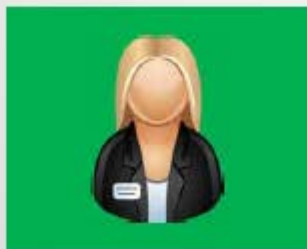


# Evading IDS, Firewalls, and Honeypots

Module 16

Unmask the **Invisible Hacker**.



# Survey: The State of Network Security 2014



**57%** of organizations either struggle to identify vulnerabilities or understand IT risk in business content

**97%** of organizations agree that business stakeholders should be made aware of vulnerabilities in their applications and "own the risk"



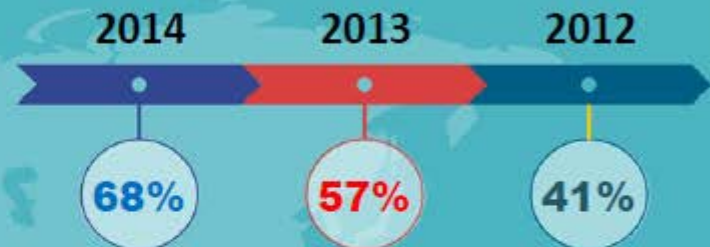
2013



2014

of organizations said time-consuming manual processes, lack of visibility into security policies and poor change management were the greatest challenge of managing network security devices

## Adoption of Next-Generation Firewalls



## Who can you trust?

**57%**

of organizations suffered a data center application outage in the last year due to misconfigured security infrastructure

**22%**

of organizations suffered 3 or more data center application outages in the last year



**AND**

2014



82%

2013



76%

of organizations suffered an application or network outage as a result of an out-of-process security change

Only **33%** of organizations are confident of their third party provider's capabilities to ensure the highest level of protection



**11%** of organizations have no confidence in their provider's ability to ensure the highest level of protection



<http://blog.algosec.com>

# Cybersecurity Market Report



"Next generation" cybersecurity spending could reach **\$15 billion to \$20 billion** in the next 3 years

FBR Capital Markets predicts **20% increase** in "next-generation cybersecurity spending" this year (2015), as companies move beyond traditional firewall and endpoint vendors

About **10% of enterprises and government agencies** have upgraded to next-generation security software, such as firewalls that detect and block threats at the application level

High profile data breaches have piqued the demand for **WAF** (web application firewall) systems. The worldwide market is expected to reach **\$777.3 million in 2018**

<http://cybersecurityventures.com>



# Module Objectives

- Understanding IDS, Firewall, and Honeypot Concepts
- IDS, Firewall and Honeypot Solutions
- Understanding different techniques to bypass IDS
- Understanding different techniques to bypass Firewalls



- IDS/Firewall Evading Tools
- Understanding different techniques to detect Honeypots
- IDS/Firewall Evasion Countermeasures
- Overview of IDS and Firewall Penetration Testing





# Module Flow

01

**IDS, Firewall  
and Honeypot  
Concepts**

02

**IDS, Firewall  
and Honeypot  
Solutions**

03

**Evading IDS**

04

**Evading  
Firewalls**



05

**IDS/Firewall  
Evading Tools**

06

**Detecting  
Honeypots**

07

**IDS/Firewall  
Evasion Counter-  
measures**

08

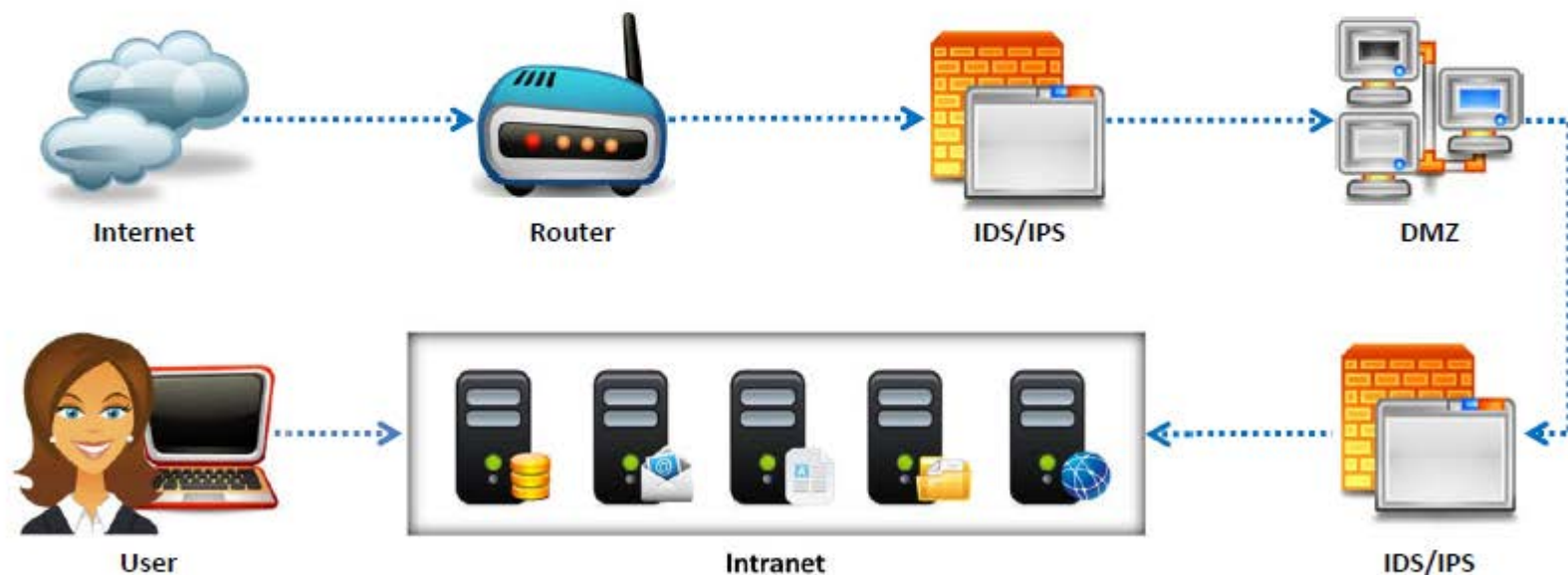
**Penetration  
Testing**

# Intrusion Detection Systems (IDS) and their Placement

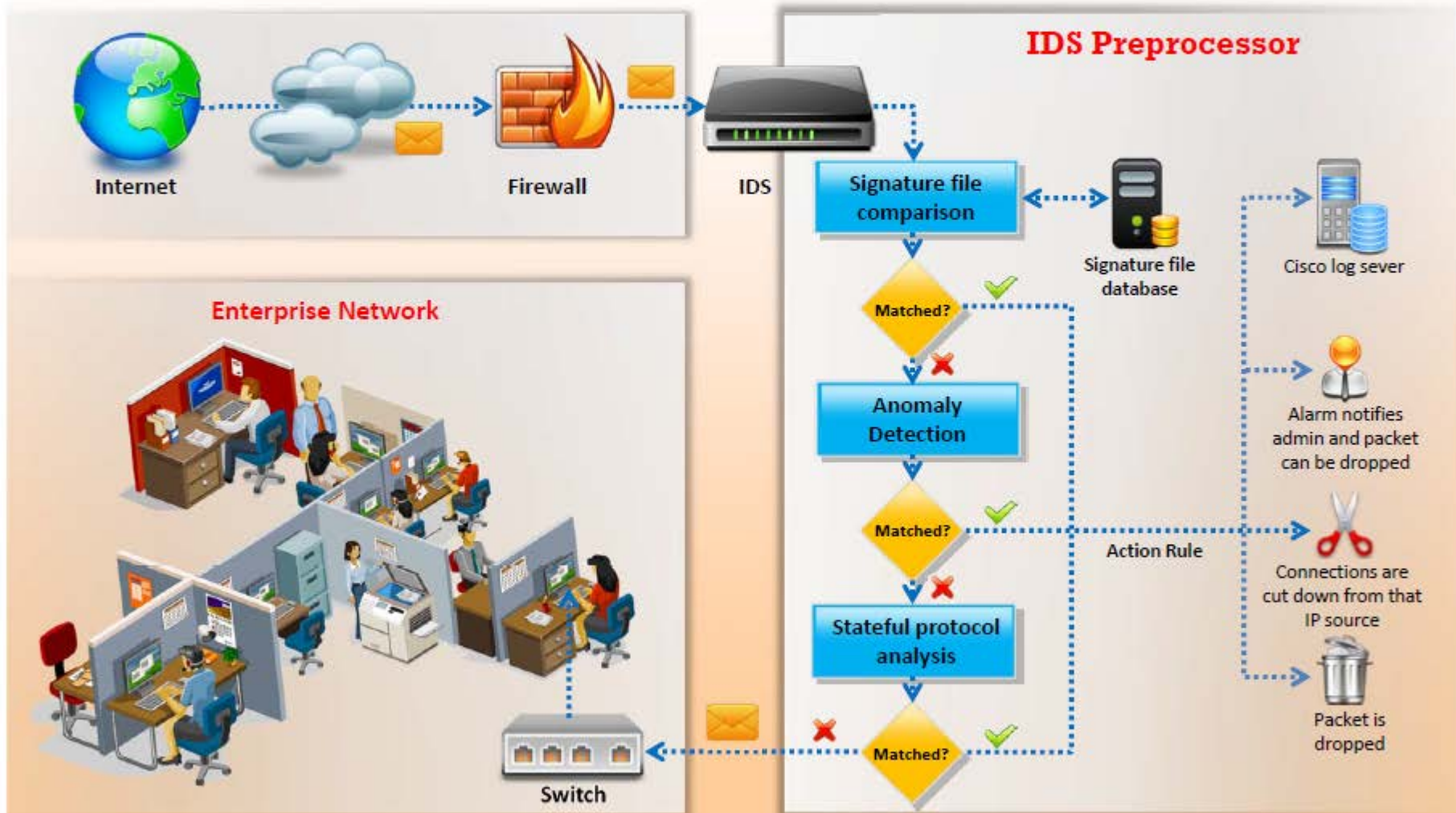


An intrusion detection system (IDS) **inspects all inbound and outbound network traffic** for suspicious patterns that may indicate a network or system security breach

The IDS **checks traffic** for signatures that match known intrusion patterns, and **signals an alarm** when a match is found



# How IDS Works





# Ways to **Detect** an Intrusion



## **Signature Recognition**

It is also known as misuse detection. Signature recognition tries to **identify events** that indicate misuse of a system resource



## **Anomaly Detection**

It detects the **intrusion based** on the fixed behavioral characteristics of the users and components in a computer system



## **Protocol Anomaly Detection**

In this type of detection, models are built to explore **anomalies** in the way vendors deploy the **TCP/IP specification**

# General Indications of Intrusions

## System Intrusions

- The presence of **new, unfamiliar** files, or programs
- Changes in file **permissions**
- **Unexplained** changes in a file's size
- **Rogue files** on the system that do not correspond to your master list of signed files
- Unfamiliar file names in **directories**
- Missing **files**

## Network Intrusions

- Repeated **probes** of the available services on your machines
- Connections from **unusual locations**
- Repeated login attempts from **remote hosts**
- **Arbitrary data** in log files, indicating attempts to cause a **DoS** or to crash a service

# General **Indications** of System Intrusions



Short or **incomplete** logs

01

Unusual graphic displays or **text messages**



02



Unusually **slow** system performance

03

Modifications to **system software** and configuration files



04



Missing logs or logs with **incorrect permissions** or ownership

05

System crashes or **reboots**



06



Gaps in the **system accounting**

07

**Unfamiliar** processes



08



# Types of Intrusion Detection Systems

## Network-Based Intrusion Detection Systems

01

- These mechanisms typically consist of a **black box** that is placed on the network in the promiscuous mode, listening for patterns indicative of an intrusion
- It detects malicious activity such as **Denial-of-Service attacks**, port scans, or even attempts to crack into computers by monitoring network traffic

## Host-Based Intrusion Detection Systems

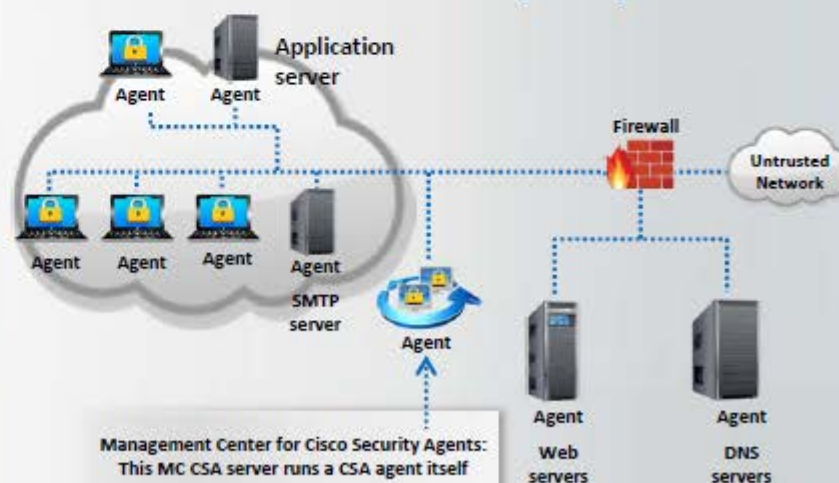
02

- These mechanisms usually include auditing for events that occur on a **specific host**
- These are not as common, due to the overhead they incur by having to **monitor each system event**

### Network-based IDS (NIDS)



### Host-based IDS (HIDS)



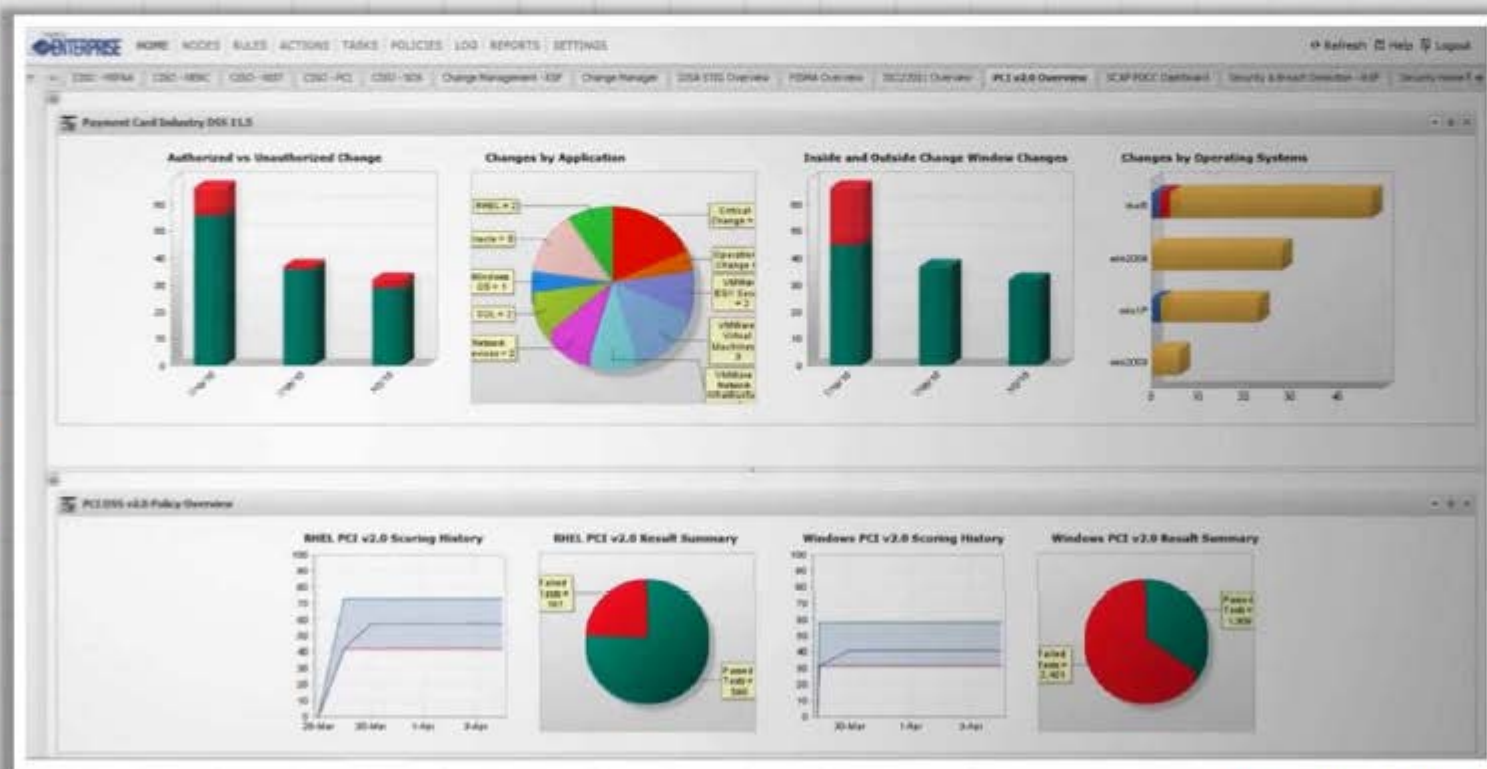
# System Integrity Verifiers (SIV)



System Integrity Verifiers **detect changes** in critical system components which help in detecting system intrusions

SIVs **compares a snapshot** of the file system with an existing baseline snapshot

Tripwire



<http://www.tripwire.com>



# Firewall

Firewalls are hardware and/or software designed to prevent **unauthorized access** to or from a private network

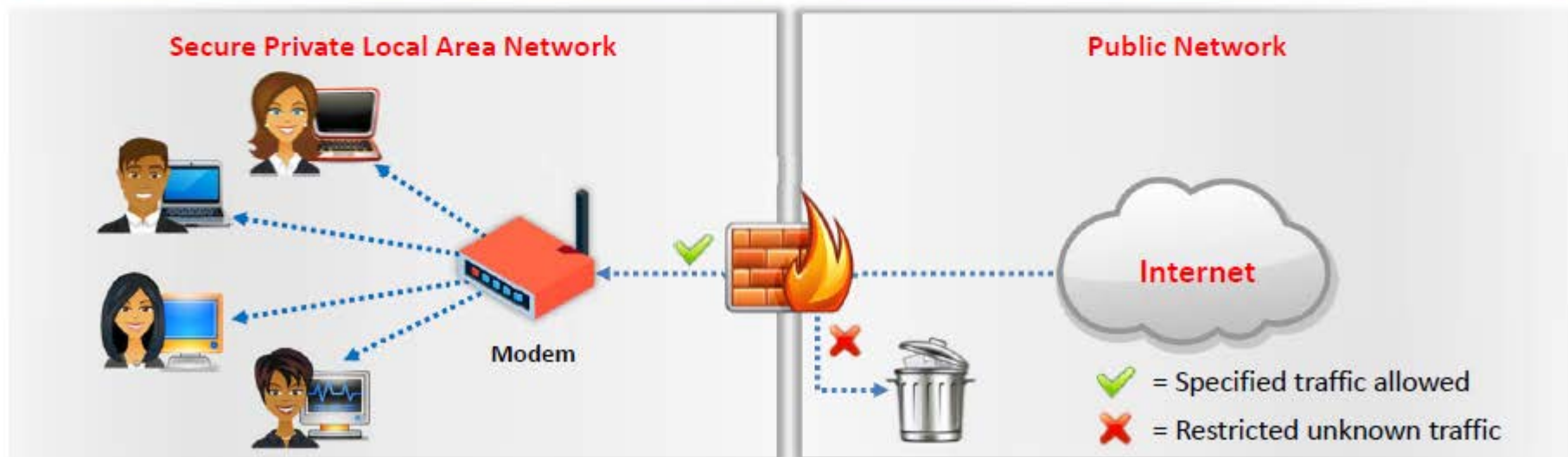


Firewalls **examine all messages entering or leaving the Intranet** and blocks those that do not meet the specified security criteria

They are placed at the junction or **gateway** between the two networks, which is usually a private network and a public network such as the Internet



Firewalls may be concerned with the type of traffic or with the **source** or **destination addresses** and ports





# Firewall Architecture

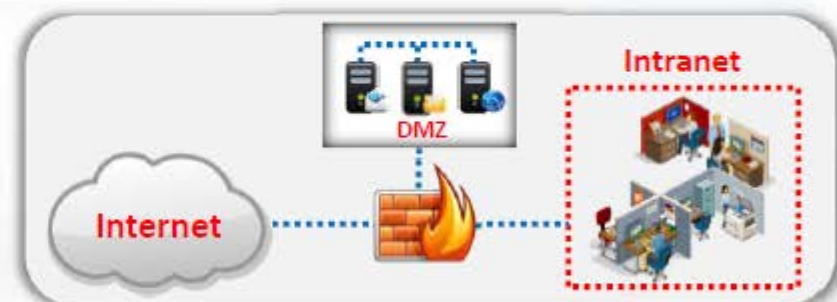
## Bastion Host

- ❏ Bastion host is a computer system designed and configured to protect **network resources** from attack
- ❏ Traffic entering or leaving the network passes through the firewall, it has two interfaces:
  - 🔵 **public interface** directly connected to the Internet
  - 🔵 **private interface** connected to the Intranet



## Screened Subnet

- ❏ The screened subnet or DMZ (additional zone) contains **hosts** that offer public services
- ❏ The DMZ zone **responds to public requests**, and has no hosts accessed by the private network
- ❏ Private zone can not be accessed by **Internet users**



## Multi-homed Firewall

- ❏ In this case, a firewall with two or more interfaces is present that allows further subdivision of the network based on the **specific security objectives** of the organization



# DeMilitarized Zone (DMZ)

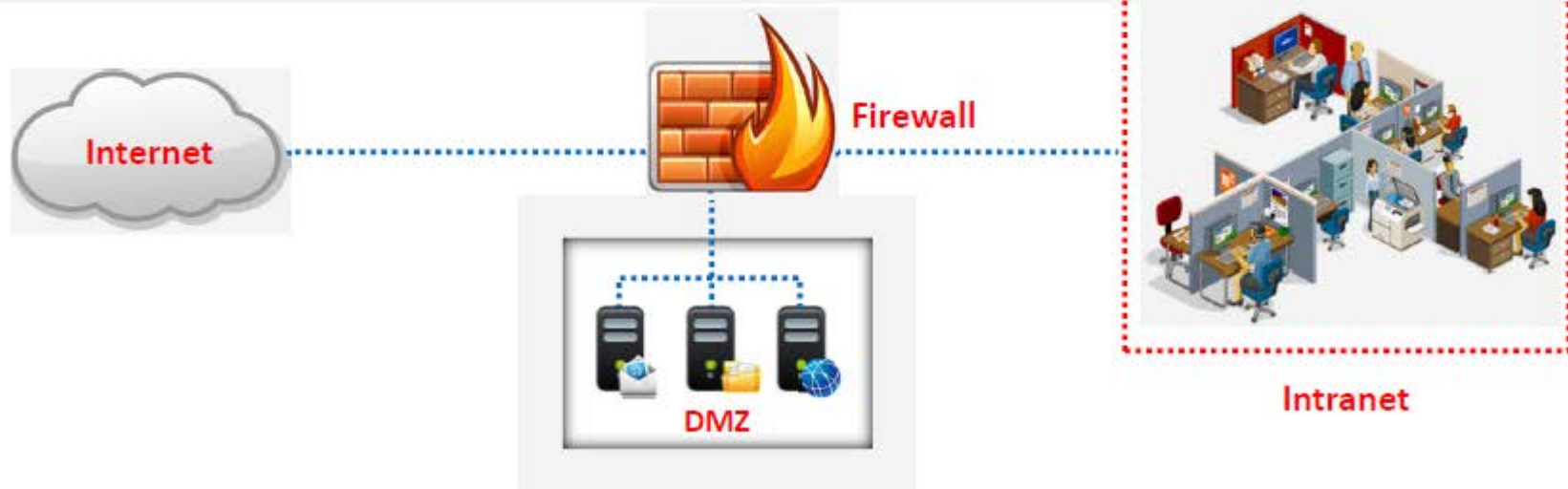
01

DMZ is a network that **serves as a buffer** between the internal secure network and insecure Internet



02

It can be created **using firewall with three or more network interfaces** assigned with specific roles such as Internal trusted network, DMZ network, and external un-trusted network



# Types of Firewall

## Packet Filters

01



02

## Circuit Level Gateways

## Application Level Gateways

03



04

## Stateful Multilayer Inspection Firewalls



# Packet Filtering Firewall



Packet filtering firewalls work at the **network layer of the OSI model** (or the IP layer of TCP/IP), they are usually a part of a router



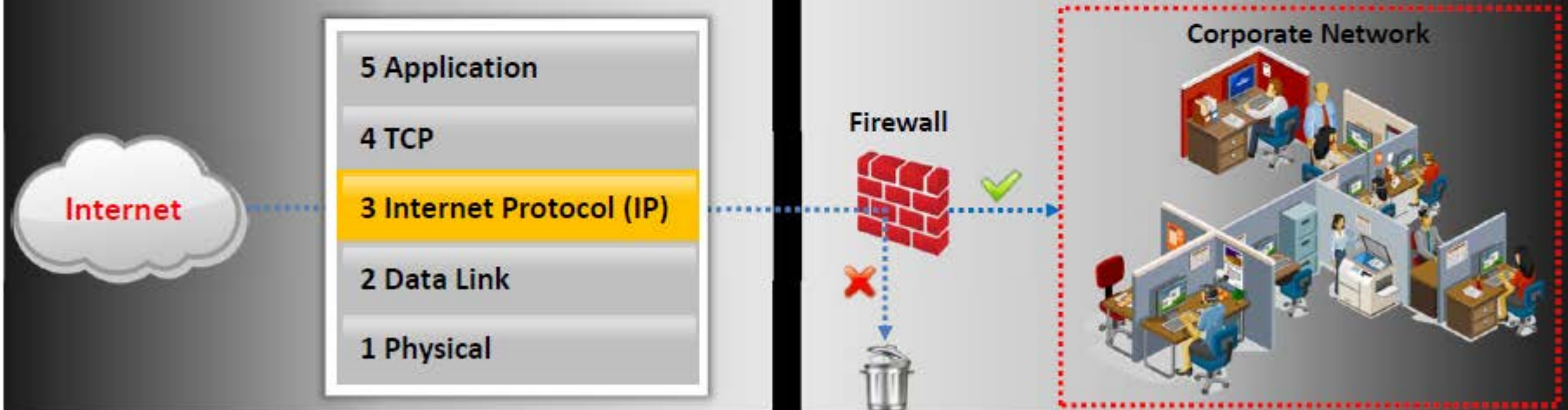
Depending on the **packet and the criteria**, the firewall can drop the packet and forward it, or send a message to the originator



In a packet filtering firewall, **each packet is compared** to a set of criteria before it is forwarded



Rules can include the source and the destination **IP address**, the source and the destination **port number**, and the **protocol** used



✓ = Traffic allowed based on source and destination **IP address, packet type, and port number**

✗ = Disallowed Traffic

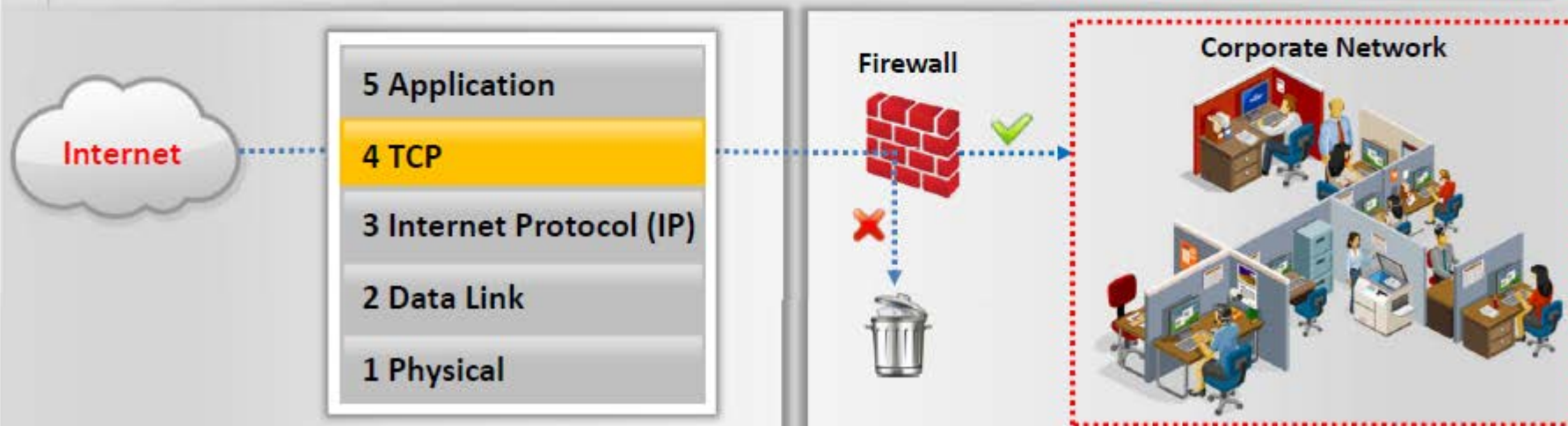
# Circuit-Level Gateway Firewall

Circuit-level gateways work at the **session layer of the OSI model** (or the TCP layer of TCP/IP)

Information passed to a **remote computer** through a circuit-level gateway appears to have originated from the gateway

They monitor **requests** to create sessions, and determine if those sessions will be allowed

Circuit proxy firewalls **allow or prevent** data streams, they do not filter individual packets



✓ = Traffic allowed based on **session rules**, such as when a session is initiated by a recognized computer

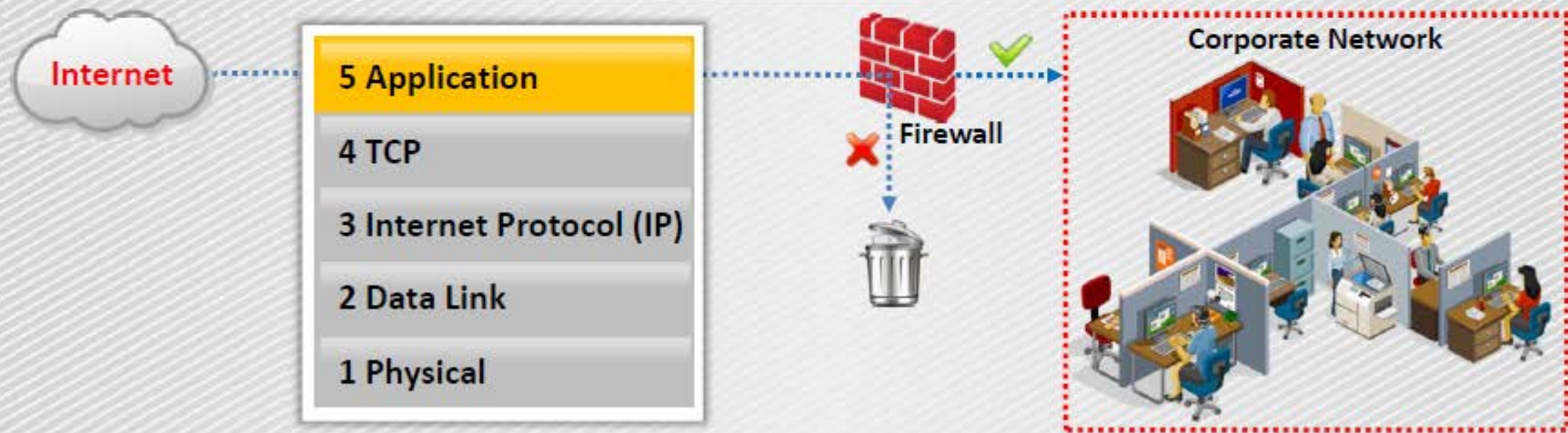
✗ = Disallowed Traffic



# Application-Level Firewall

- Application-level gateways (proxies) can filter packets at the **application layer of the OSI model** (or the application layer of TCP/IP)
- Incoming and outgoing traffic is **restricted to services** supported by proxy; all other service requests are denied

- Application-level gateways configured as a web proxy **prohibit** FTP, gopher, telnet, or other traffic
- Application-level gateways examine traffic and filter on **application-specific commands** such as http:post and get



- ✓ = Traffic allowed based on **specified applications** (such as a browser) or a **protocol**, such as FTP, or combinations
- ✗ = Disallowed Traffic



# Stateful Multilayer Inspection Firewall

Stateful multilayer inspection firewalls **combine the aspects of the other three types** of firewalls



They **filter packets** at the network layer of the OSI model (or the IP layer of TCP/IP), to determine whether session packets are legitimate, and they evaluate the contents of packets at the application layer



5 Application

4 TCP

3 Internet Protocol (IP)

2 Data Link

1 Physical

Firewall



Corporate Network



= Traffic is filtered at three layers based on a wide range of the **specified application, session, and packet filtering rules**



= Disallowed Traffic

# Honeypot



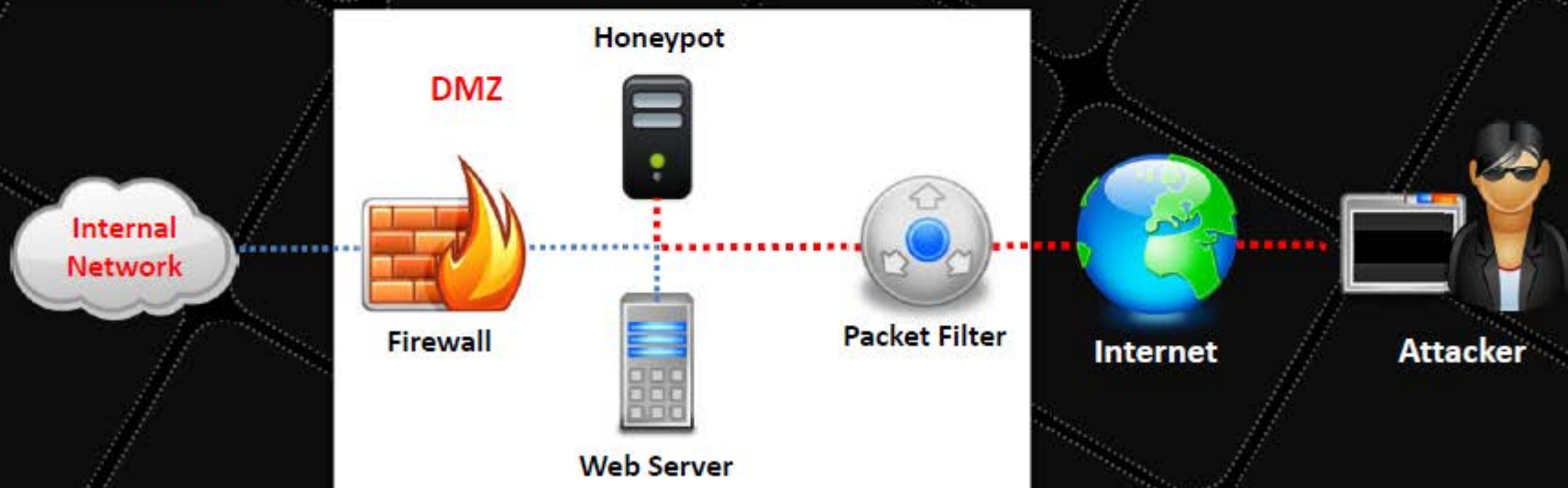
A honeypot is an information system resource that is expressly **set up to attract and trap people** who attempt to penetrate an organization's network



It has no authorized activity, does not have any production value, and any traffic to it is **likely a probe, attack, or compromise**



A honeypot can **log port access attempts, or monitor an attacker's keystrokes**. **These could be early warnings** of a more concerted attack





# Types of Honeypots

01

## Low-interaction Honeypots

- These honeypots simulate only a **limited number of services** and applications of a target system or network
- Can not be compromised completely
- Generally, set to collect higher level information about attack vectors such as network probes and worm activities
- Ex: Specter, Honeyd, and KFSensor



02

## High-interaction Honeypots

- These honeypots **simulates all services** and applications
- Can be **completely compromised** by attackers to get full access to the system in a controlled area
- Capture **complete information** about an attack vector such attack techniques, tools and intent of the attack
- Ex: Symantec Decoy Server and Honeynets



# Module Flow

01

**IDS, Firewall  
and Honeypot  
Concepts**

02

**IDS, Firewall  
and Honeypot  
Solutions**

03

**Evading IDS**

04

**Evading  
Firewalls**



05

**IDS/Firewall  
Evading Tools**

06

**Detecting  
Honeypots**

07

**IDS/Firewall  
Evasion Counter-  
measures**

08

**Penetration  
Testing**

# Intrusion Detection Tool: Snort



1

Snort is an open source network intrusion detection system, capable of performing real-time **traffic analysis and packet logging on IP networks**

2

It can perform **protocol analysis and content searching/matching**, and is used to detect a variety of **attacks and probes**, such as buffer overflows, stealth port scans, CGI attacks, SMB probes, and OS fingerprinting attempts

3

It uses a flexible **rules language** to describe traffic that it should collect or pass, as well as a **detection engine** that utilizes a modular plug-in architecture

4

## Uses of Snort:

- Straight packet sniffer like tcpdump
- Packet logger (useful for network traffic debugging, etc.)
- Network intrusion prevention system

```
Administrator: C:\Windows\system32\cmd.exe - snort -dev -i 1
NPF_{71849606-3085-4016-BA15-0EF4D88EBD36} MS Tunnel Interface Driver
C:\Snort\bin>snort -dev -i 1
Running in packet dump mode

--- Initializing Snort ---
Initializing Output Plugins!
pcap DAQ configured to passive.
The DAQ version does not support reload.
Acquiring network traffic from "\Device\NPF_{3358AA1D-5082-485B-8F75-0D08D5F2310D}"
Decoding Ethernet

--- Initialization Complete ---

--> Snort! <--
Version 2.9.5.6-UIN32 CRE (Build 2009)
By Martin Roesch & The Snort Team: http://www.snort.org/snort/snort-t
ean

Copyright (C) 1998-2013 Sourcefire, Inc., et al.
Using PCRE version: 8.10 2010-06-25
Using ZLIB version: 1.2.3

Commencing packet processing (pid=1520)
```

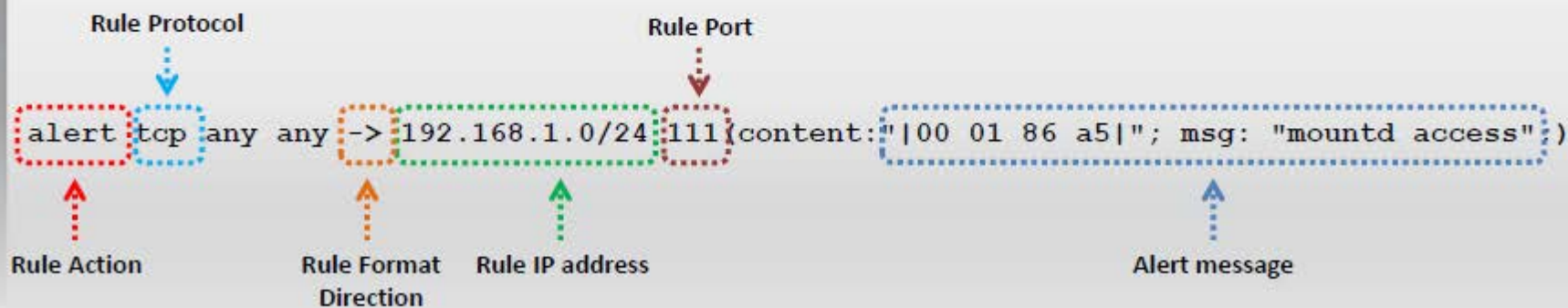
```
Administrator: C:\Windows\system32\cmd.exe - snort -i 1 -A console -c C:\Snort\etc\snort.conf -l C:\
01/15-23:26:26.257393 [**] [1:472:7] ICMP-INFO PING [**] [Classification: Potentially Bad Traffic] [Priority: 2] (ICMP) 10.0.0.29 -> 10.0.0.3
01/15-23:26:27.274231 [**] [1:472:7] ICMP-INFO PING [**] [Classification: Potentially Bad Traffic] [Priority: 2] (ICMP) 10.0.0.29 -> 10.0.0.3
01/15-23:26:28.288982 [**] [1:472:7] ICMP-INFO PING [**] [Classification: Potentially Bad Traffic] [Priority: 2] (ICMP) 10.0.0.29 -> 10.0.0.3
01/15-23:26:29.304263 [**] [1:472:7] ICMP-INFO PING [**] [Classification: Potentially Bad Traffic] [Priority: 2] (ICMP) 10.0.0.29 -> 10.0.0.3
01/15-23:26:30.320364 [**] [1:472:7] ICMP-INFO PING [**] [Classification: Potentially Bad Traffic] [Priority: 2] (ICMP) 10.0.0.29 -> 10.0.0.3
01/15-23:26:31.335445 [**] [1:472:7] ICMP-INFO PING [**] [Classification: Potentially Bad Traffic] [Priority: 2] (ICMP) 10.0.0.29 -> 10.0.0.3
01/15-23:26:32.350890 [**] [1:472:7] ICMP-INFO PING [**] [Classification: Potentially Bad Traffic] [Priority: 2] (ICMP) 10.0.0.29 -> 10.0.0.3
01/15-23:26:33.367181 [**] [1:472:7] ICMP-INFO PING [**] [Classification: Potentially Bad Traffic] [Priority: 2] (ICMP) 10.0.0.29 -> 10.0.0.3
01/15-23:26:34.382408 [**] [1:472:7] ICMP-INFO PING [**] [Classification: Potentially Bad Traffic] [Priority: 2] (ICMP) 10.0.0.29 -> 10.0.0.3
01/15-23:26:35.397887 [**] [1:472:7] ICMP-INFO PING [**] [Classification: Potentially Bad Traffic] [Priority: 2] (ICMP) 10.0.0.29 -> 10.0.0.3
01/15-23:26:36.414873 [**] [1:472:7] ICMP-INFO PING [**] [Classification: Potentially Bad Traffic] [Priority: 2] (ICMP) 10.0.0.29 -> 10.0.0.3
01/15-23:26:37.429215 [**] [1:472:7] ICMP-INFO PING [**] [Classification: Potentially Bad Traffic] [Priority: 2] (ICMP) 10.0.0.29 -> 10.0.0.3
```

<http://www.snort.org>

# Snort Rules

- Snort's rule engine enables **custom rules** to meet the needs of the network
- Snort rules help in differentiating between **normal Internet activities** and **malicious activities**
- Snort rules must be contained on a **single line**, the Snort rule parser **does not handle rules on multiple lines**
- Snort rules come with two logical parts:
  - **Rule header**: Identifies **rule's actions** such as alerts, log, pass, activate, dynamic, etc.
  - **Rule options**: Identifies rule's **alert messages**

## Example:





# Snort Rules: Rule Actions and IP Protocols

## Rule Actions

- The rule header stores the complete **set of rules** to identify a packet, and determines the action to be performed or what rule to be applied
- The rule action **alerts Snort** when it finds a packet that matches the rule criteria
- Three available actions in Snort:
  - ⚙ **Alert** - Generate an alert using the selected alert method, and then log the packet
  - ⚙ **Log** - Log the packet
  - ⚙ **Pass** - Drop (ignore) the packet



## IP Protocols

Three available IP protocols that Snort supports for suspicious behavior:

- I** TCP
- II** UDP
- III** ICMP



# Snort Rules: The Direction Operator and IP Addresses



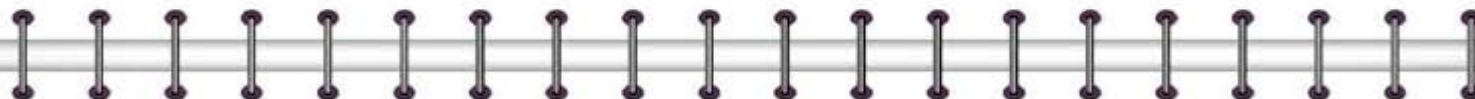
## The Direction Operator



- This operator indicates the direction of interest for the traffic; traffic can flow in either single direction or bi-directionally
- Example of a Snort rule using the Bidirectional Operator:



```
log !192.168.1.0/24 any <> 192.168.1.0/24 23
```



## IP Addresses



- Identifies IP address and port that the rule applies to
- Use keyword "**any**" to define any IP address
- Use numeric IP addresses qualified with a CIDR netmask
- Example IP Address Negation Rule:



```
alert tcp !192.168.1.0/24 any -> 192.168.1.0/24 111 (content:  
"|00 01 86 a5|"; msg: "external mounth access");
```

# Snort Rules: Port Numbers

Port numbers can be listed in different ways, including "any" ports, static port definitions, port ranges, and by negation

Port ranges are indicated with the **range operator** ":"

Example of a  
Port Negation

```
log tcp any any -> 192.168.1.0/24 !6000:6010
```



Protocols	IP address	Action
Log UDP any any ->	92.168.1.0/24 1:1024	Log UDP traffic coming from any port and destination ports ranging from 1 to 1024
Log TCP any any ->	192.168.1.0/24 :5000	Log TCP traffic from any port going to ports less than or equal to 5000
Log TCP any :1024 ->	192.168.1.0/24 400:	Log TCP traffic from the well known ports and going to ports greater than or equal to 400



# Intrusion Detection System: TippingPoint



- TippingPoint IPS is **in-line threat protection** that defends critical data and applications without affecting performance and productivity
- It contains over **8,700 security filters** written to address zero-day and known vulnerabilities



Network Criteria	
IPS Device / Segment Criteria	
Event Criteria	
Severity	Name
<input checked="" type="checkbox"/> Critical	1456: MS-SQL: Slammer-
<input type="checkbox"/> Low	1259: SMB: nbstat Query
<input type="checkbox"/> Low	8249: TCP: TCP Persist Time
<input checked="" type="checkbox"/> Critical	12957: HTTP: Apple QuickTime
<input checked="" type="checkbox"/> Critical	1456: MS-SQL: Slammer-Sap
<input type="checkbox"/> Low	4062: HTTP: Embedded OpenType/TrueType Font Download
<input type="checkbox"/> Low	4062: HTTP: Embedded OpenType/TrueType Font Download
<input type="checkbox"/> Low	4062: HTTP: Embedded OpenType/TrueType Font Download
<input type="checkbox"/> Low	4062: HTTP: Embedded OpenType/TrueType Font Download
<input type="checkbox"/> Low	8249: TCP: TCP Persist Time
<input type="checkbox"/> Low	4062: HTTP: Embedded OpenType/TrueType Font Download
<input type="checkbox"/> Low	4062: HTTP: Embedded OpenType/TrueType Font Download
<input type="checkbox"/> Low	8249: TCP: TCP Persist Time
<input type="checkbox"/> Low	4062: HTTP: Embedded OpenType/TrueType Font Download
<input type="checkbox"/> Low	7120: TCP: Segment Overlap With Different Data, e.g., Fragroute
<input checked="" type="checkbox"/> Major	2023: HTTP: Cross Site Scripting in GET Request
<input checked="" type="checkbox"/> Major	12639: HTTP: Apache HTTP Server X-Forwarded-For Denial-of-S

Time	Severity	Name	Category	Action	Hit Count	Profile
7/11/13 10:37:02 AM CDT	Critical	1456: MS-SQL: Slammer-Sap	Exploits	Block	1	1. HP IT Tes
7/11/13 10:37:02 AM CDT	Low	4062: HTTP: Embedded OpenType/TrueType Font Download	Security Policy	Block	1	1. HP IT Tes
7/11/13 10:37:02 AM CDT	Low	4062: HTTP: Embedded OpenType/TrueType Font Download	Security Policy	Block	1	1. HP IT Tes
7/11/13 10:37:02 AM CDT	Low	4062: HTTP: Embedded OpenType/TrueType Font Download	Vulnerabilities	Block	1	1. HP IT Tes
7/11/13 10:37:02 AM CDT	Low	4062: HTTP: Embedded OpenType/TrueType Font Download	Exploits	Block	1	1. HP IT Tes
7/11/13 10:37:01 AM CDT	Low	8249: TCP: TCP Persist Time	Security Policy	Block	4	1. HP IT Tes
7/11/13 10:37:01 AM CDT	Low	4062: HTTP: Embedded OpenType/TrueType Font Download	Security Policy	Block	1	1. HP IT Tes
7/11/13 10:37:01 AM CDT	Low	4062: HTTP: Embedded OpenType/TrueType Font Download	Security Policy	Block	4	1. HP IT Tes
7/11/13 10:37:01 AM CDT	Low	8249: TCP: TCP Persist Time	Security Policy	Block	2	1. HP IT Tes
7/11/13 10:37:01 AM CDT	Low	4062: HTTP: Embedded OpenType/TrueType Font Download	Security Policy	Block	1	1. HP IT Tes
7/11/13 10:37:01 AM CDT	Low	7120: TCP: Segment Overlap With Different Data, e.g., Fragroute	Traffic Normaliz...	Block	2	1. HP IT Tes
7/11/13 10:37:01 AM CDT	Major	2023: HTTP: Cross Site Scripting in GET Request	Vulnerabilities	Block	1	1. HP IT Tes
7/11/13 10:37:01 AM CDT	Major	12639: HTTP: Apache HTTP Server X-Forwarded-For Denial-of-S	Exploits	Block	1	1. HP IT Tes

<http://www8.hp.com>

# Intrusion Detection Tools

**CEH**  
Certified Ethical Hacker



**IBM Security Network  
Intrusion Prevention System**  
<http://www-03.ibm.com>



**OSSEC**  
<http://www.ossec.net>



**Peek & Spy**  
<http://networkdynamics.com>



**Cisco Intrusion Prevention  
Systems**  
<http://www.cisco.com>



**INTOUCH INSA-Network  
Security Agent**  
<http://www.ttinet.com>



**AIDE (Advanced Intrusion  
Detection Environment)**  
<http://aide.sourceforge.net>



**SilverSky**  
<https://www.silversky.com>



**SNARE (System iNtrusion Analysis  
& Reporting Environment)**  
<http://www.intersectalliance.com>



**IDP8200 Intrusion Detection  
and Prevention Appliances**  
<https://www.juniper.net>



**Vanguard Enforcer**  
<http://www.go2vanguard.com>

# Intrusion Detection Tools

(Cont'd)



**Check Point Threat Prevention Appliance**  
<http://www.checkpoint.com>



**FortiGate**  
<http://www.fortinet.com>



**fragroute**  
<http://www.monkey.org>



**Enterasys® Intrusion Prevention System**  
<http://www.extremenetworks.com>



**Next-Generation Intrusion Prevention System (NGIPS)**  
<http://www.sourcefire.com>



**AlienVault Unified Security Management**  
<http://www.alienvault.com>



**Outpost Network Security**  
<http://www.agnitum.com>



**Cyberoam Intrusion Prevention System**  
<http://www.cyberoam.com>



**Check Point IPS Software Blade**  
<http://www.checkpoint.com>



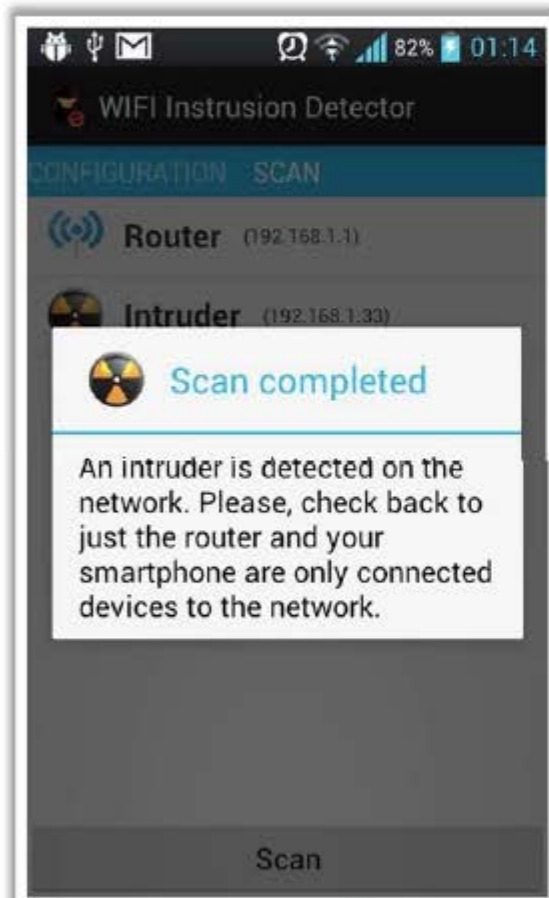
**McAfee Host Intrusion Prevention for Desktops**  
<http://www.mcafee.com>



# Intrusion Detection Tools for Mobile



## WiFi Intrusion Detection



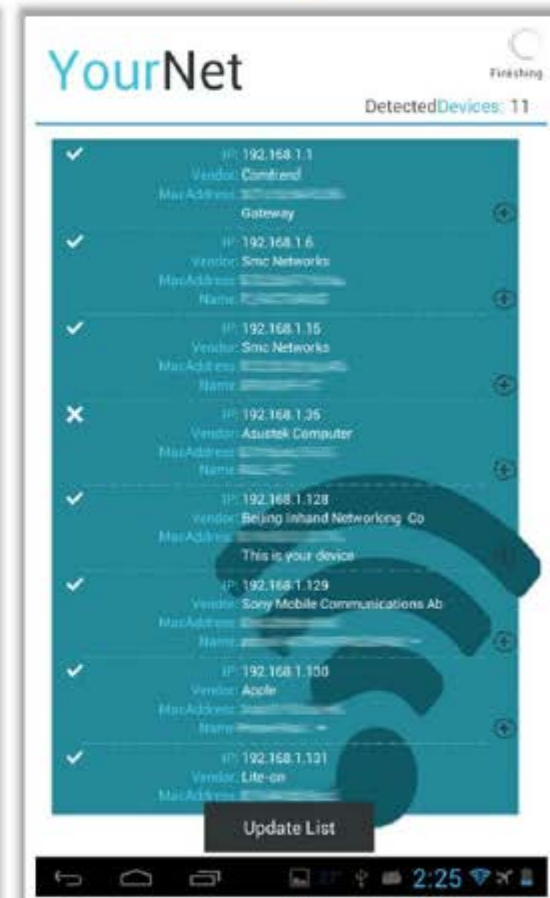
<https://play.google.com>

## Wifi Intruder Detector Pro



<https://play.google.com>

## Wifi Inspector

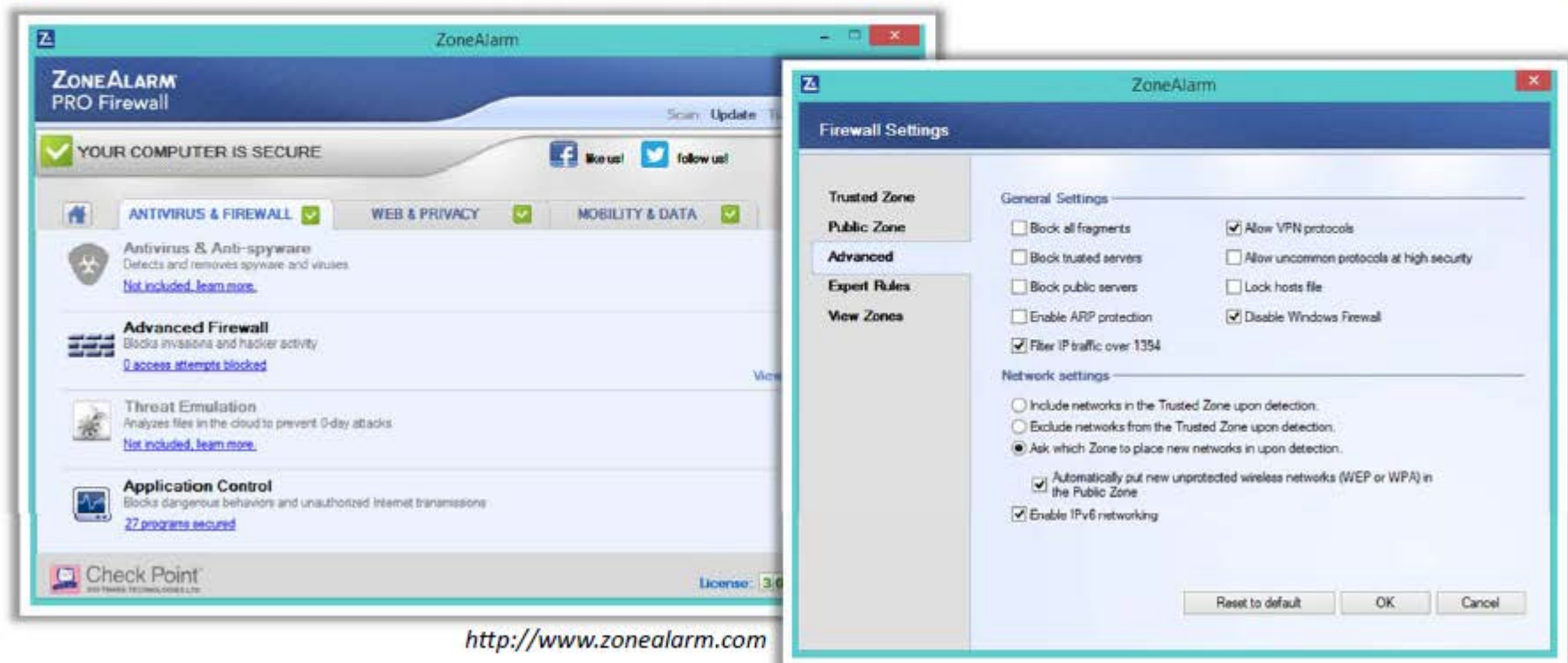


<https://play.google.com>

# Firewall: ZoneAlarm PRO Firewall 2015



- ZoneAlarm PRO Firewall 2015 **monitors programs for suspicious behavior** spotting and stopping new attacks that bypass traditional anti-virus protection
- It makes your **PC invisible to hackers** and stops spyware from sending your data out to the Internet



<http://www.zonealarm.com>

# Firewall: Comodo Firewall



- Keeps you updated on all **suspicious files**
- Prevention-based technology **stops viruses**
- Automatic updates for the most **current protection**



<http://personalfirewall.comodo.com>



# Firewalls



**Cisco ASA 1000V Cloud Firewall**  
<http://www.cisco.com>



**Novell BorderManager**  
<http://www.novell.com>



**Check Point Firewall Software Blade**  
<http://www.checkpoint.com>



**Untangle NG Firewall**  
<https://www.untangle.com>



**eScan Enterprise Edition**  
<http://www.escanav.com>



**Sonicwall**  
<http://www.sonicwall.com>



**Jetico Personal Firewall**  
<http://www.jetico.com>



**Online Armor**  
<http://www.online-armor.com>



**Outpost Security Suite**  
<http://free.agnitum.com>

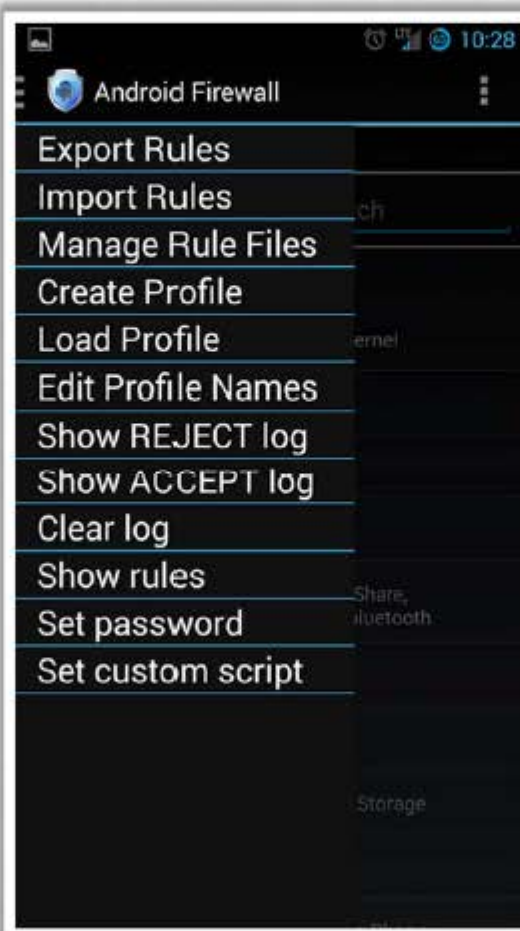
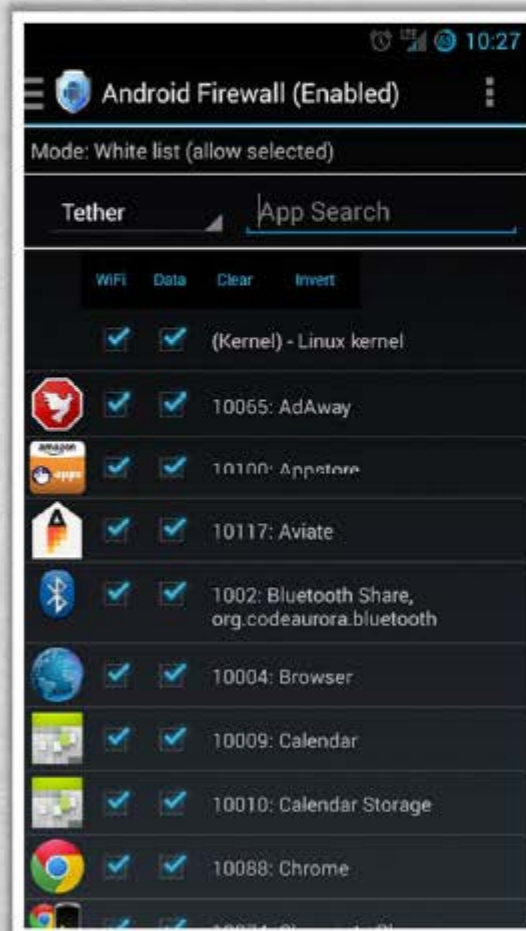


**FortiGate-5101C**  
<http://www.fortinet.com>

# Firewalls for Mobile: **Android Firewall** and **Firewall iP**

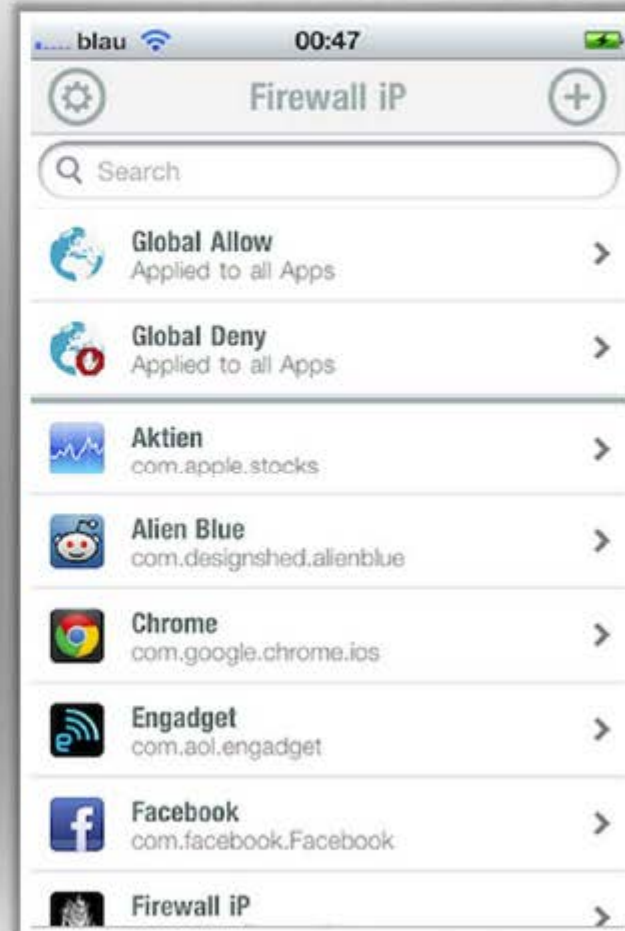


## Android Firewall



<https://play.google.com>

## Firewall iP



<http://cydia.saurik.com>

# Firewalls for Mobile



## Mobiwol: NoRoot Firewall

<http://www.mobiwol.com>



## Android Firewall Gold

<https://play.google.com>



## DroidWall

<https://code.google.com>



## Droid Firewall

<https://play.google.com>



## AFWall+

<https://github.com>



## Privacy Shield

<http://www.snoopwall.com>



## Firewall Plus

<http://squariolabs.com>



## aFirewall

<http://afirewall.wordpress.com>



## Root Firewall

<http://www.rootuninstaller.com>



## NoRoot Firewall

<https://play.google.com>



# Honeypot Tools: KFSensor and SPECTER

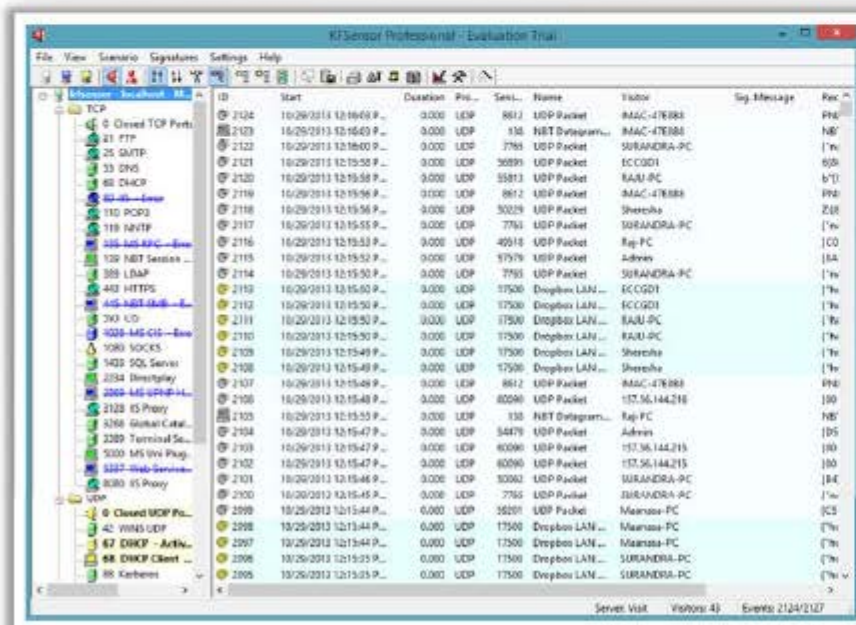
CEH  
Certified Ethical Hacker

## KFSensor

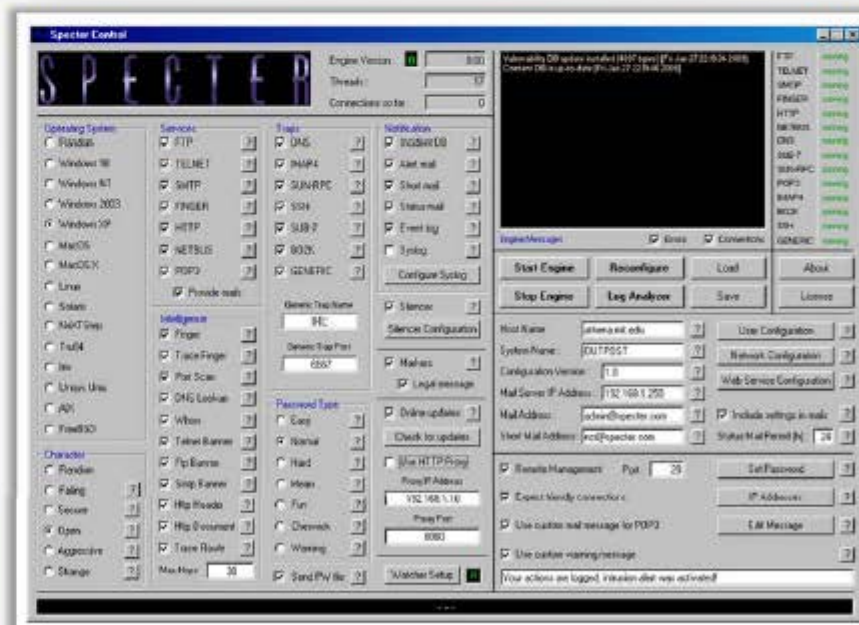
KFSensor is a **host-based** Intrusion Detection System (IDS) that acts as a honeypot to attract and detect hackers and worms by **simulating vulnerable system services** and **Trojans**

## SPECTER

SPECTER is a smart **honeypot-based** intrusion detection system that offers common **Internet services** such as SMTP, FTP, POP3, HTTP, and TELNET which appear perfectly normal to the attackers but in fact are traps



<http://www.keyfocus.net>



<http://www.specter.com>

# Honeypot Tools



**LaBrea Tarpit**

<http://labrea.sourceforge.net>



**WinHoneyd**

<http://www2.netvigilance.com>



**PatriotBox**

<http://www.alkasis.com>



**HIHAT**

<http://hihat.sourceforge.net>



**Kojoney**

<http://kojoney.sourceforge.net>



**Argos**

<http://www.few.vu.nl>



**HoneyBOT**

<http://www.atomicsoftwaresolutions.com>



**Glastopf**

<http://glastopf.org>



**Google Hack Honeypot**

<http://ghh.sourceforge.net>



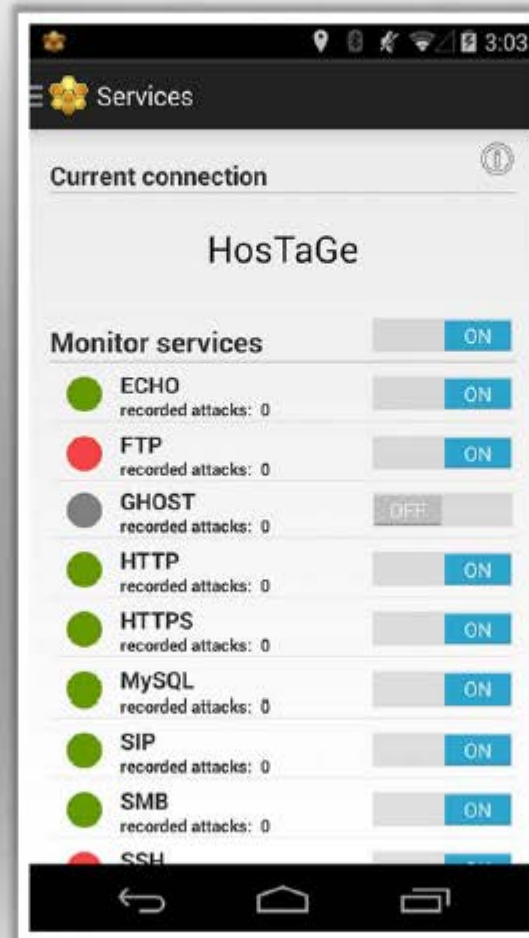
**Send-Safe Honeypot Hunter**

<http://www.send-safe.com>

# HoneyPot Tool for Mobile: HosTaGe



- HosTaGe is generic honeypot for mobile devices that aim on the **detection of malicious, wireless network environments**
- As most malware propagate over the network via specific protocols, a low-interaction honeypot located at a mobile device can **check wireless networks for actively propagating malware**



<http://www.tk.informatik.tu-darmstadt.de>



# Module Flow

01

**IDS, Firewall  
and Honeypot  
Concepts**

02

**IDS, Firewall  
and Honeypot  
Solutions**

03

**Evading IDS**

04

**Evading  
Firewalls**



05

**IDS/Firewall  
Evading Tools**

06

**Detecting  
Honeypots**

07

**IDS/Firewall  
Evasion Counter-  
measures**

08

**Penetration  
Testing**

# Insertion Attack

An IDS **blindly believes and accepts a packet** that an end system rejects

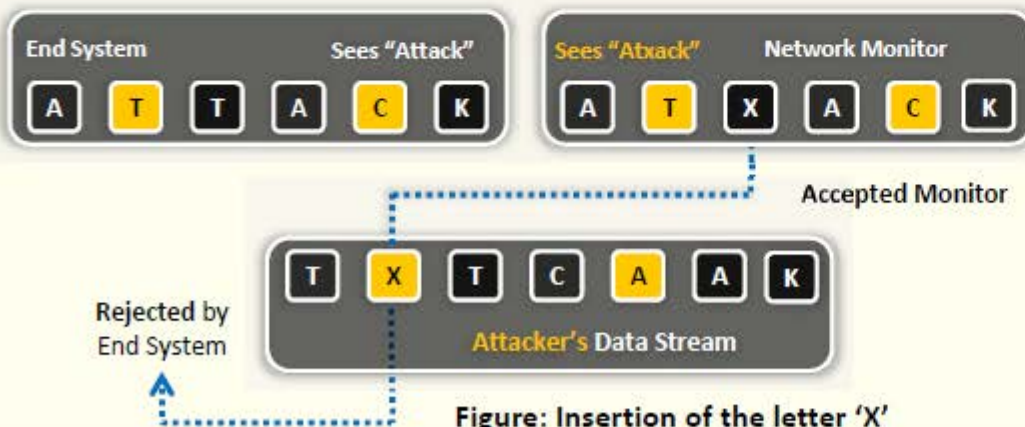
This attack occurs when **NIDS is less strict in processing packets**

Hence, the IDS gets **more packets than the destination**



An attacker exploits this condition and **inserts data into the IDS**

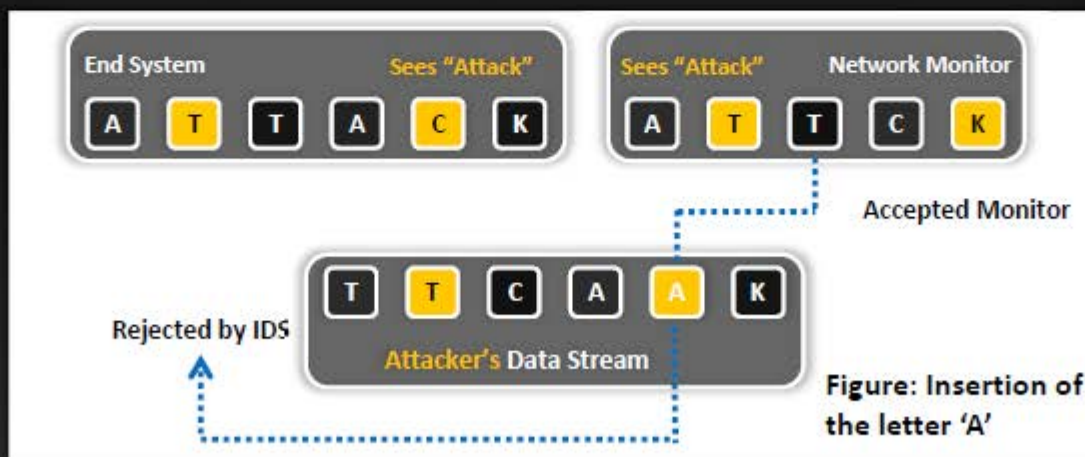
Attacker obscures extra traffic and IDS concludes traffic is harmless



- An attacker sends one-character packets to the target system via the IDS with **varying TTL** such that some packets reach to the IDS but not the target system
- This will result in the IDS and the target system having **two different character strings**

# Evasion

- 1 In this evasion technique, an end system **accepts a packet** that an IDS rejects
- 2 Using this technique, an attacker **exploits** the host computer
- 3 Attacker sends **portions of the request** in packets that the IDS mistakenly rejects, allowing the removal of parts of the stream from the IDS
- 4 For example, if the malicious sequence is sent **byte-by-byte**, and one byte is rejected by the IDS, the IDS cannot detect the attack
- 5 Here, the IDS gets fewer packets than the destination





# Denial-of-Service Attack (DoS)

01

Many IDSs use a **centralized server for logging** alerts

02

If attackers know the **IP address of the centralized server** they can perform **DoS** or other hacks to slow down or crash the server

03

As a result, attackers **intrusion attempts will not be logged**

Using this evasion technique, an attacker:

1

Causes the device to lock up

2

Causes personnel to be unable to investigate all the alarms

3

Causes more alarms than can be handled by management systems (such as databases, etc.)

4

Fills up disk space causing attacks to not be logged

5

Consumes the device's processing power and allows attacks to sneak by

# Obfuscating

- 1 An IDS can be evaded by obfuscating or **encoding the attack payload** in a way that the target computer understands but the IDS will not
- 2 Attackers can **encode attack patterns in unicode** to bypass IDS filters, but be understood by an IIS web server
- 3 **Polymorphic code** is another means to circumvent **signature-based IDSs** by creating unique attack patterns, so that the attack does not have a single detectable signature
- 4 Attackers manipulate the **path referenced in the signature** to fool the HIDS
- 5 Attacks on **encrypted protocols** such as HTTPS are obfuscated if the attack is encrypted

# False Positive Generation

Attackers with the knowledge of the target IDS, **craft malicious packets** just to generate alerts



These packets are sent to the IDS to generate a **large number of false positive alerts**



Attackers then use these false positive alerts to **hide real attack traffic**



Attackers can bypass IDS unnoticed as it is **difficult to differentiate** the attack traffic from the large volume of false positives



# Session Splicing

1



A technique used to bypass IDS where an attacker **splits the attack traffic** in to many packets such that no single packet triggers the IDS

2



It is effective against IDSs **that do not reconstruct** packets before checking them against intrusion signatures

3



If attackers are aware of **delay in packet reassembly** at the IDS, they can add delays between packet transmissions to bypass the reassembly

4



Many IDSs **stops reassembly** if they do not receive packets within a certain time

5



IDS will stop working if the target host keeps session active for a time longer than the **IDS reassembly time**

6



Any attack attempt after a successful splicing attack will **not be logged** by the IDS

# Unicode Evasion Technique

1



Unicode is a **character coding system** to support the worldwide interchange, processing, and display of the written texts

2



**For example**, / → %u2215, e → %u00e9 (UTF-16) and © → %c2%a9, ≠ → %e2%89%a0 (UTF-8)

3



Attackers can convert **attack strings to Unicode characters** to avoid pattern and signature matching at the IDS

4



Attackers can **encode URLs in HTTP requests** using Unicode characters to bypass **HTTP-based** attack detection at the IDS

# Fragmentation Attack

Fragmentation can be used as an attack vector when **fragmentation timeouts** vary between IDS and host



If fragment reassembly timeout is **10 seconds** at the IDS and **20 seconds** at the target system, attackers will send the second fragment after **15 seconds** of sending the first fragment



In this scenario, the IDS will **drop the fragment** as the second fragment is received after its reassembly time but the target system will reassemble the fragments



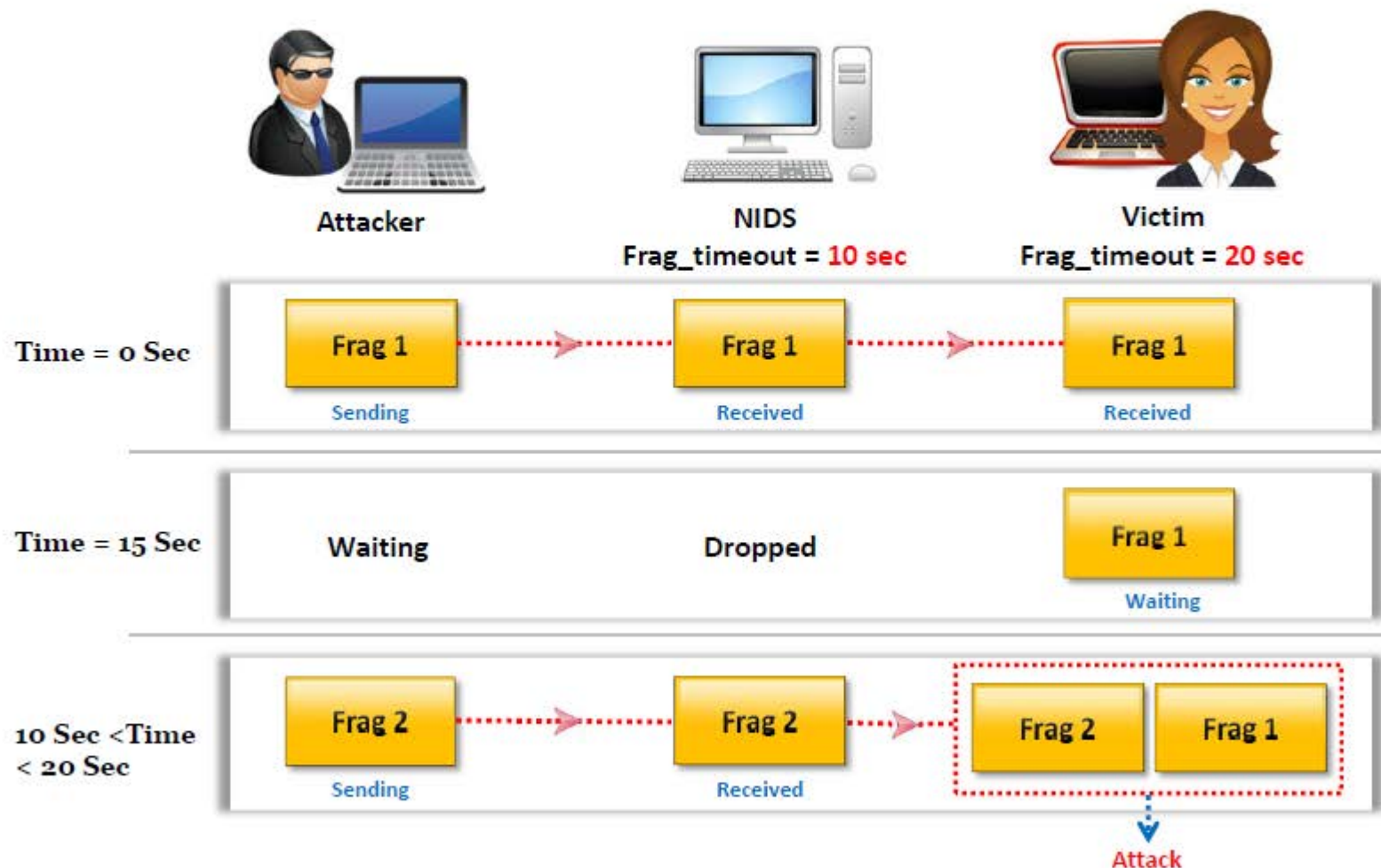
Attackers will keep sending the fragments with **15 second delays** until all the attack payload is reassembled at the target system





# Fragmentation Attack

(Cont'd)



# Fragmentation Attack

(Cont'd)

A similar fragmentation attack works when the **IDS timeout exceeds the victim's**

**1**

Victim and IDS receive **frag 2 and 4** out of 4 fragments, both carry a false payload

**2**

Victim drops these two fragments after **30 sec**, and does not send ICMP since frag 1 never received

**3**

Victim and IDS receive **frag 1 and 3** out of 4 fragments

**4**

IDS reassembles 4 received fragments, but computed net **checksum** is invalid, so packet is dropped

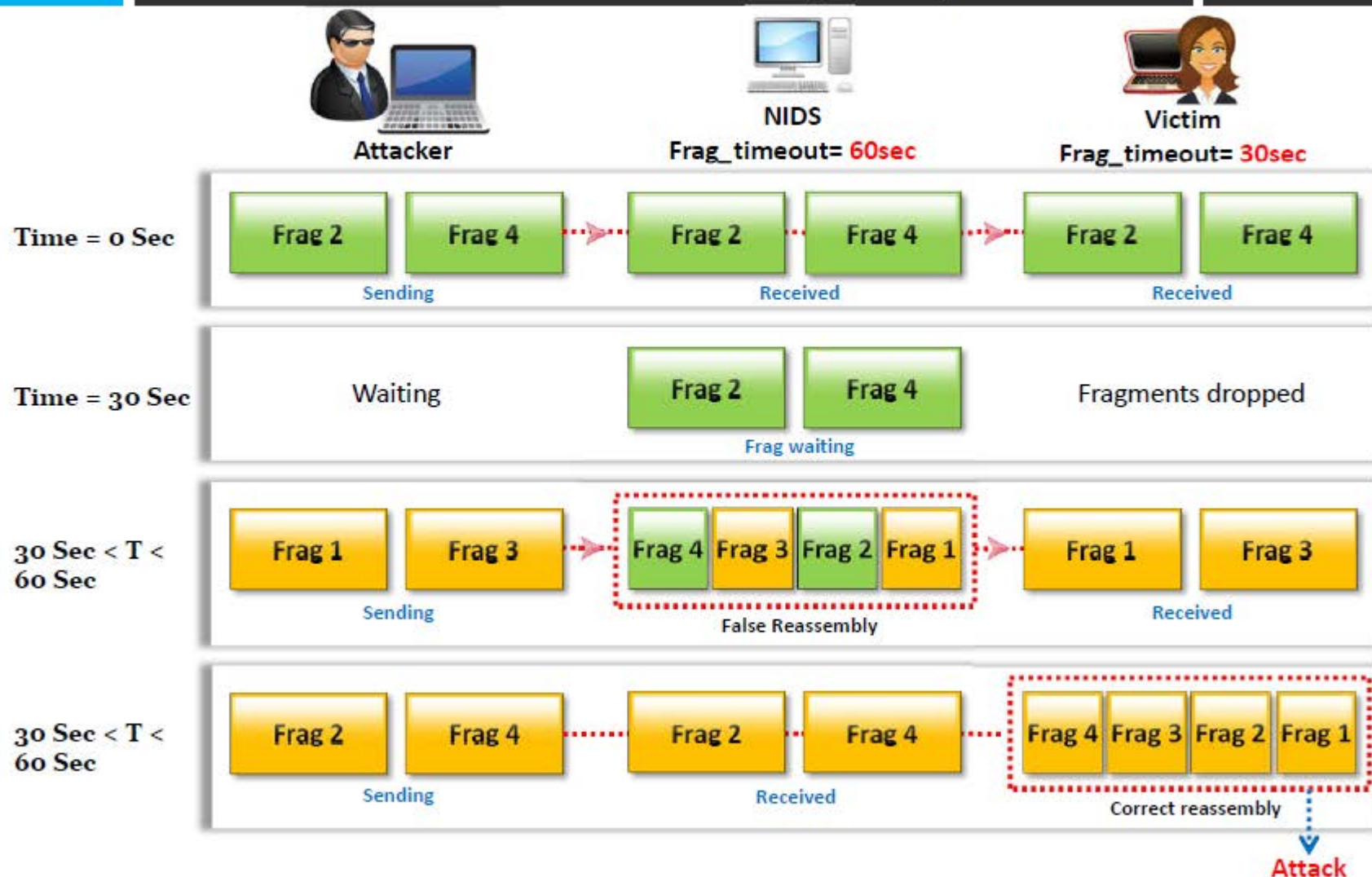
**5**

Victim and IDS receive real **frag 2 and 4** out of 4 fragments

**6**

Victim reassembles 4 received fragments and is **attacked**; IDS times out frag 2 and 4 and drops

# Fragmentation Attack (Cont'd)





# Overlapping Fragments



An IDS evasion technique is to **craft a series of packets** with TCP sequence numbers configured to overlap



For example, the first packet will include **80 bytes** of payload, but the second packet's sequence number will be **76 bytes** after the start of the first packet



When the target computer **reassembles the TCP stream**, it must decide how to handle the four overlapping bytes



Some OS will take the **original fragments with a given offset** (e.g., Windows W2K/XP/2003) and some operating systems will take the subsequent fragments with a given offset (e.g., Cisco IOS)



Attacker



Windows XP



Cisco IOS



Sending



Received



Received



Sending



Reassembled



Reassembled

# Time-To-Live Attacks

- These attacks require the attacker to have a **prior knowledge of the topology** of the victim's network
- This information can be obtained using tools such as **traceroute** which gives information on the **number of routers between the attacker and the victim**

Attacker breaks malicious traffic into  
**3 fragments**

1

4

Attacker sends **frag 3** with high TTL

Attacker sends frag 1 with **high TTL**,  
false frag 2 with low TTL

2

5

IDS reassembles 3 fragments into  
meaningless packet and **drops**

IDS receives both fragments, victim  
receives **first fragment only**

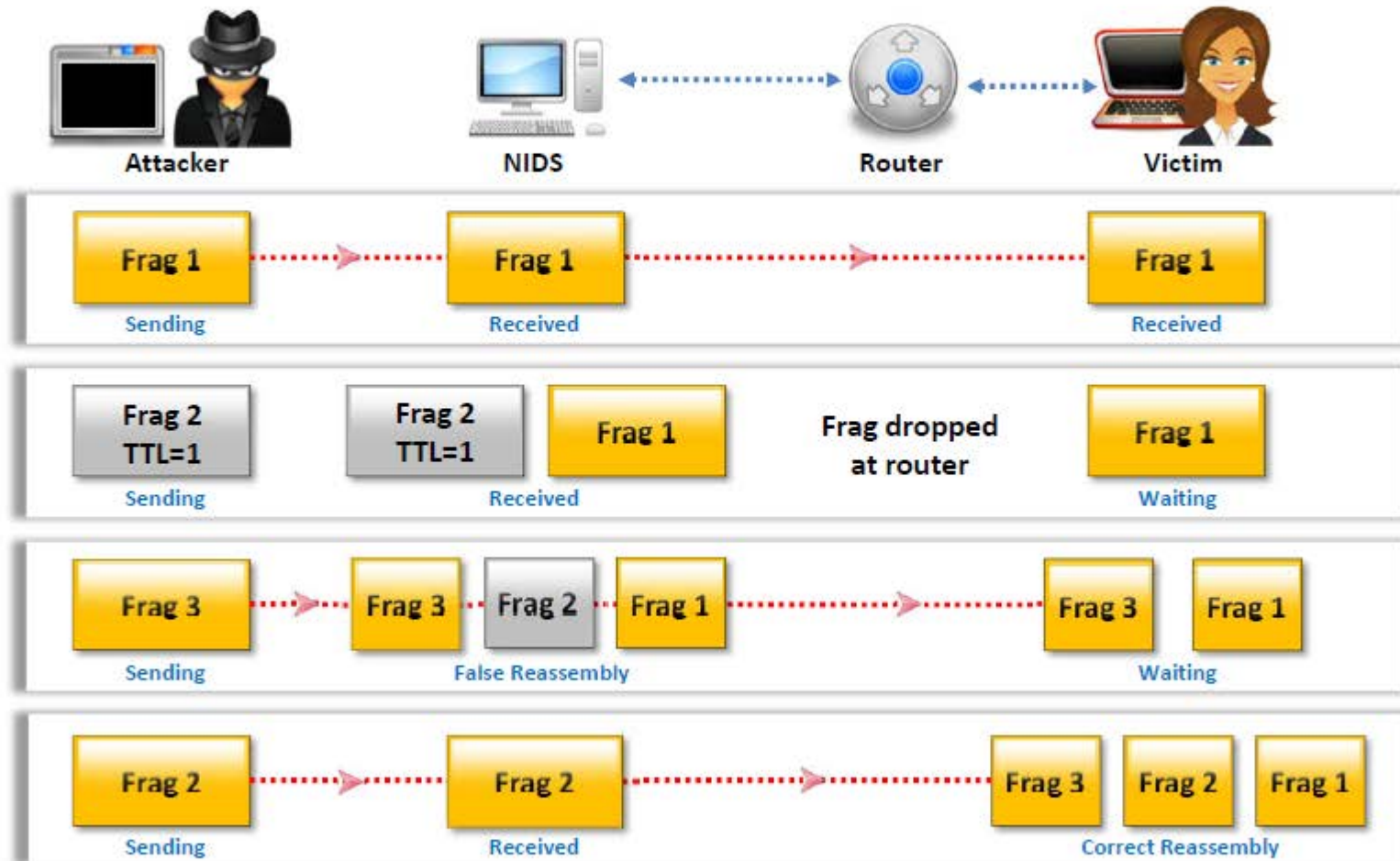
3

6

Victim receives real frag 2, and **suffers attack**, while no log entry created

# Time-To-Live Attacks

(Cont'd)





# Invalid RST Packets

TCP uses 16-bit checksum field for **error-checking** of the header and data

01

**Reset (RST) flag** in a TCP header is used to close a TCP connection

02

In invalid reset attack, attackers send **RST packet** to the IDS with an invalid checksum

03

IDS stop processing the packet thinking that the **TCP communication session** has ended but the target system will receive the packet

04

The target system checks the **RST packet's checksum** and drops it

05

The attack enables **attackers to communicate** with the target system while the IDS thinks that the communication has ended

06

# Urgency Flag

01

Urgent (URG) flag in the TCP header is used to mark the data that require **urgent processing** at the receiving end



02

If the URG flag is set, the TCP protocol sets the Urgent Pointer field to a **16-bit offset value** that points to the last byte of urgent data in the segment



03

Many IDSs do **not consider the urgent pointer** and process all the packets in the traffic whereas the target system processes only the urgent data



04

This results in the IDS and the target systems having **different set of packets**, which can be exploited by attackers to pass the attack traffic



## Urgency flag attack example

"1 Byte data, next to Urgent data, will be lost, when Urgent data and normal data are combined."

Packet 1: ABC

Packet 2: DEF Urgency Pointer: 3

Packet 3: GHI

End result: ABCDEFHI

- This example illustrates how the urgency flag works in conjunction with the urgency pointer
- According to the RFC 1122, the urgency pointer causes one byte of data next to the urgent data to be lost when urgent data is combined with normal data



# Polymorphic Shellcode

01

Most IDSs contain **signatures** for commonly used strings within shellcode



02

This is easily bypassed by using **encoded shellcode** containing a stub that decodes the shellcode that follows



03

This means that shellcode can be completely different **each time it is sent**



04

Polymorphic shellcode allows attackers to **hide their shellcode** by encrypting it in a simplistic form



05

It is difficult for IDSs to identify this data as **shellcode**



06

This method also hides the **commonly used strings** within shellcode, making shellcode signatures useless





# ASCII Shellcode

ASCII shellcode includes characters which are present only in **ASCII standard**

Attackers can use ASCII shellcode to bypass the IDS signature as the **pattern matching** does not work effectively with the ASCII values

Scope of ASCII shellcode is **limited** as all assembly instructions cannot be converted to ASCII values directly

This limitation can be overcome by using other **sets of instructions** for converting to ASCII values properly

**The following is an ASCII shellcode example:**

```
char shellcode[] =  
"LLLLYhb0pLX5b0pLHSSPPWQPPaPWSUTBRDJfh5t  
DS"  
"RajYX0Dka0TkafhN9fYf1Lkb0Tkdjfy0Lkf0Tkg  
fh"  
"6rfYf1Lki0tkkh95h8Y1LkmjpY0Lkq0tkrh2wnu  
X1"  
"Dks0tkwjfX0Dkx0tkx0tkyCjnY0LkzC0TkzCCjt  
X0"  
"DkzC0tkzCj3X0Dkz0TkzC0tkzChjG3IY1LkzCCC  
C0"  
"tkzChpfcMX1DkzCCCC0tkzCh4pCnY1Lkz1TkzCC  
CC"  
"fhJGfXf1Dkzf1tkzCCjHX0DkzCCCCjvY0LkzCCC  
jd"  
"X0DkzC0TkzCjWX0Dkz0TkzCjdX0DkzCjXY0Lkz0  
tk"  
"zMdgvvn9F1r8F55h8pG9wnuvjrNfrVx2LGkG3ID  
pf"  
"cM2KgmnJGgbinYshdvD9d";
```

When executed, the shellcode above executes a **"/bin/sh"** shell. **'bin'** and **'sh'** are contained in the last few bytes of the shellcode.

# Application-Layer Attacks



Applications accessing media files (audio, video and images) **compress** them to smaller size for maximizing data transfer rate



IDS cannot verify the **signature of compressed file** format



This enables an attacker to **exploit the vulnerabilities** in compressed data



IDS can recognize particular conditions favorable for attack but other alternative forms of attack are also possible, for example, various integer values can be used to **exploit integer overflow vulnerabilities**



This makes the detection of attack traffic **extremely difficult** at the IDS

# Desynchronization – Pre-Connection SYN

01

If a SYN packet is received **after the TCP control block is opened**, the IDS resets the appropriate sequence number to match that of the newly received SYN packet



02

Attackers send **fake SYN packets** with a completely invalid sequence number to desynchronize the IDS



03

This **stops IDS** from monitoring all, legitimate and attack, traffic





# Desynchronization – Post-Connection SYN

1

For this technique, attempt to **desynchronize the IDS** from the actual sequence numbers that the kernel is honoring

2

Send a **post connection SYN packet** in the data stream, which will have **divergent sequence** numbers, but otherwise meet all of the necessary criteria to be accepted by the target host

3

However, the target host will ignore this **SYN packet**, as it references an already established connection

4

The intent of this attack is to get the IDS to **resynchronize** its notion of the sequence numbers to the new SYN packet

5

It will then ignore any data that is a **legitimate part of the original stream**, because it will be awaiting a different sequence number

6

Once succeeded in resynchronizing the IDS with a SYN packet, send an **RST packet with the new sequence number** and close down its notion of the connection

# Other Types of Evasion

## Encryption

When the attacker has already established an **encrypted session with the victim**, it results in the most effective evasion attack



## Flooding

The attacker sends loads of **unnecessary traffic to produce noise**, and if IDS does not analyze the noise traffic well, then the true attack traffic may go undetected



# Module Flow

01

**IDS, Firewall  
and Honeypot  
Concepts**

02

**IDS, Firewall  
and Honeypot  
Solutions**

03

**Evading IDS**

04

**Evading  
Firewalls**



05

**IDS/Firewall  
Evading Tools**

06

**Detecting  
Honeypots**

07

**IDS/Firewall  
Evasion Counter-  
measures**

08

**Penetration  
Testing**



# Firewall Identification:

## Port Scanning

Port scanning is used to **identify open ports** and services running on these ports

Open ports can be further probed to identify the **version of services**, which helps in finding vulnerabilities in these services

Some firewalls **will uniquely identify themselves** in response to simple port scans

For example: **Check Point's FireWall-1** listens on TCP ports 256, 257, 258, and 259, NetGuard GuardianPro firewall listens on TCP 1500 and UDP 1501



# Firewall Identification:

## Firewalking

01



A technique that uses TTL values to determine gateway **ACL filters** and map networks by analyzing IP packet responses

Attackers send a TCP or UDP packet to the targeted firewall with a **TTL set to one hop greater** than that of the firewall

02



03



If the packet makes it through the gateway, it is forwarded to the next hop where the TTL equals one and elicits an ICMP "**TTL exceeded in transit**" to be returned, as the original packet is discarded

This method helps locate a firewall, additional probing permits **fingerprinting** and **identification of vulnerabilities**

04



# Firewall Identification:

## Banner Grabbing



Banners are **service announcements** provided by services in response to connection requests, and often carry vendor version information



Banner grabbing is a simple method of **fingerprinting** that helps in detecting the vendor of a firewall, and the firmware's version



The three main services which send out banners are **FTP**, **telnet**, and **web servers**



An example of SMTP banner grabbing is:  
**telnet mail.targetcompany.org 25**

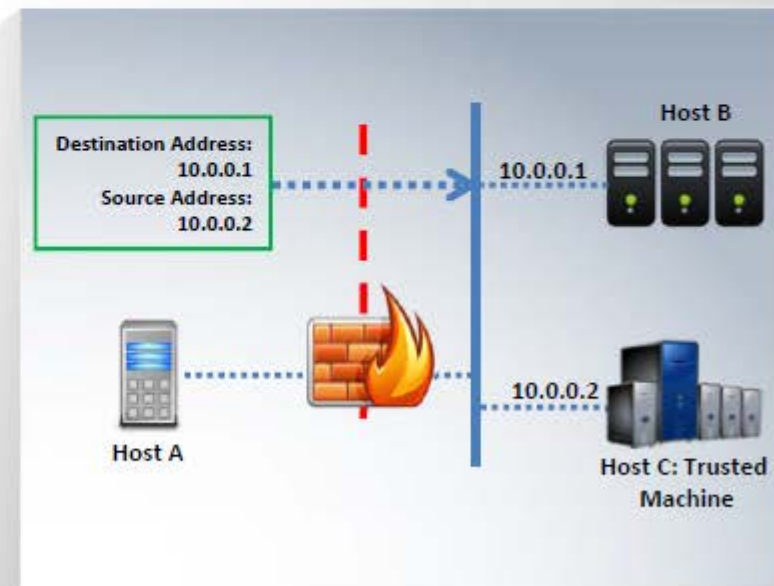




# IP Address Spoofing

- IP address spoofing is a hijacking technique in which an attacker **masquerades as a trusted host** to conceal his identity, spoof a Web site, hijack browsers, or gain unauthorized access to a network
- Attackers modify the **addressing information** in the IP packet header and the source address bits field in order to bypass the firewall

- For example, let's consider **three hosts**: A, B and C
- Host **C is a trusted machine** of host B
- Host A masquerades to be as host C by **modifying the IP address** of the malicious packets that he intends to send to the host B
- When the **packets are received**, host B thinks that they are from host C, but are actually from host A



# Source Routing

Source routing allows the sender of a packet to partially or completely **specify the route**, the packet takes through the network



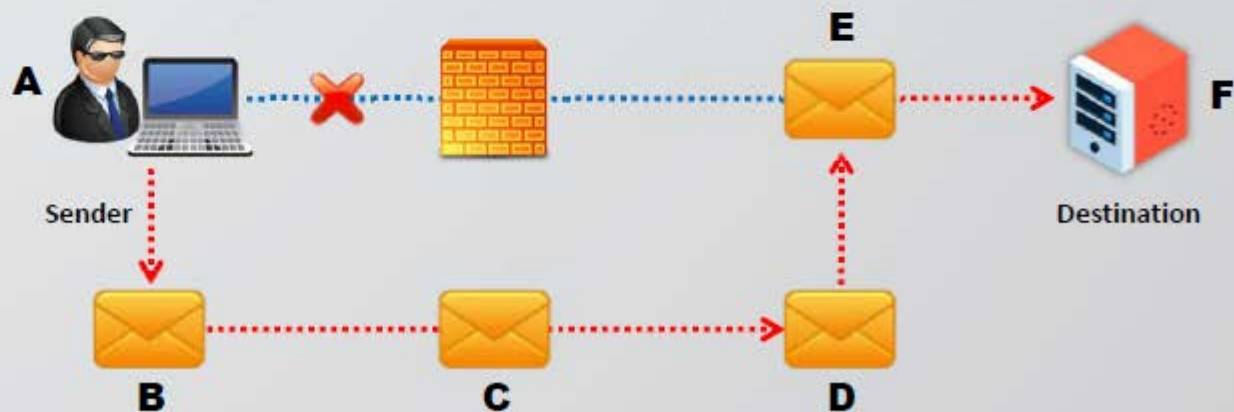
As the packet travels through the nodes in the network, each **router examines** the destination IP address and **chooses the next hop** to direct the packet to the destination



In source routing, the **sender** makes some or all of these decisions on the router



The figure shows source routing, where the originator dictates eventual route of traffic



# Tiny Fragments

01

Attackers create **tiny fragments** of outgoing packets forcing some of the TCP packet's header information into the next fragment

03

The attack will succeed if the **filtering router examines only the first fragment** and allow all the other fragments to pass through

02

The IDS filter rules that specify **patterns will not match** with the fragmented packets due to broken header information

04

This attack is used to **avoid user defined filtering rules** and works when the **firewall checks only for the TCP header information**

IP-3ar0JI0B0K				MK=1, Fragment Offset=0			
Source Port				Destination Port			
Sequence Number							
Acknowledgement Sequence Number							
Data Offset	Reserved	-	ACK	-	-	-	Window
Checksum						Urgent Pointer=0	
0							



# Bypass Blocked Sites Using IP Address in Place of URL

01

This method involves typing the **IP address** directly in browser's address bar in place of typing the **blocked website's domain name**

02

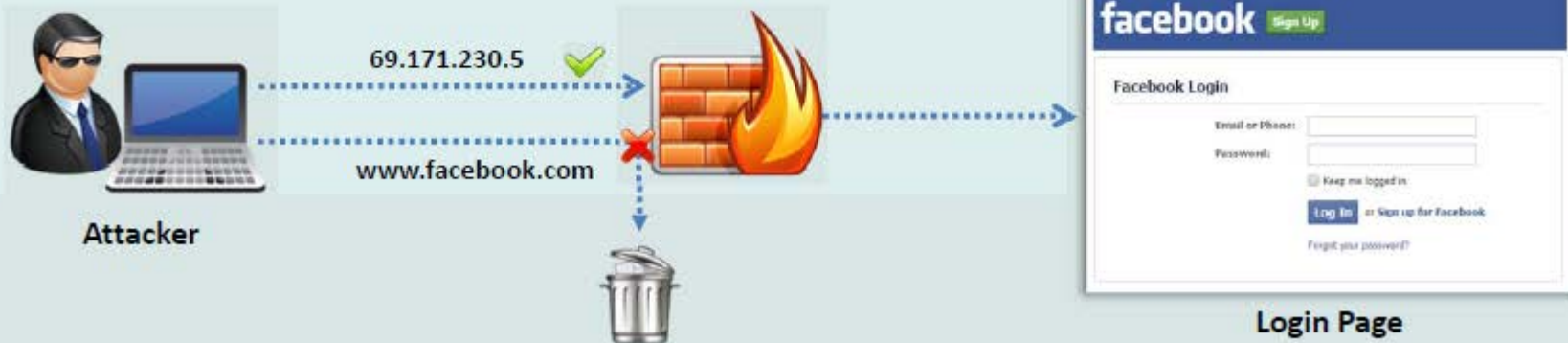
For example, to access Orkut, type its **IP address** instead of typing domain name

03

Use services such as **Host2ip** to find the IP address of the blocked website

04

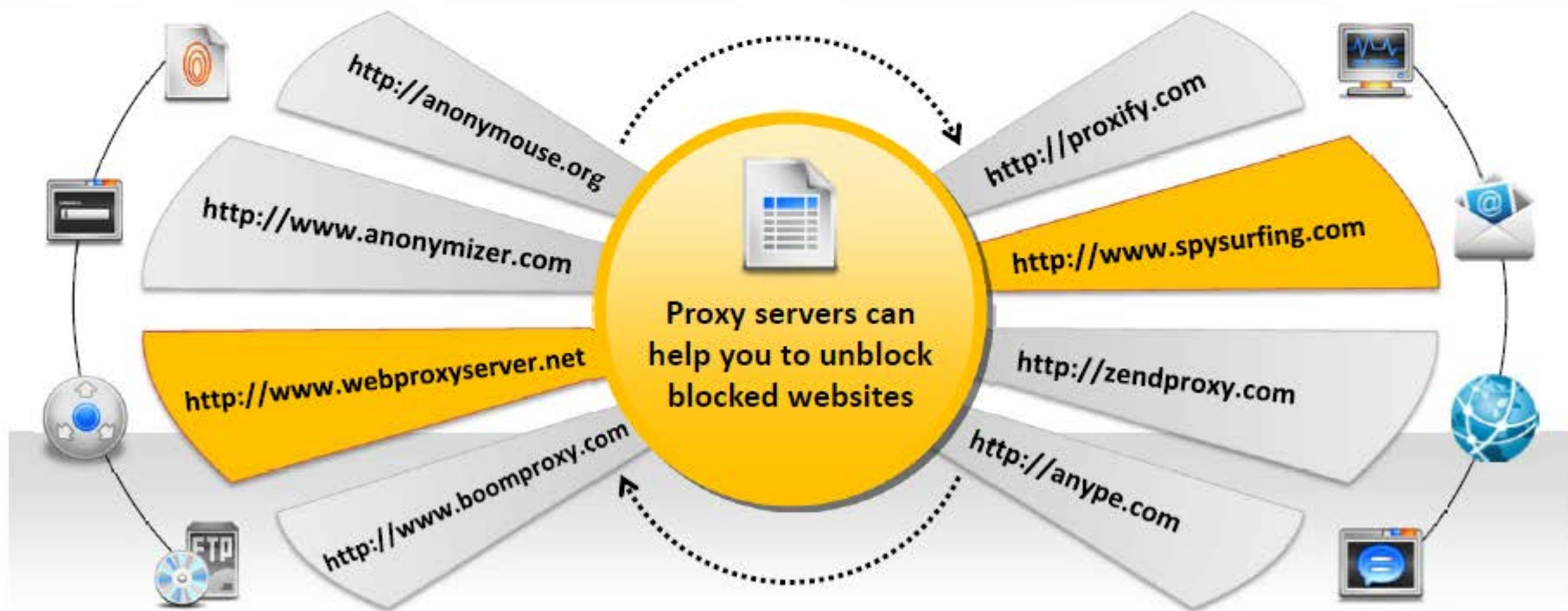
This method fails if the blocking software **tracks the IP address** sent to the web server



# Bypass Blocked Sites Using Anonymous Website Surfing Sites

- Many websites around the net enable **surfing the Internet** anonymously
- Some websites provide options to **encrypt the URL's** of the websites

- These proxy websites will **hide the actual IP address** and will show another IP address, which could **prevent the website from being blocked** thus allowing access to them





# Bypass a Firewall Using Proxy Server



Find an appropriate **proxy server**



On the **Tools** menu of any **Internet browser**, go to LAN of Network Connections tab, and then click LAN/Network Settings



Under **Proxy server settings**, select the use a proxy server for LAN



In the Address box, **type the IP address** of the proxy server

In the Port box, type the port number that is used by the proxy server for **client connections** (by default, 8080)



Click to select **the bypass proxy server** for local addresses check box if you do not want the proxy server computer to be used when connected to a computer on the local network



Click **OK** to close the **LAN Settings** dialog box



Click **OK** again to close **the Internet Options** dialog box





# Bypassing Firewall through ICMP Tunneling Method



It allows tunneling a **backdoor shell** in the data portion of ICMP Echo packets

RFC 792, which delineates **ICMP operation**, does not define what should go in the data portion

The **payload portion** is arbitrary and is not examined by most of the firewalls, thus any data can be inserted in the payload portion of the ICMP packet, including a **backdoor application**

Some administrators keep **ICMP open** on their firewall because it is useful for tools like **ping** and **traceroute**

Assuming that ICMP is allowed through a firewall, use **Loki ICMP tunneling** to execute commands of choice by tunneling them inside the payload of **ICMP echo packets**



# Bypassing Firewall through ACK Tunneling Method

1

It allows tunneling a backdoor application with TCP packets with the ACK bit set

2

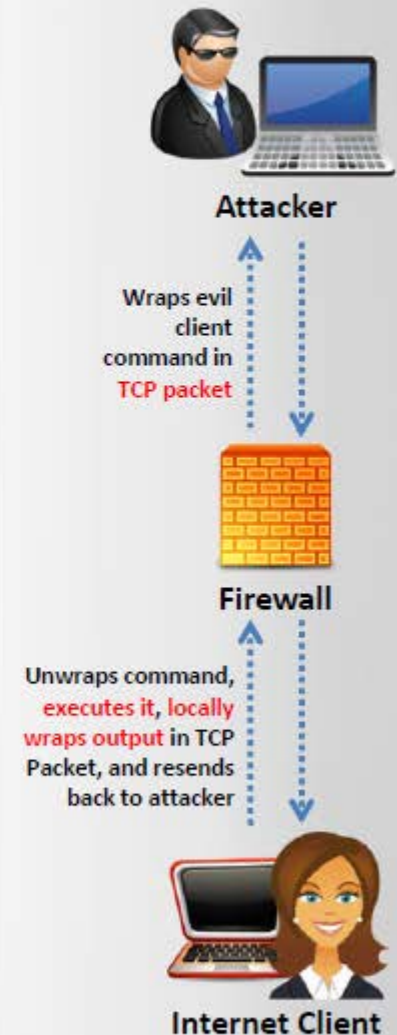
ACK bit is used to acknowledge receipt of a packet

3

Some firewalls do not check packets with ACK bit set because ACK bits are supposed to be used in response to legitimate traffic

4

Tools such as AckCmd (<http://ntsecurity.nu>) can be used to implement ACK tunneling



# Bypassing Firewall through HTTP Tunneling Method

1

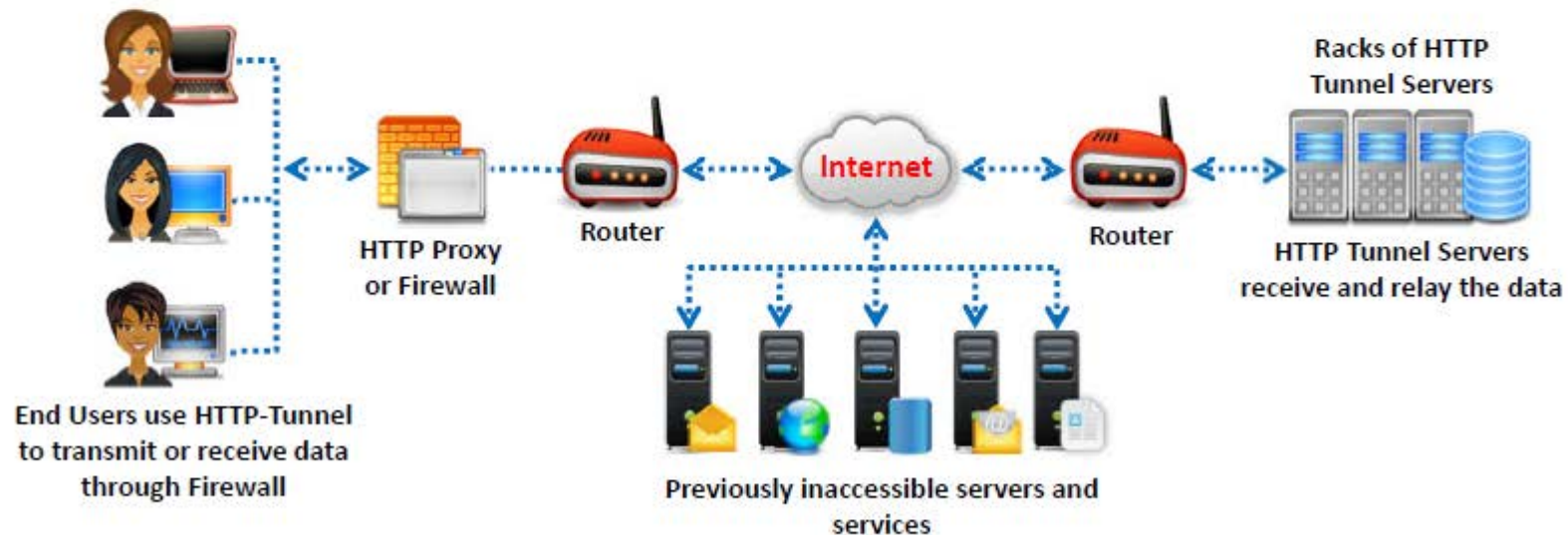
HTTP Tunneling technology allows attackers to **perform various Internet tasks** despite the restrictions imposed by firewalls

2

This method can be implemented if the target company has a **public web server with port 80** used for HTTP traffic, that is unfiltered on its firewall

3

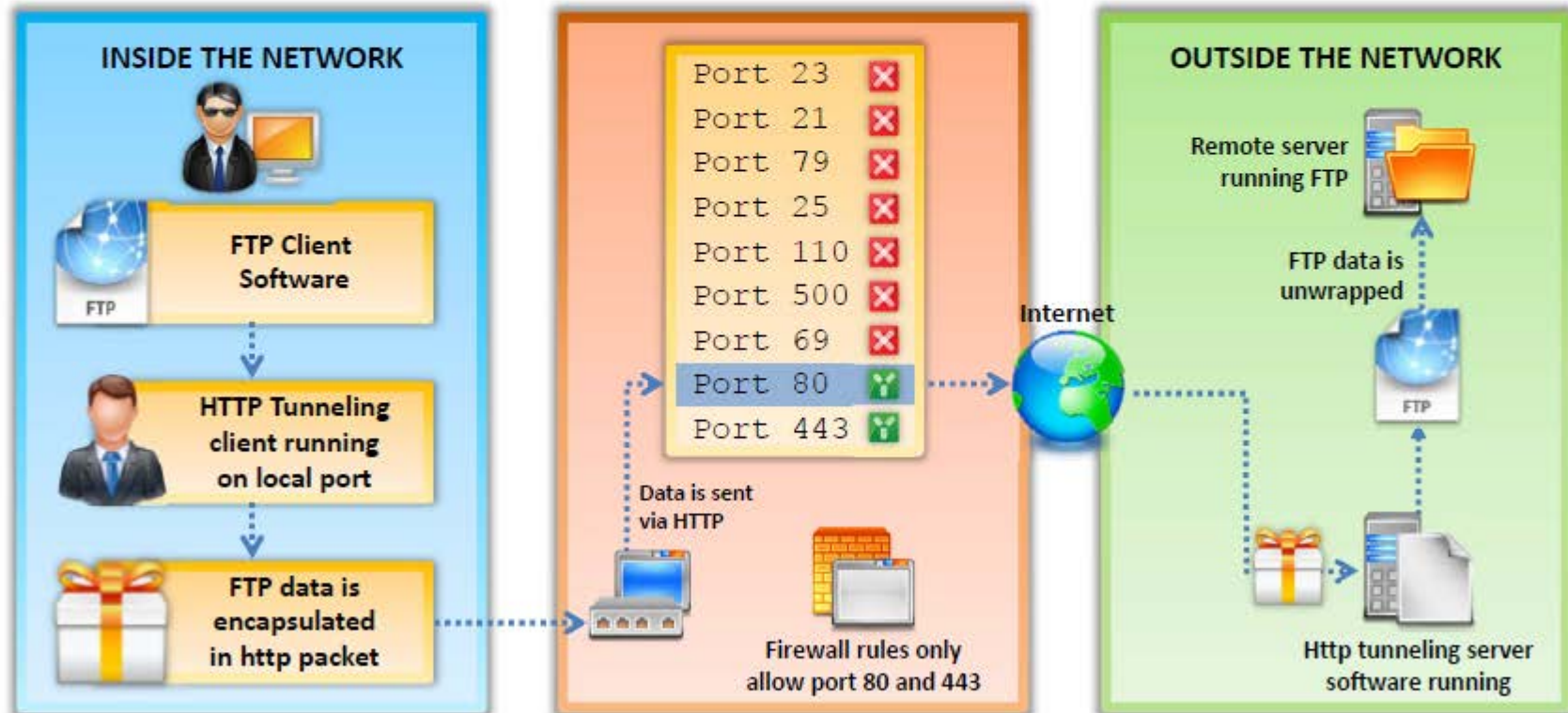
Encapsulates data inside HTTP traffic (port 80)





# Why do I Need HTTP Tunneling

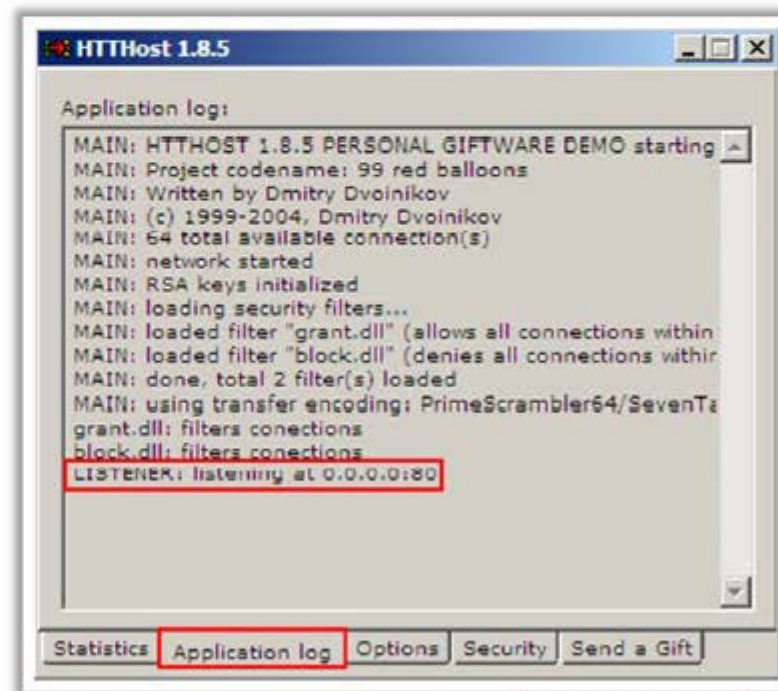
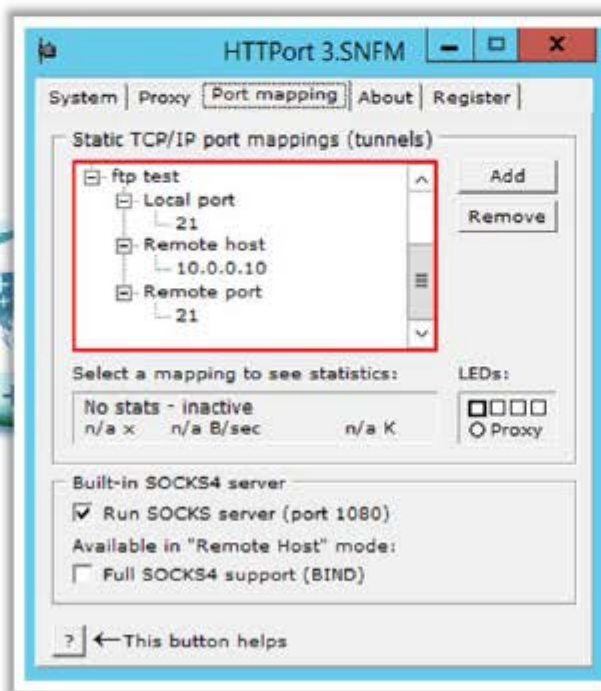
- Organizations firewall all ports except **80** and **443**, and you may want to use FTP
- HTTP tunneling will enable use of **FTP via HTTP protocol**



# HTTP Tunneling Tools: HTTPort and HTTPHost



- HTTPort allows you to **bypass your HTTP proxy**, which is blocking you from the Internet
- It allows you to use various **Internet software from behind the proxy**, ex. e-mail, instant messengers, P2P file sharing, ICQ, News, FTP, IRC, etc.



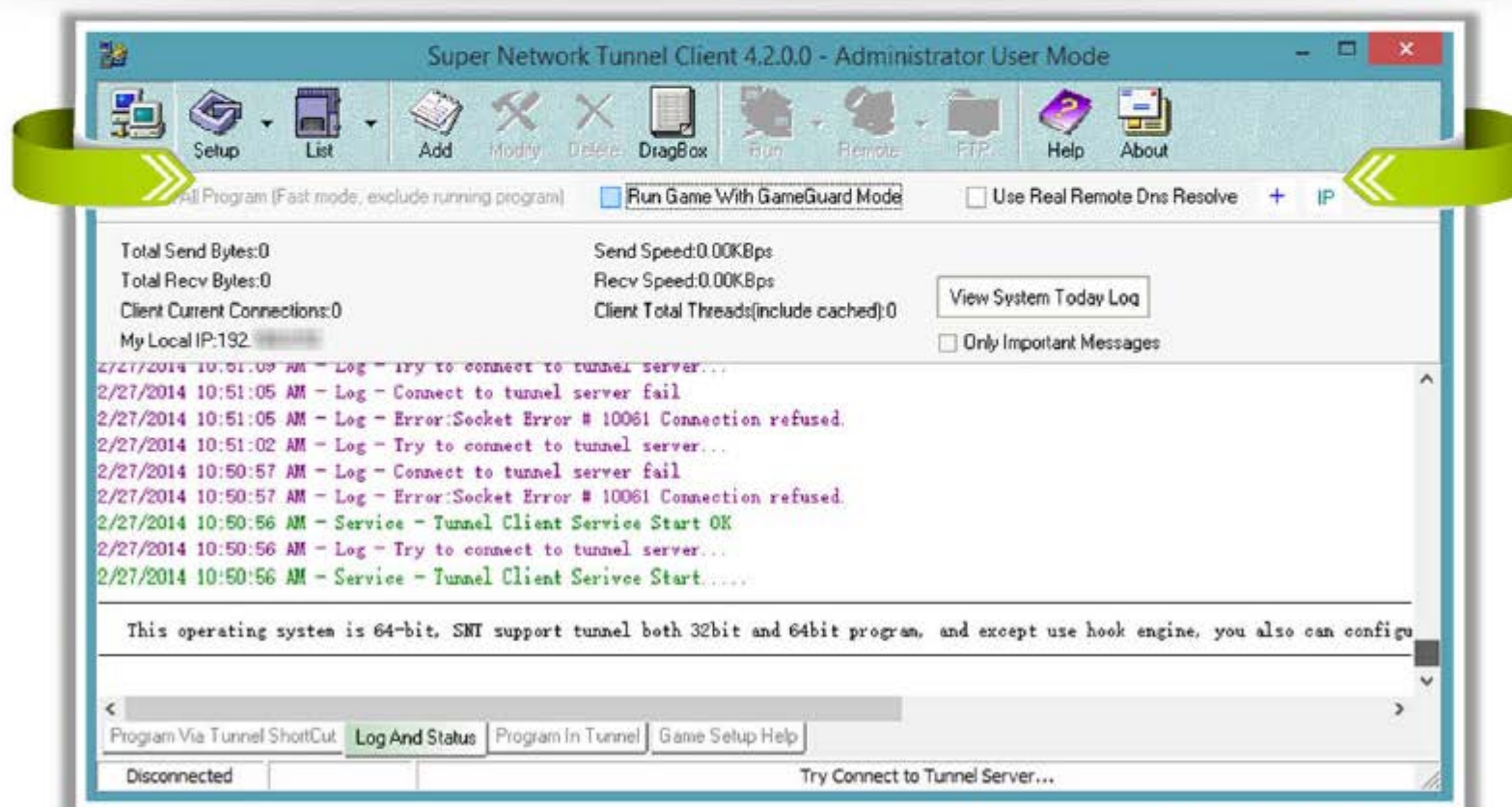
<http://www.targeted.org>



# HTTP Tunneling Tool: Super Network Tunnel

CEH  
Certified Ethical Hacker

- A **two-way http tunnel** software connecting two computers
- Works like **VPN tunneling** but uses HTTP protocol to establish a connection



<http://www.networktunnel.net>

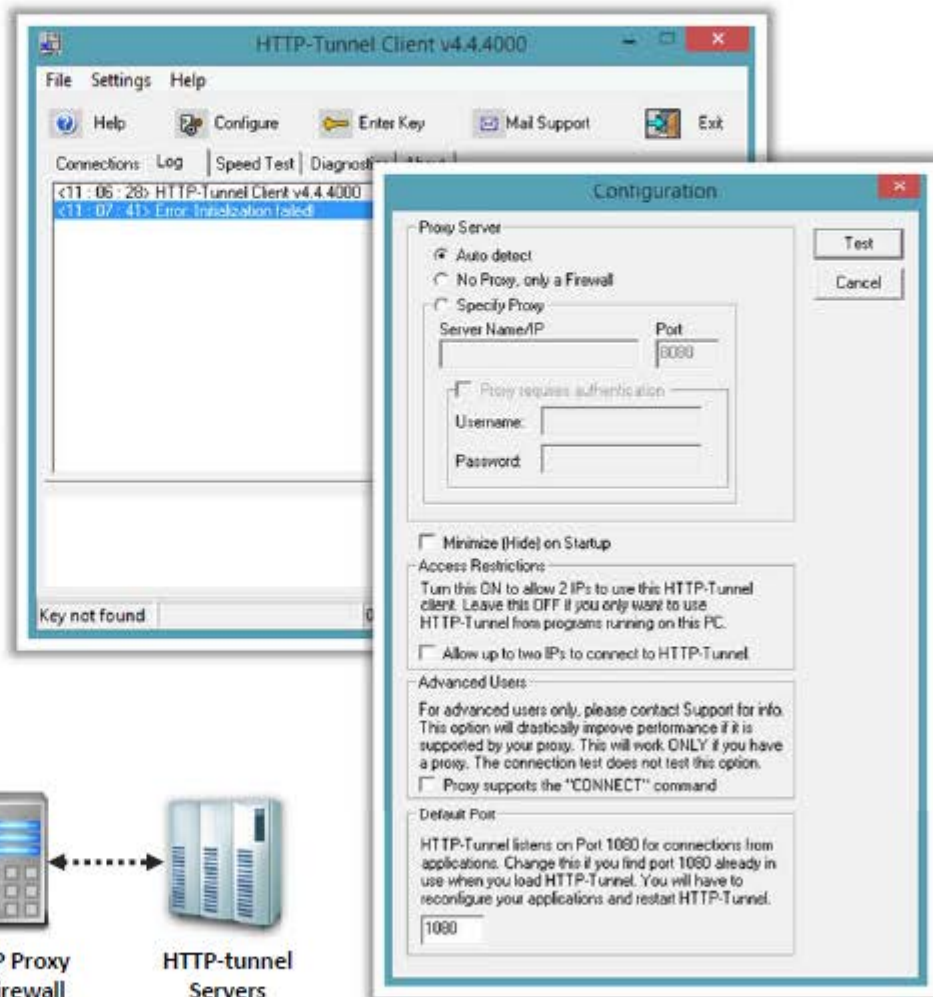


# HTTP Tunneling Tool: HTTP-Tunnel

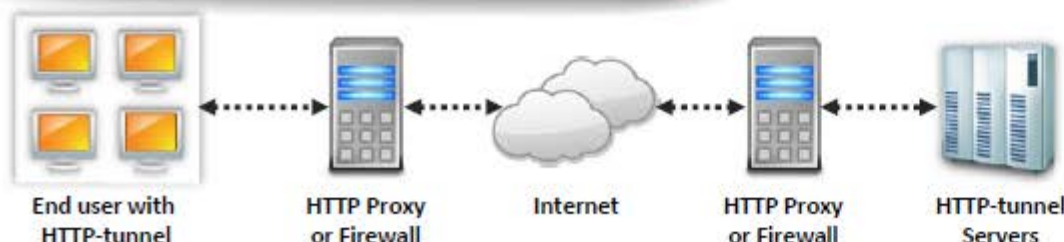


HTTP-Tunnel acts as a **socks server**, allowing you to use your Internet applications safely despite **restrictive firewalls**

SOCKet Secure (SOCKS) is an Internet protocol that **routes network packets** between a client and server through a proxy server



<http://www.http-tunnel.com>

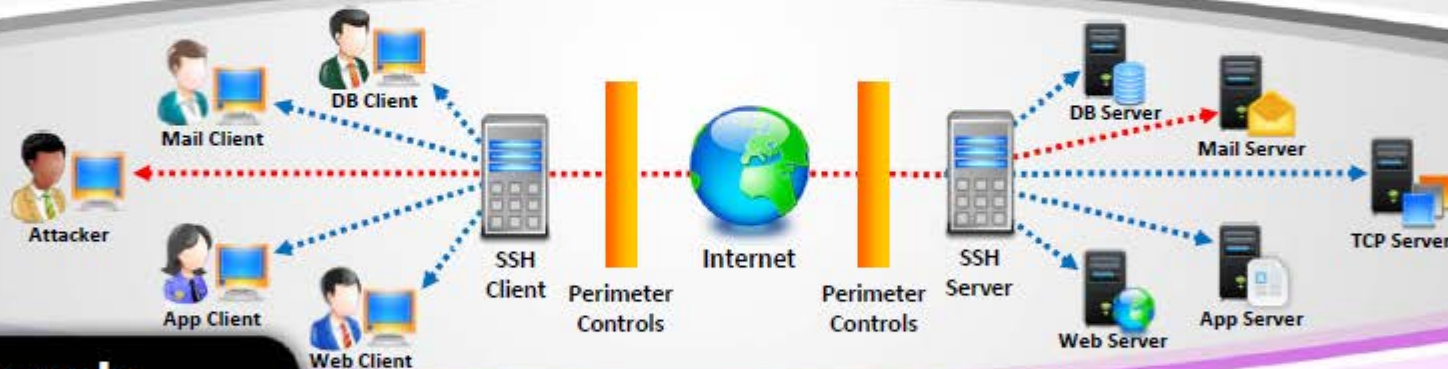


# Bypassing Firewall through SSH Tunneling Method



## OpenSSH

Attackers use OpenSSH to **encrypt and tunnel** all the traffic from a local machine to a remote machine to avoid detection by perimeter security controls



## Example

```
ssh -f user@certifiedhacker.com -L 5000:certifiedhacker.com:25 -N
```

`-f` => background mode, `user@certifiedhacker.com` => user name and server you are logging into, `-L 5000:certifiedhacker.com:25` => local-port:host:remote-port, and `-N` => Do not execute the command on the remote system

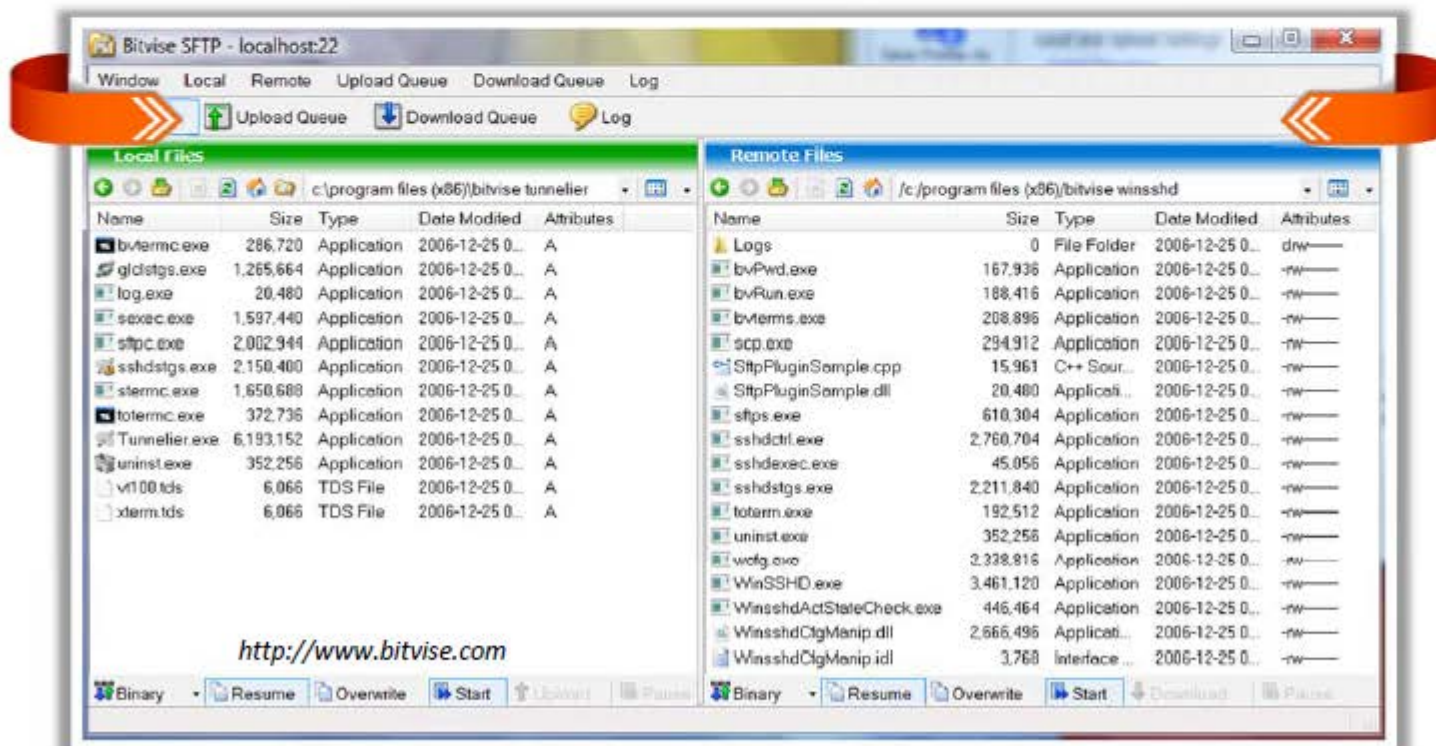
- ❖ This forwards the **local port 5000** to **port 25** on `certifiedhacker.com` encrypted
- ❖ Simply point your email client to use `localhost:5000` as the SMTP server



# SSH Tunneling Tool: Bitvise

**CEH**  
Certified Ethical Hacker

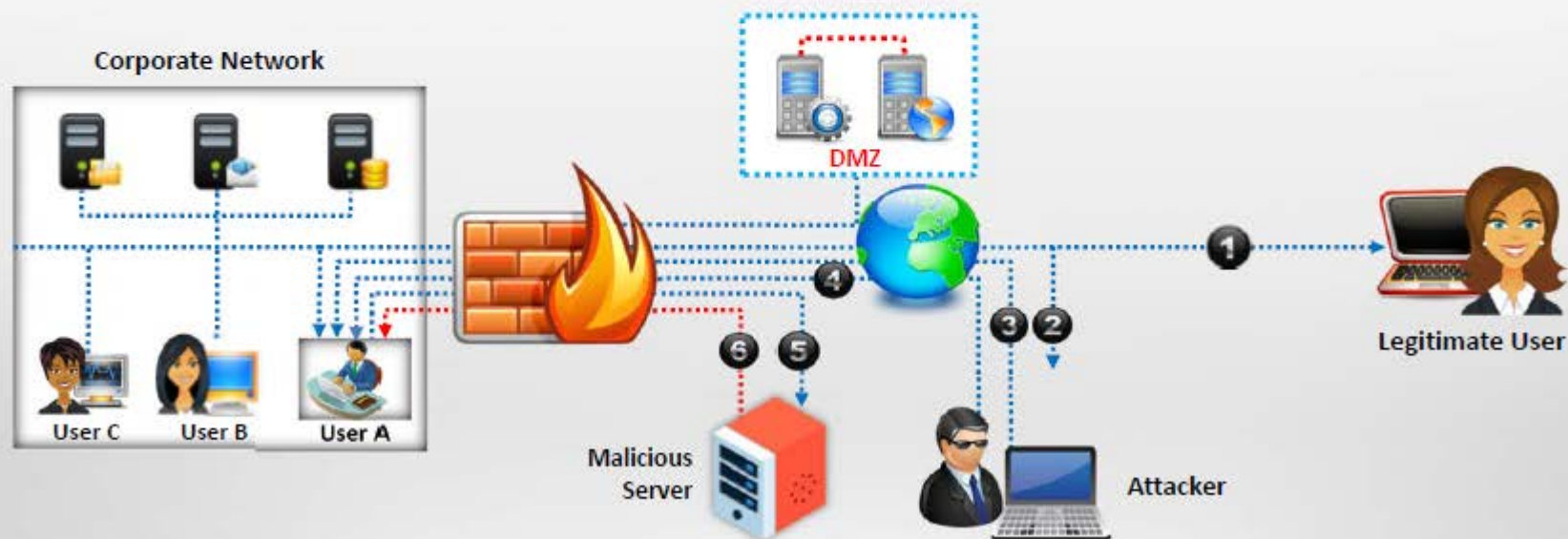
- Bitvise SSH Server provides secure **remote login capabilities** to Windows workstations and servers
- SSH Client includes powerful tunneling features including **dynamic port forwarding** through an integrated proxy, and also **remote administration** for the SSH Server





# Bypassing Firewall through External Systems

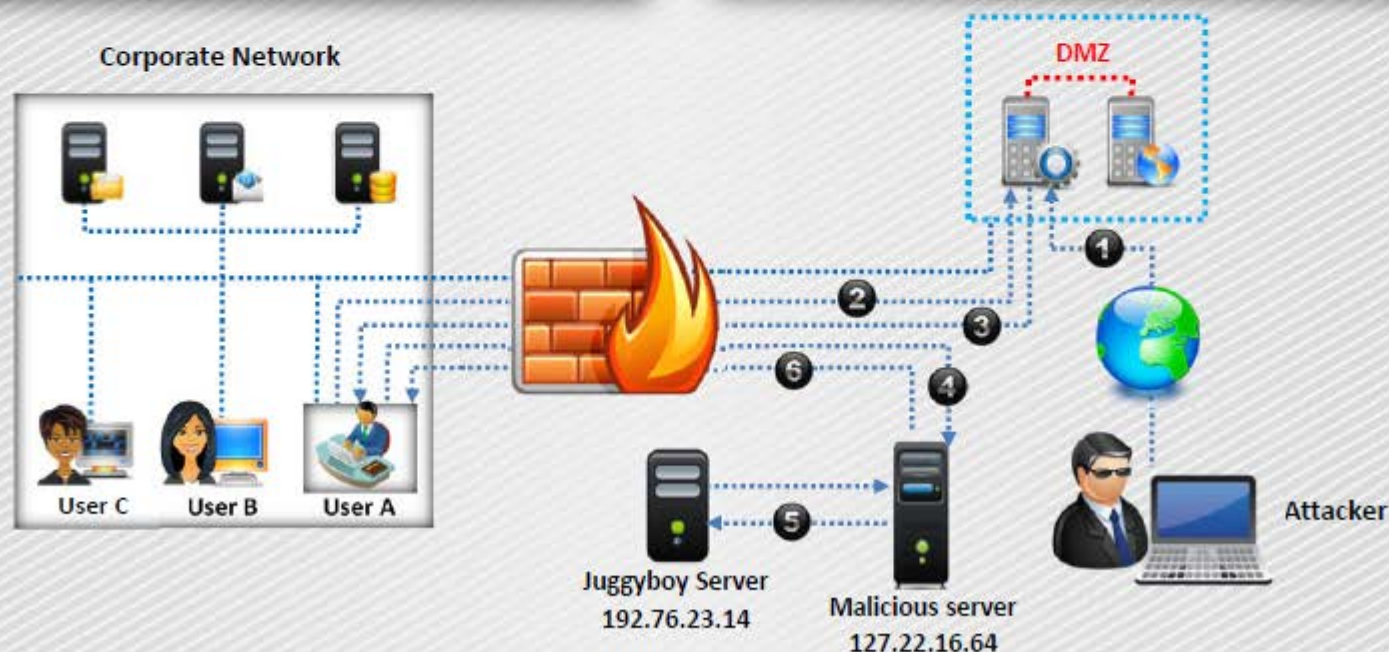
1. Legitimate user works with some **external system** to access the corporate network
2. Attacker sniffs the **user traffic**, steals the **session ID** and **cookies**
3. Attacker **accesses the corporate network** bypassing the firewall and gets **Windows ID** of the running Netscape 4.x/ Mozilla process on user's system
4. Attacker then issues an **openURL()** command to the found window
5. User's web browser is redirected to the **attacker's Web server**
6. The malicious codes embedded in the attacker's web page are **downloaded and executed** on the user's machine



# Bypassing Firewall through MITM Attack

1. Attacker performs **DNS server poisoning**
2. User A requests for `WWW.juggyboy.com` to the **corporate DNS server**
3. Corporate DNS server sends the **IP address (127.22.16.64) of the attacker**

4. User A accesses the **attacker's malicious server**
5. Attacker connects with the **real host and tunnels the user's HTTP traffic**
6. The malicious codes embedded in the attacker's web page are **downloaded and executed** on the user's machine





# Bypassing Firewall through Content



In this method, the attacker **sends the content containing malicious code** to the user and tricks him/her to open it so that the malicious code can be executed



## Examples:

Sending an email containing malicious executable file or Microsoft office document capable of exploiting **macro bypass exploit**



There are many file formats that can be used as **malicious content carrier**



# Module Flow

01

**IDS, Firewall  
and Honeypot  
Concepts**

02

**IDS, Firewall  
and Honeypot  
Solutions**

03

**Evading IDS**

04

**Evading  
Firewalls**



05

**IDS/Firewall  
Evading Tools**

06

**Detecting  
Honeypots**

07

**IDS/Firewall  
Evasion Counter-  
measures**

08

**Penetration  
Testing**

# IDS/Firewall Evasion Tool: Traffic IQ Professional



Traffic IQ Professional enables security professionals to **audit and validate the behavior of security devices** by generating the **standard application traffic or attack traffic** between two virtual machines

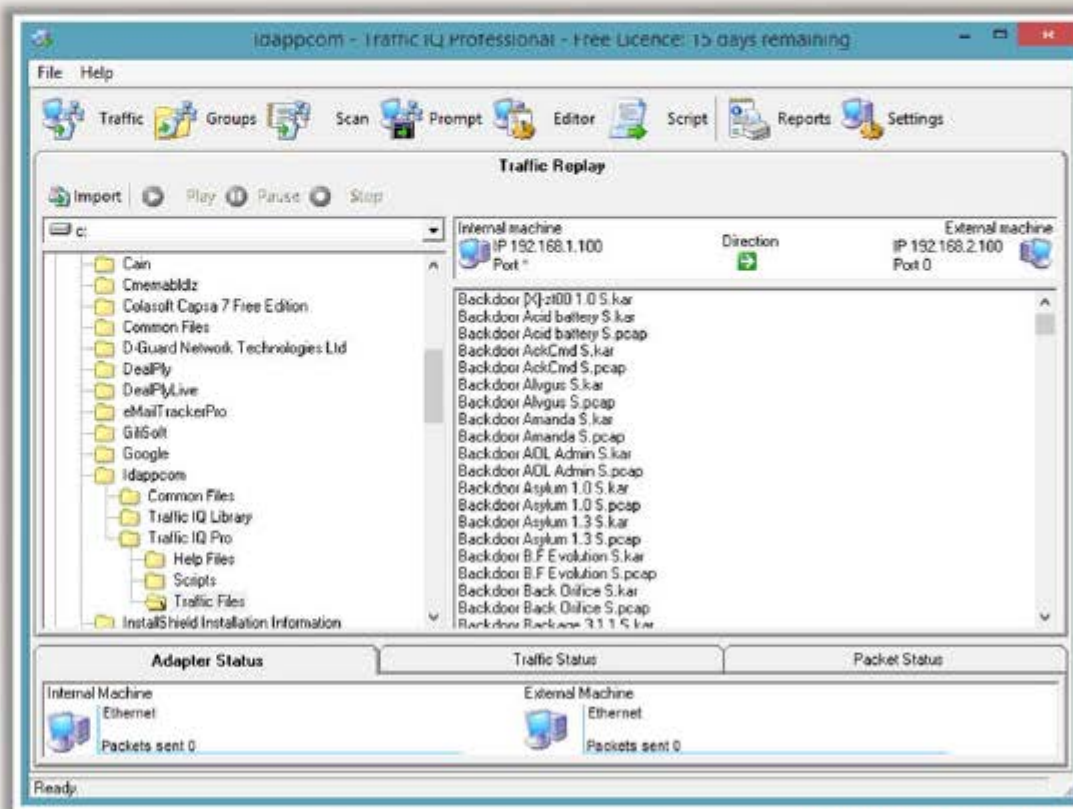
Traffic IQ Professional can be used to **assess, audit, and test the behavioral characteristics** of any non-proxy packet-filtering device including:

01 Application firewall systems

02 Intrusion detection systems

03 Intrusion prevention systems

04 Routers and switches



<http://www.idappcom.com>

# IDS/Firewall Evasion Tool:

## tcp-over-dns



01

tcp-over-dns contains a special **dns server** and a special **dns client**

02

The client and server work in tandem to provide a **TCP (and UDP!)** tunnel through the standard DNS protocol



### Command Prompt

```
C:\Users\P\Desktop\tcp-over-dns-1.3>java -jar tcp-over-dns-server.jar --domain
test123.test.com --forward-port 808 --forward-address 192.168.168.2 --mtu=400
--log-level 3
000000.0 main: tcp-over-dns-server starting up
000000.0 main: Hosting domain: test123.test.com
000000.0 main: DNS listening on: /0.0.0.0:53
000000.0 main: Forwarding to: /192.168.168.2:808
000000.0 main: MTU: 400
000000.0 main: Log level: 3
045533.5 DNS Serve /0.0.0.0:53: New tcp client connection:254
045636.6 Client timeout: Client timeout (ClientID:254).
045636.6 TCP comm /192.168.168.2:2037: Local TCP socket closed.
```

<http://analogbit.com>



# IDS/Firewall Evasion Tools



**Snare Agent for Windows**

<http://www.intersectalliance.com>



**Freenet**

<https://freenetproject.org>



**AckCmd**

<http://ntsecurity.nu>



**GTunnel**

<http://gardennetworks.org>



**Tomahawk**

<http://tomahawk.sourceforge.net>



**Hotspot Shield**

<http://www.anchorfree.com>



**Your Freedom**

<http://www.your-freedom.net>



**Proxifier**

<http://www.proxifier.com>



**Atelier Web Firewall Tester**

<http://www.atelierweb.com>



**Vpn One Click**

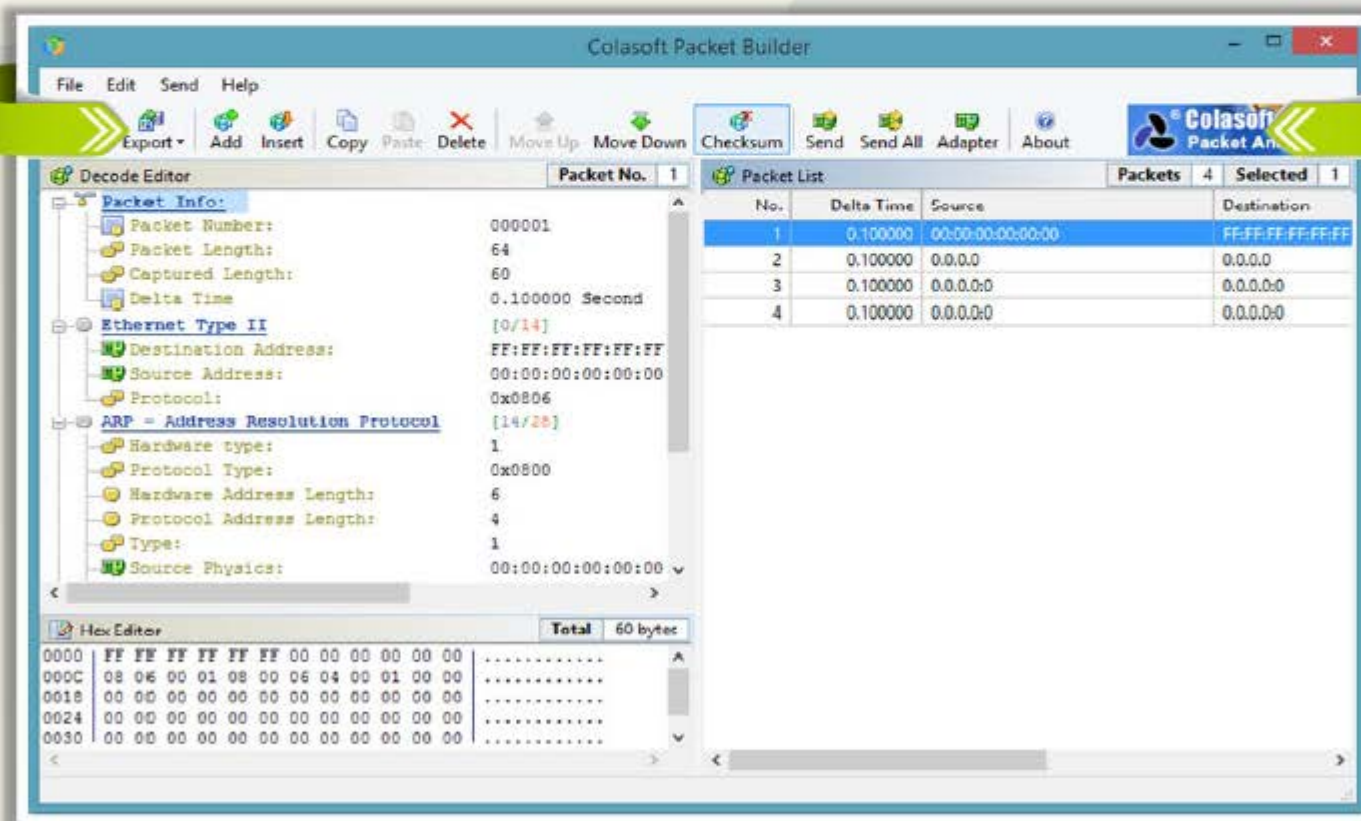
<http://www.vpnoneclick.com>

# Packet Fragment Generator: Colasoft Packet Builder



**Colasoft  
Packet Builder**

Colasoft packet builder is a network packet crafter, packet generator or packet editor that network professionals use to **build (or craft)** all types of custom network



<http://www.colasoft.com>

# Packet Fragment Generators



**CommView**

<http://www.tamos.com>



**fping 3**

<http://fping.org>



**hping3**

<http://www.hping.org>



**NetScanTools Pro**

<http://www.netscantools.com>



**Multi-Generator (MGEN)**

<http://cs.itd.nrl.navy.mil>



**pktgen**

<http://www.linuxfoundation.org>



**Net-Inspect**

<http://search.cpan.org>



**PACKETH**

<http://packeth.sourceforge.net>



**Ostinato**

<https://code.google.com>



**Packet Generator**

<http://www.tamos.com>



# Module Flow

01

**IDS, Firewall  
and Honeypot  
Concepts**

02

**IDS, Firewall  
and Honeypot  
Solutions**

03

**Evading IDS**

04

**Evading  
Firewalls**



05

**IDS/Firewall  
Evading Tools**

06

**Detecting  
Honeypots**

07

**IDS/Firewall  
Evasion Counter-  
measures**

08

**Penetration  
Testing**

# Detecting Honeypots

1

Attackers can determine the **presence of honeypots** by probing the services running on the system



2

Attackers craft **malicious probe packets** to scan for services such as HTTP over SSL (HTTPS), SMTP over SSL (SMTPS), and IMAP over SSL (IMAPS)



3

Ports that show a particular service running but deny a **three-way handshake connection** indicate the presence of a honeypot



4

**Tools to probe honeypots:**

- Send-safe Honeypot Hunter
- Nessus
- Hping



**Note:** Attackers can also defeat the purpose of honeypots by using multi-proxies (TORs) and hiding their conversation using encryption and steganography techniques

# Honeypot Detection Tool: **Send-Safe Honey**pot Hunter



Send-Safe Honeypot Hunter is a tool designed for checking **lists of HTTPS and SOCKS proxies** for "honey pots"

## Features:

01

Checks lists of **HTTPS, SOCKS4, and SOCKS5 proxies** with any ports



02

Checks **several remote or local proxylists** at once



03

Can upload **"Valid proxies"** and **"All except honeypots"** files to FTP



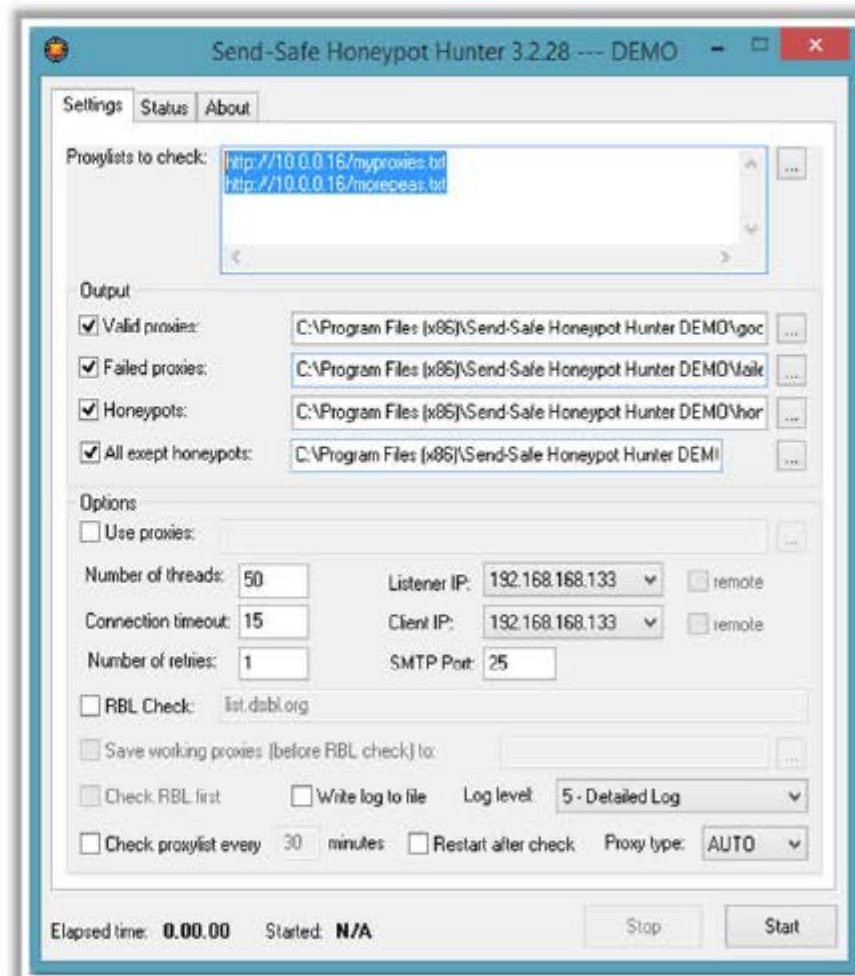
04

Can process **proxylists** automatically every specified period of time



05

May be used for **usual proxylist** validating as well



<http://www.send-safe.com>



# Module Flow

01

**IDS, Firewall  
and Honeypot  
Concepts**

02

**IDS, Firewall  
and Honeypot  
Solutions**

03

**Evading IDS**

04

**Evading  
Firewalls**



05

**IDS/Firewall  
Evading Tools**

06

**Detecting  
Honeypots**

07

**IDS/Firewall  
Evasion Counter-  
measures**

08

**Penetration  
Testing**

# Countermeasures



**Shut down switch ports** associated with the known attack hosts

Perform an **in-depth analysis** of ambiguous network traffic for all possible threats



**Reset (RST)** malicious TCP sessions

Look for the **nop opcode** other than 0x90 to defend against the polymorphic shellcode problem



Train users to identify attack patterns and **regularly update/patch** all the systems and network devices

Deploy IDS after a **thorough analysis** of network topology, nature of network traffic, and the number of host to monitor



# Countermeasures

(Cont'd)



Use a **traffic normalizer** to remove potential ambiguity from the packet stream before it reaches to the IDS

Ensure that IDSs **normalize fragmented packets** and allow those packets to be reassembled in the proper order



**Define DNS server** for client resolver in routers or similar network devices

**Harden the security** of all communication devices such as modems, routers, switches, etc.



If possible, block **ICMP TTL expired** packets at the external interface level and change the **TTL field to a large value**, ensuring that the end host always receives the packets



# Module Flow

01

**IDS, Firewall  
and Honeypot  
Concepts**

02

**IDS, Firewall  
and Honeypot  
Solutions**

03

**Evading IDS**

04

**Evading  
Firewalls**



05

**IDS/Firewall  
Evading Tools**

06

**Detecting  
Honeypots**

07

**IDS/Firewall  
Evasion Counter-  
measures**

08

**Penetration  
Testing**

# Firewall/IDS Penetration Testing



Firewall/IDS penetration testing helps in evaluating the Firewall and IDS for **ingress** and **egress** traffic filtering capabilities

## Why Firewall/IDS Pen Testing?



To check if firewall/IDS properly enforces an **organization's firewall/IDS policy**

To check the **amount of network information accessible** to an intruder



To check if the **IDS and firewalls** enforces organization's network security policies

To check the firewall/IDS for **potential breaches of security** that can be exploited



To check if the firewall/IDS is good enough to **prevent the external attacks**

To evaluate the **correspondence of firewall/IDS rules** with respect to the actions performed by them



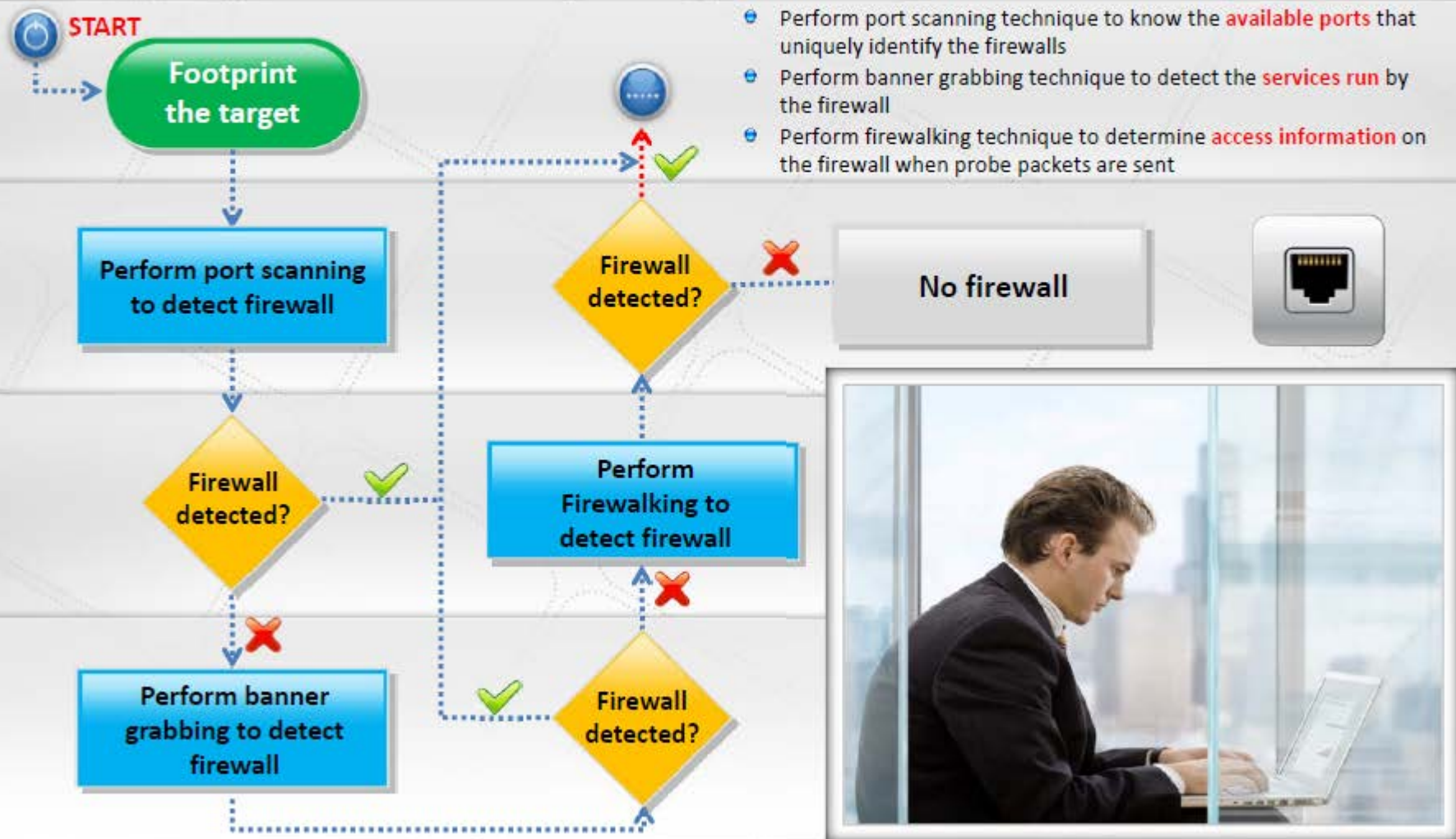
To check the effectiveness of the **network's security perimeter**

To verify whether the **security policy is correctly enforced** by a sequence of firewall/IDS rules or not



# Firewall Penetration Testing

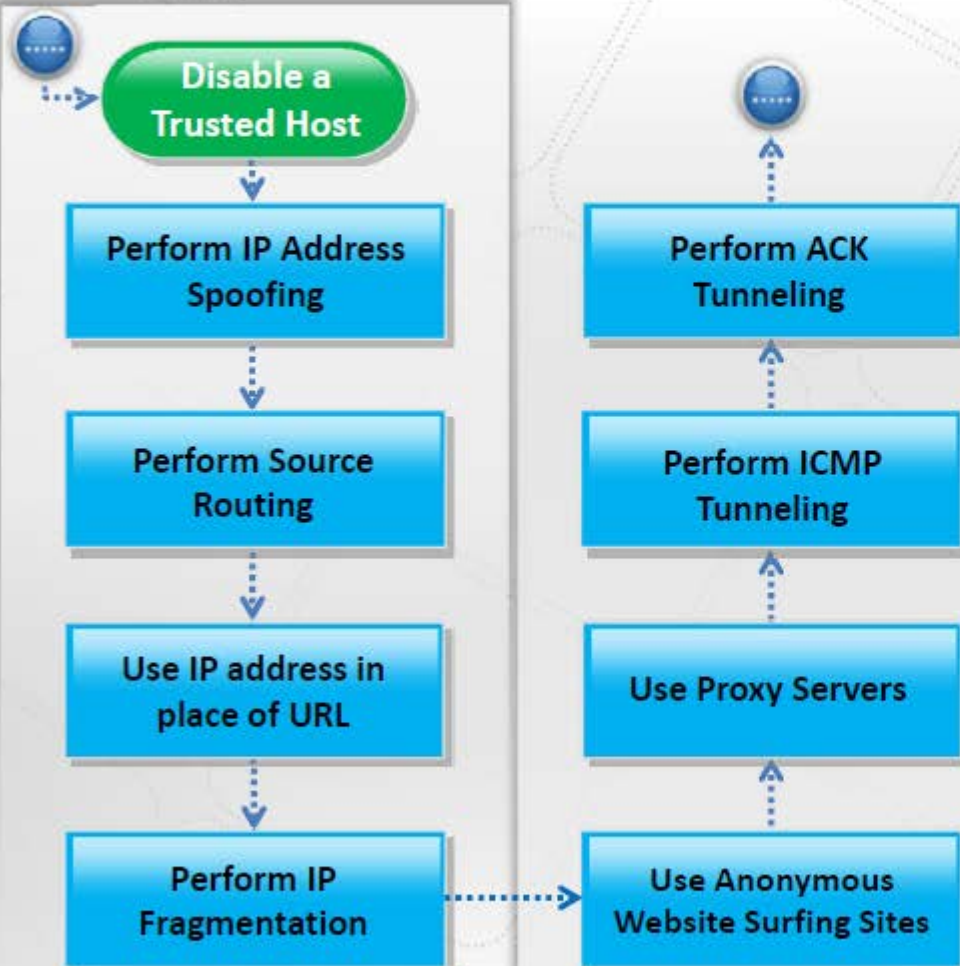
CEH  
Certified Ethical Hacker





# Firewall Penetration Testing

(Cont'd)



- Perform **IP address spoofing** to gain unauthorized access to a computer or a network
- Perform **fragmentation attack** to force the TCP header information into the next fragment in order to bypass the firewall
- Use **proxy servers** that block the actual IP address and display another thereby allowing **access to the blocked website**
- Perform **ICMP tunneling** to tunnel a backdoor application in the data portion of ICMP Echo packets
- Perform **ACK tunneling** using tools such as **AckCmd** to tunnel backdoor application with TCP packets with the ACK bit set

# Firewall Penetration Testing

(Cont'd)

**CEH**  
Certified Ethical Hacker



Perform HTTP Tunneling

- Perform HTTP tunneling using tools such as **HTTPPort**, **HTTHost**, **Super Network Tunnel**, **HTTP-Tunnel**, etc. to tunnel the traffic across TCP port 80
- Perform SSH tunneling using tools such as **Bitvise** to encrypt and tunnel all the traffic from a local machine to a remote machine

Perform SSH Tunneling

- Gain **access to the corporate network** by sniffing the user's traffic and stealing the session ID and cookies
- Perform MITM attack in order to **own corporate DNS server** or to spoof DNS replies to it

Use External Systems

Perform MITM Attack

Document All the Findings





# IDS Penetration Testing



- Perform **obfuscating technique** to encode attack packets that IDS would not detect but an IIS web server would decode and become attacked
- Try to bypass IDS by hiding attack traffic in a large volume of **false positive alerts** (false positive generation attack)
- Use **session splicing technique** to bypass IDS by keeping the session active for a longer time than the IDS reassembly time
- Try **Unicode representations** of characters to evade the IDS signature
- Perform **fragmentation attack with** IDS fragmentation reassembly timeout **less and more than** that of the Victim

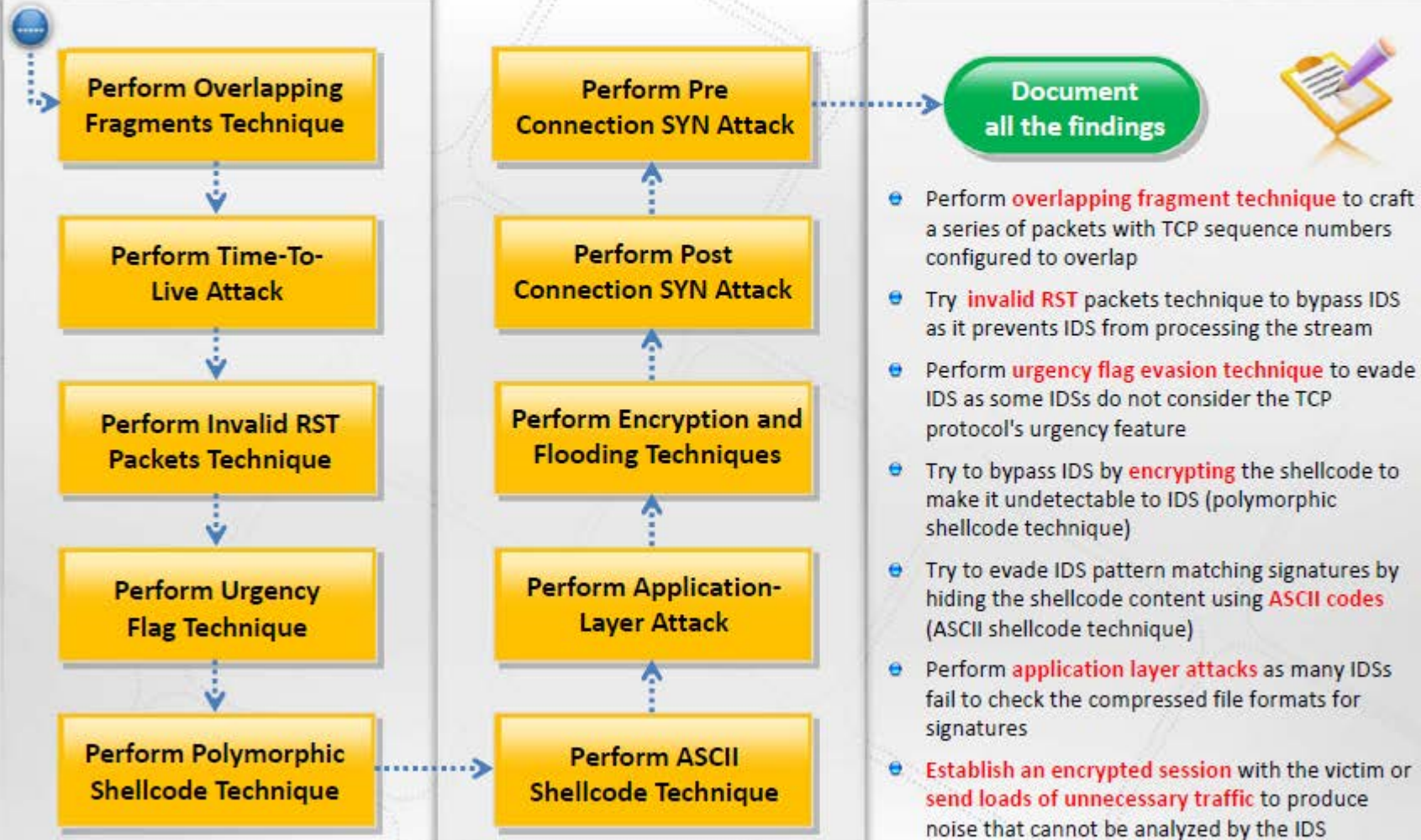




# IDS Penetration Testing

(Cont'd)

CEH  
Certified Ethical Hacker



# Module Summary



- ☐ An intrusion detection system (IDS) inspects all inbound and outbound network traffic for suspicious patterns that may indicate a network or system security breach
- ☐ Network-based intrusion detection systems typically consist of a black box that is placed on the network in the promiscuous mode, listening for patterns indicative of an intrusion
- ☐ Host-based intrusion detection systems usually include auditing for events that occur on a specific host
- ☐ Firewalls are software or hardware-based system designed to prevent unauthorized access to or from a private network
- ☐ A honeypot is an information system resource that is expressly set up to attract and trap people who attempt to penetrate an organization's network
- ☐ Firewall is identified by three techniques namely port scanning, banner grabbing, and firewalking
- ☐ Attackers can determine the presence of honeypots by probing the services running on the system
- ☐ Firewall/IDS penetration testing helps in evaluating the Firewall and IDS for ingress and egress traffic filtering capabilities