirement
  - Need to install winpcap
  - Latest wireshark installer contains winpcap, don’t worry
  - (On Windows Vista) Need Administrator Privilege to capture
- GUI
  - Dramatically improved

Why Wireshark

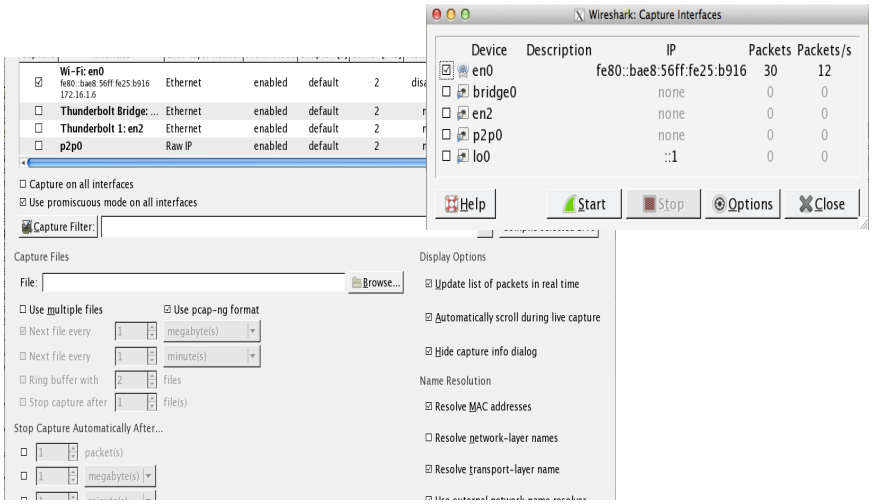
- network administrators use it to **troubleshoot network problems**
- network security engineers use it to **examine security problems**
- developers use it to **debug protocol implementations**
- people use it to **learn network protocol** internals
- Wireshark isn't an intrusion detection system.
- Wireshark will not manipulate things on the network, it will only "measure" things from it.

How to Install

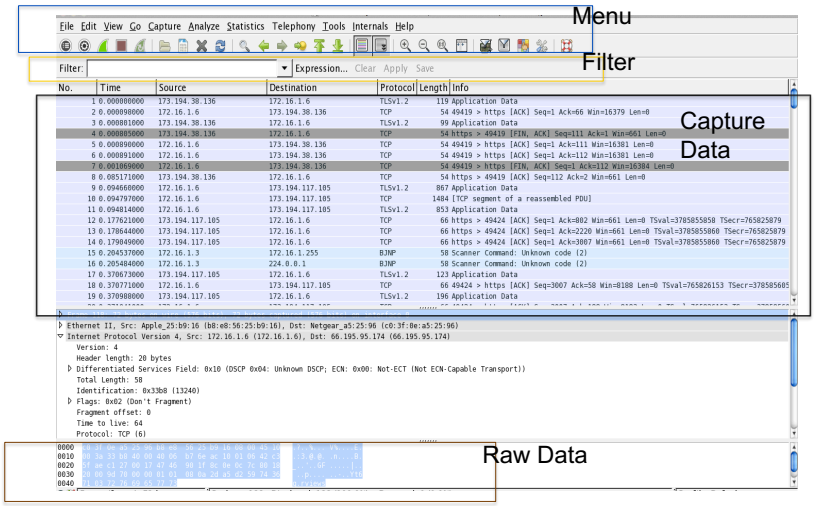
- Very straight forward
- Just double-click and follow the instructions.

- https://www.wireshark.org/download.html

Capture



Dashboard



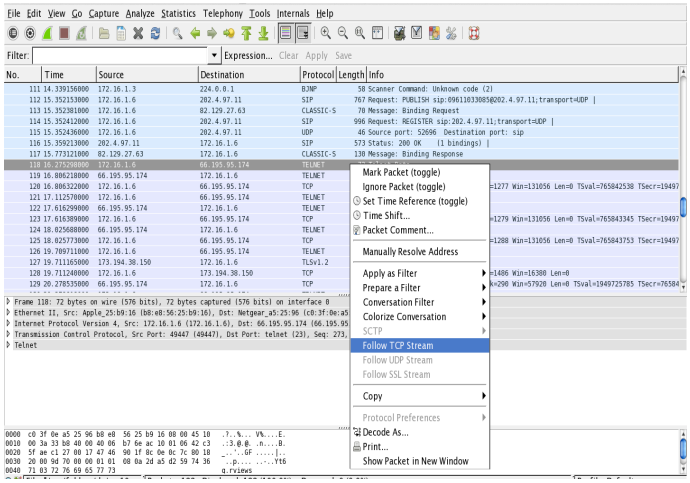
Filters

- Capture filter
  - Capture Traffic that match capture filter rule
  - save disk space
  - prevent packet loss
- Display filter
- Tweak appearance

Apply Filters

- ip.addr == 10.0.0.1 [Sets a filter for any packet with 10.0.0.1, as either the source or dest]
- ip.addr==10.0.0.1 && ip.addr==10.0.0.2 [sets a conversation filter between the two defined IP addresses]
- http or dns [sets a filter to display all http and dns]
- tcp.port==4000 [sets a filter for any TCP packet with 4000 as a source or dest port]
- tcp.flags.reset==1 [displays all TCP resets]
- http.request [displays all HTTP GET requests]
- tcp contains rviews [displays all TCP packets that contain the word 'rviews'. Excellent when searching on a specific string or user ID]
- !(arp or icmp or dns) [masks out arp, icmp, dns, or whatever other protocols may be background noise. Allowing you to focus on the traffic of interest]

Follow TCP Stream

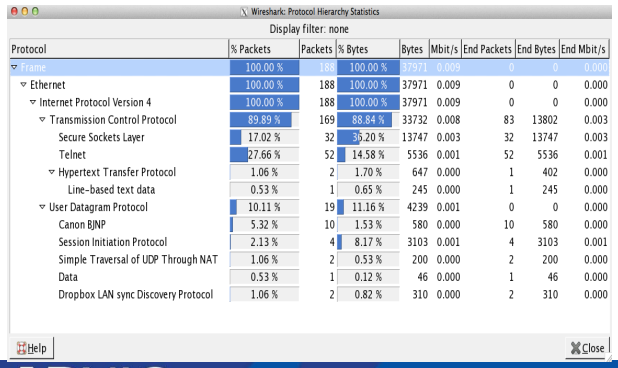


Follow TCP Stream

- Build TCP Stream
  - Select TCP Packet -> Follow TCP Stream

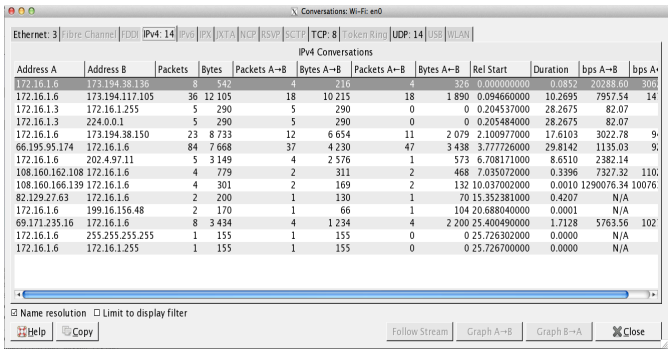
Use “Statistics”

- What protocol is used in your network
  - Statistics -> Protocol Hierarchy



Use “Statistics”

- Which host most chatty
  - Statistics -> Conversations



Need CLI?

- If you stick to character based interface, try tshark.exe
- C:\program files\wireshark\tshark.exe

Tcpdump & Wireshark

- tcpdump -i <interface> -s 65535 -w <some-file>

Exercise

- Install Wireshark into your PC
- Run wireshark and Capture inbound/outbound traffic
- Download capture files from
  - Follow the instructor's guide.

Exercise 1: Good Old Telnet

- File
  - telnet.pcap
- Question
  - Reconstruct the telnet session.
- Q1: Who logged into 192.168.0.1
  - Username _____, Password _____ .
- Q2: After logged in what did the user do?
  - Tip
  - telnet traffic is not secure

Exercise 2: Massive TCP SYN

- File
  - massivesyn1.pcap and massivesyn2.pcap
- Question
  - Point the difference with them.
- Q1: massivesyn1.pcap is a _____ attempt.
- Q2: massivesyn2.pcap is a _____ attempt.
- Tip
  - Pay attention to Src IP

Exercise 3: Chatty Employees

- File
  - chat.dmp
- Question
- Q1: What kind protocol is used? _____
- Q2: This is conversation between _____@hotmail.com and _____@hotmail.com
- Q3: What do they say about you(sysadmin)?
- Tip
  - Your chat can be monitored by network admin.

Exercise 4: Suspicious FTP activity

- File
  - ftp1.pcap
- Question
  - Q1: 10.121.70.151 is FTP _____ .
  - Q2: 10.234.125.254 is FTP _____ .
  - Q3: FTP Err Code 530 means _____ .
  - Q4: 10.234.125.254 attempt _____.
- Tip
  - How many login error occur within a minute?

Exercise 5: Unidentified Traffic

- File
  - Foobar.pcap
- Question
  - Q1: see what's going on with wireshark gui
    - Statistics -> Conversation List -> TCP (*)
  - Q2: Which application use TCP/6346? Check the web.

Exercise 6: Covert channel

- File
  - covertinfo.pcap
- Question
  - Take a closer look! This is not a typical ICMP Echo/Reply…
  - Q1: What kind of tool do they use? Check the web.
  - Q2: Name other application which tunneling user traffic.

Exercise 7: SIP

- File
  - sip_chat.pcap
- Questions:
  - Q1: Can we listen to SIP voice?
  - Q2: How!!

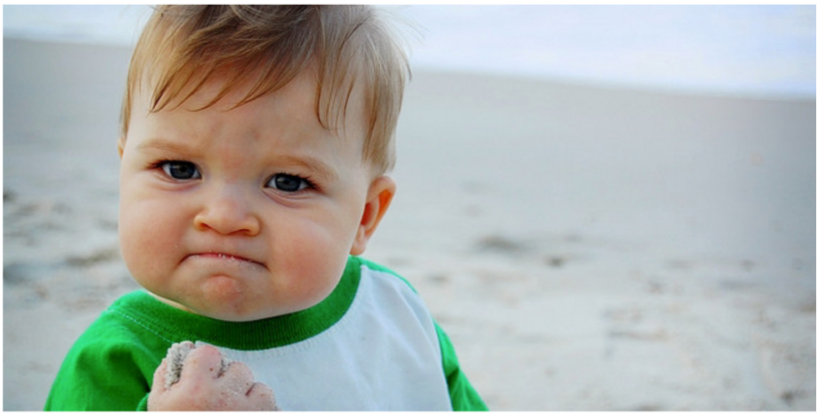
Virustotal

- https://www.virustotal.com/
- Checking virus

LAB

Defense and Mitigation – Community

- You can't do it all alone!



Defense and Mitigation – Community

- … and luckily, there is a great community providing services/tools, such as
  - Security @ APNIC https://www.apnic.net/security
  - Passive DNS by cert.at
  - Panopticon Shared Proxy by circl.lu et al.
  - openresolverproject.com / www.openresolver.nl
  - n6 Reports by cert.pl
  - CAP Reports by Team Cymru
  - phishtank.com, spamcop.net
  - Contacts contacts contacts
  - …and many more – what else do you know / offer?


Thank You!
END OF SESSION
APNIC
367