Proxies | IDS

INGI2347: COMPUTER SYSTEM SECURITY (Spring 2014)

Marco Canini

UCL
Université catholique de Louvain

Lecture slides adapted from UCL INGI2347 by Gildas Avoine and Upenn CSE331 by Steve Zdancewic
Reproduced with permission

Plan for today

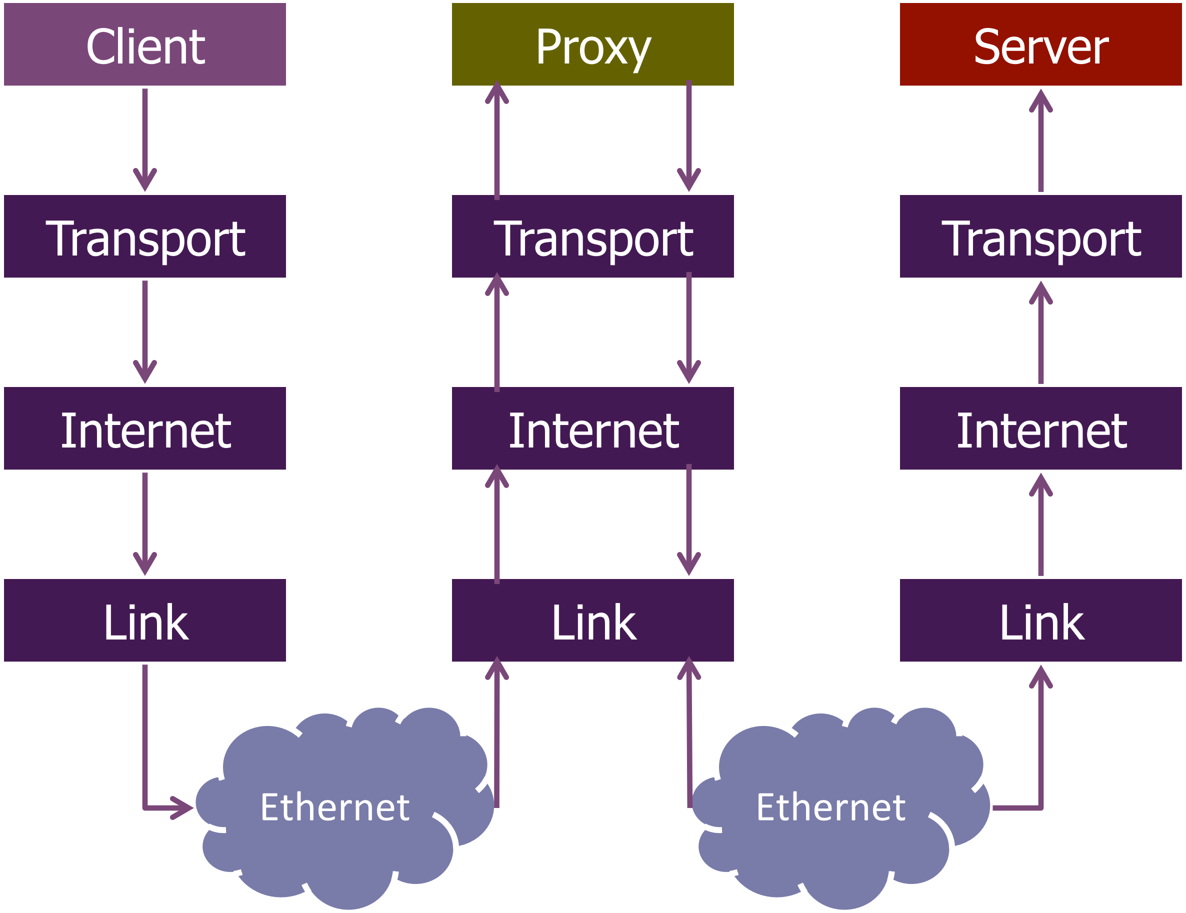
Lecture 5

- Proxies

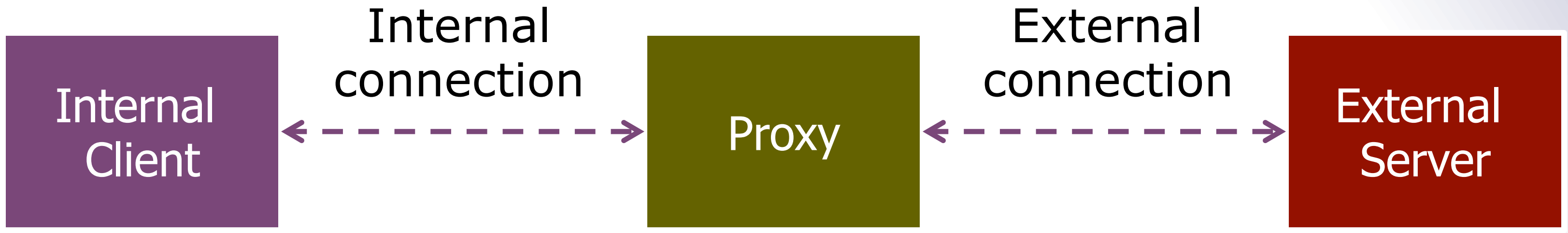
  NEXT

  - Proxies features and benefits
  - Types of proxies
  - Reverse proxies
- Intrusion Detection Systems

Proxies are application relays



Proxy



- Proxy acts like both a client and a server
- Can provide other services too
  - Examples: caching, load balancing, mobile page transformation, content transcoding/compression/translation
- A typical example of the defense in depth and choke point principles

Proxy Benefits

- Prevent direct connections from the internal network towards the Internet
  - Choke point
  - Possibly authentication
- Able to filter application-level info
  - URL or DNS blacklists, URL filtering
  - Content type (MIME) filtering, keyword filtering
  - Virus, exploit, …

Cache Feature

- The proxy can keep a copy of all the contents it has served in a cache
- When another client asks for the same content, it can provide the cached copy
  - Ensure content is up-to-date (Example: in HTTP, use header info)
- The transfer is much faster (increase in QoE)
- We can save on bandwidth (limit cost)

HTTP Without Proxy

```
$ telnet www.example.com 80
GET /index.html HTTP/1.0

HTTP/1.1 200 OK
Date: Mon, 10 Feb 2014 15:21:38 GMT
Server: Apache/2.2.3 (CentOS)
Expires: Mon, 10 Feb 2014 15:21:38 GMT
Cache-Control: no-cache
Pragma: no-cache
Connection: close
Content-Type: text/html;charset=UTF-8

<html>
<head><title>Example</title></head>
<body>…
```

HTTP With Proxy

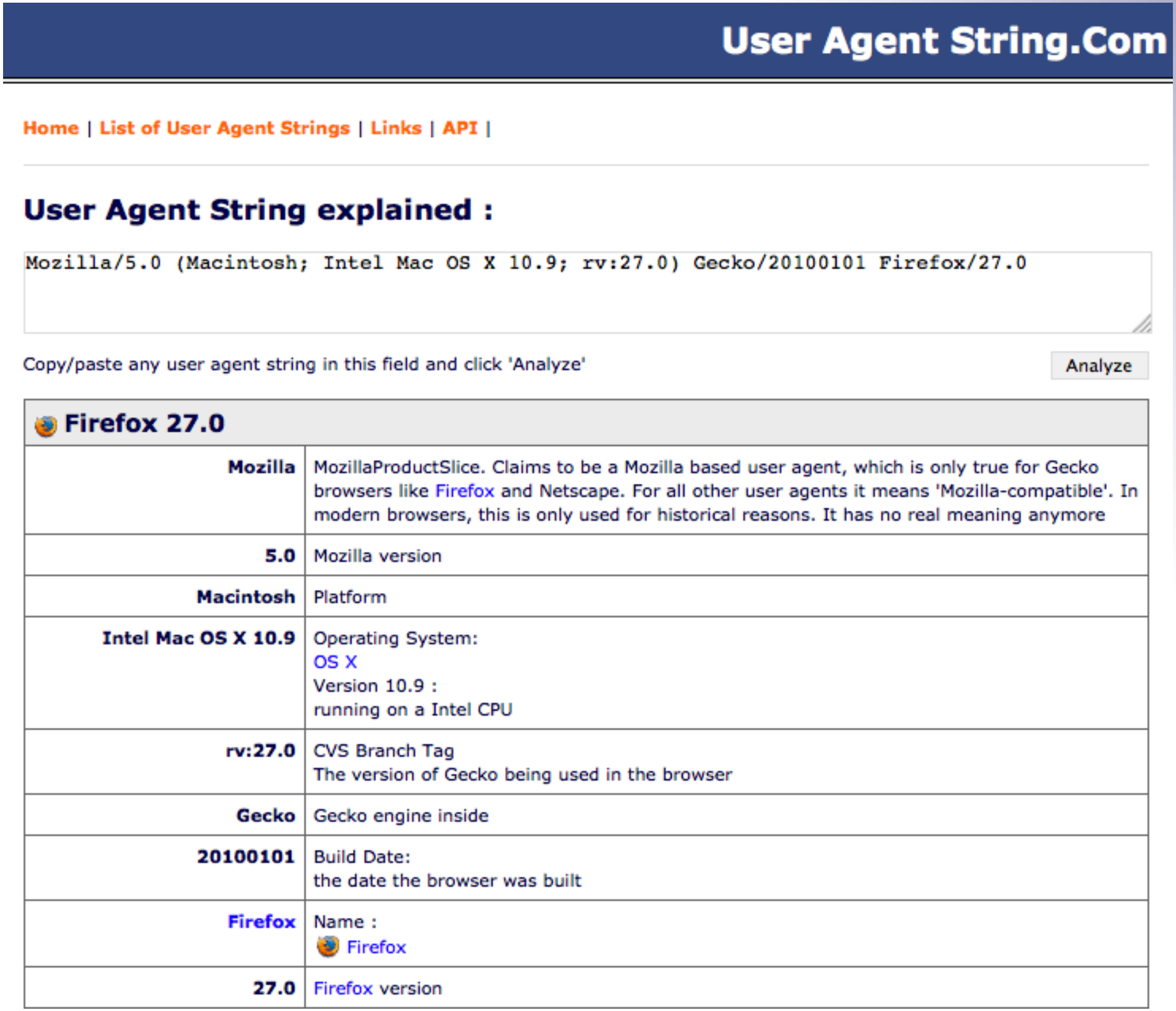
```
$ telnet www.example.com 80
GET http://www.example.com/index.html HTTP/1.0

HTTP/1.1 200 OK
Date: Mon, 10 Feb 2014 15:21:38 GMT
Server: Apache/2.2.3 (CentOS)
Expires: Mon, 10 Feb 2014 15:21:38 GMT
Cache-Control: no-cache
Pragma: no-cache
Connection: close
Content-Type: text/html;charset=UTF-8

<html>
<head><title>Example</title></head>
<body>…
```

- Requires browser configuration!

Anonymity?

- Surf anonymously?

- IP address of proxy is visible
- HTTP request headers are visible
- Traffic can be analyzed by the proxy operator



User Agent String explained :

```
Mozilla/5.0 (Macintosh; Intel Mac OS X 10.9; rv:27.0) Gecko/20100101 Firefox/27.0
```

Copy/paste any user agent string in this field and click 'Analyze'

Firefox 27.0

Mozilla	MozillaProductSlice. Claims to be a Mozilla based user agent, which is only true for Gecko browsers like Firefox and Netscape. For all other user agents it means 'Mozilla-compatible'. In modern browsers, this is only used for historical reasons. It has no real meaning anymore
5.0	Mozilla version
Macintosh	Platform
Intel Mac OS X 10.9	Operating System: OS X Version 10.9 : running on a Intel CPU
rv:27.0	CVS Branch Tag The version of Gecko being used in the browser
Gecko	Gecko engine inside
20100101	Build Date: the date the browser was built
Firefox	Name : Firefox
27.0	Firefox version

+

Types of Proxies

Transparent Proxy (intercepting proxy)

- Traffic targeted at a certain port is automatically redirected towards the proxy by the firewall
- Pros
  - Avoid having to configure browsers
  - Enforce usage of the proxy
  - Enable load balancing
- Cons
  - Doesn’t work for servers that are not on the configured port

Transparent Proxy (intercepting proxy)

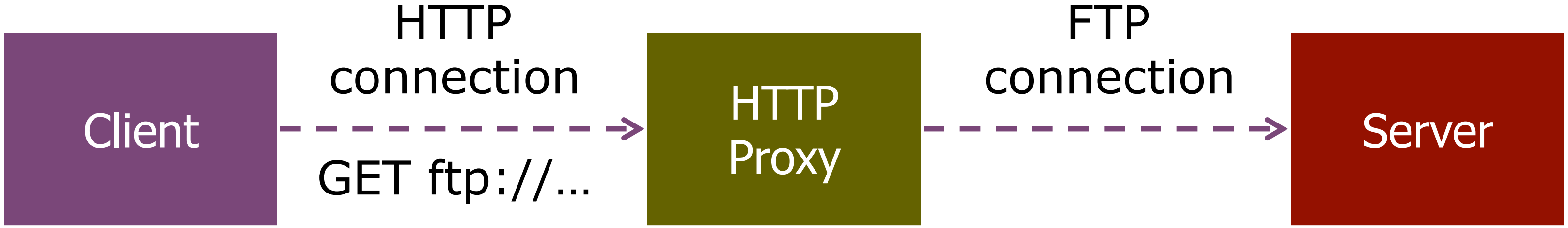
- “A 'transparent proxy' is a proxy that does not modify the request or response beyond what is required for proxy authentication and identification.” [RFC2616]
- Detection of the use of a proxy:
  - Compare IP address of client with that observed by the server
    - (if client is not NATed)
  - Examine HTTP request headers at the server
  - Make a connection to an IP address at which there is no server

FTP Proxy

- FTP summary
  - FTP uses a command connection and a data connection
  - The data connection can be directed towards the client (active mode, default setting) or towards the server (passive mode)
- The FTP protocol has not been designed to be used through a proxy

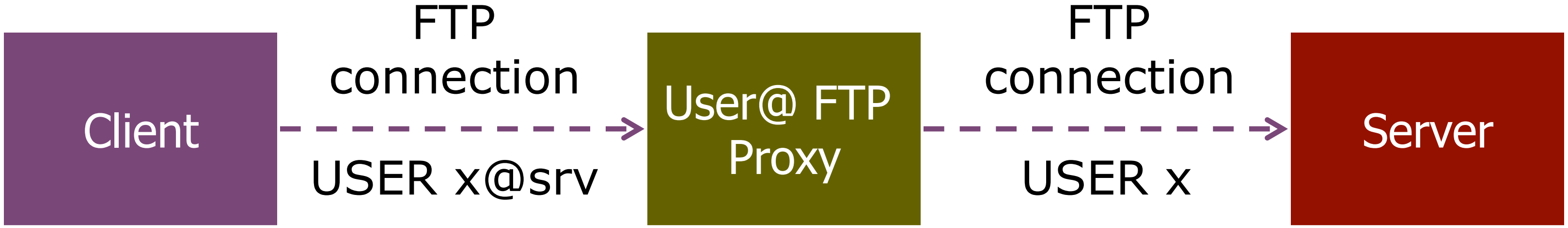
FTP Proxy using HTTP

- Browsers allow specifying URLs in the form ftp://example.com/filename
- If the browser is configured to use an HTTP proxy, it will ask the proxy for that URL
- The HTTP proxy carries out the FTP transfer and provides the document as part of the HTTP reply



User@ FTP Proxy

- User@ proxy behaves like a standard FTP server
- Access server through the proxy by connecting with USER user@server as username to the proxy
- The latter connects to the server and relays the password, commands and the data
- The two connections can use active or passive mode independently



SMTP Proxy

- SMTP was conceived for relaying mail hop by hop
- Hence, any SMTP server can work as a proxy
- Outbound (forward path):
  - Use specified as SMTP server for outgoing mail in the mail client
- Inbound (reverse path):
  - Proxy registered in the DNS as the official server for that domain
  - Proxy has to be configured to forward all mail to the internal server

DNS Proxy

- DNS protocol is designed to retransmit requests from one server to another
- DNS servers can work as proxies
- DNS servers have a cache to limit traffic and reduce response times
- It is a good idea to configure a DNS proxy to direct all its request towards a bigger server (e.g., that of an ISP) to take advantage of a bigger cache

SOCKS Proxy

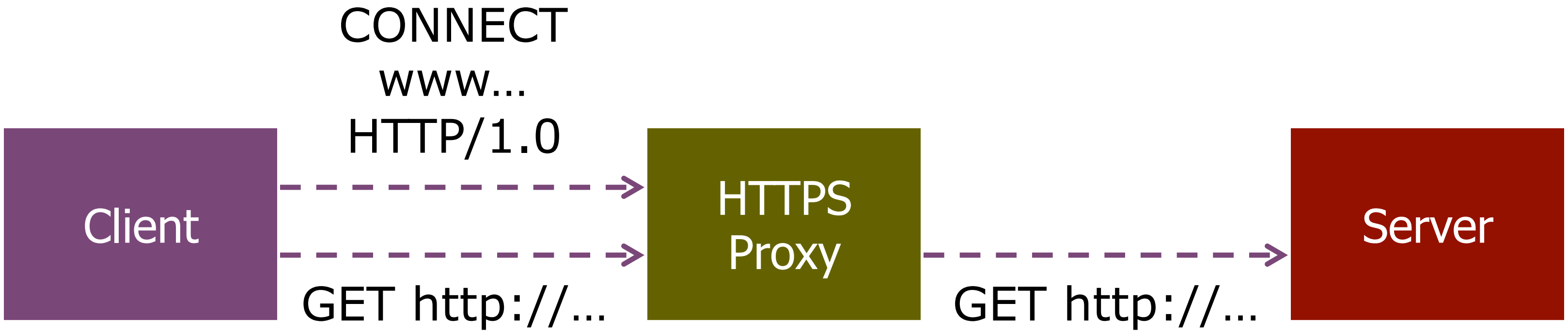
- SOCKS (Socket Server) proxy is a general proxy for TCP (and UDP) connections
- Accept a client connection, then open another one towards the server
- Then relay data between the two connections
- SOCKS allows any protocol to pass via a proxy
  - **Both an advantage and limitation**

HTTPS Proxy

- HTTPS is the secure version of HTTP
  - (encryption, authentication)
- HTTPS proxies are **NOT** a secure version of HTTP proxies!
- HTTPS encrypts and authenticates end-to-end
- HTTPS proxy just transparently relays data between client and server
  - Much like SOCKS

HTTPS Proxy: Implementation

- HTTPS proxy uses the HTTP command CONNECT that indicates the server address
- Replies with a status and becomes transparent



HTTPS Proxy: Security Issues

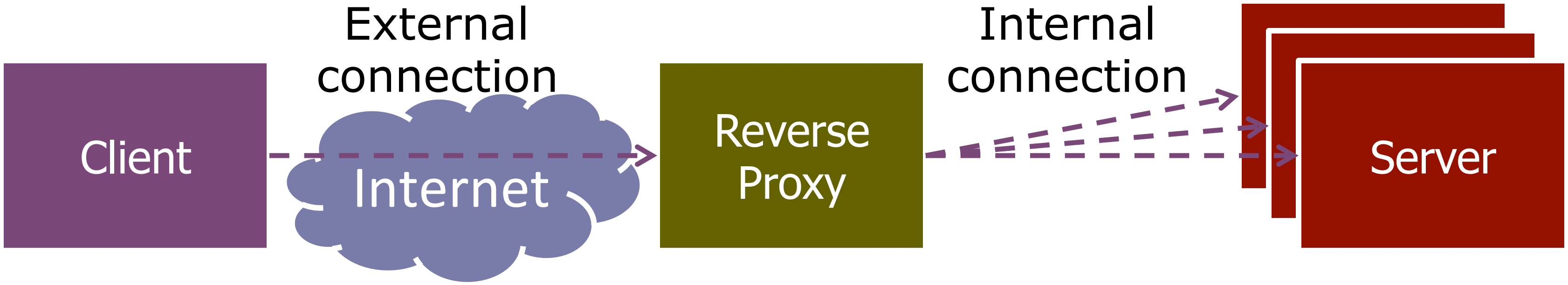
- The HTTPS proxy allows relaying any type of protocol (it is transparent, just like SOCKS)
- To limit abuses, the available ports are often limited to 443 (HTTPS) and 563 (SNEWS)
- To allow any protocol to cross a firewall, it is sufficient to run the server on port 443 and pass through an HTTPS proxy

+

Reverse Proxies

Reverse Proxy

- Appears to clients to be an ordinary server
- Requests are forwarded to one or more origin servers which handle the requests
- Client has no knowledge of the origin servers



Reverse HTTP Proxy

- Filter requests (blocking exploits)
- Authenticate clients even before they communicate with the server
  - Cannot attack the server unless authenticated
- Accelerate servers
  - Encryption acceleration
  - Caching static content
  - Load balancing
- Accelerate clients via content compression

HA
PROXY

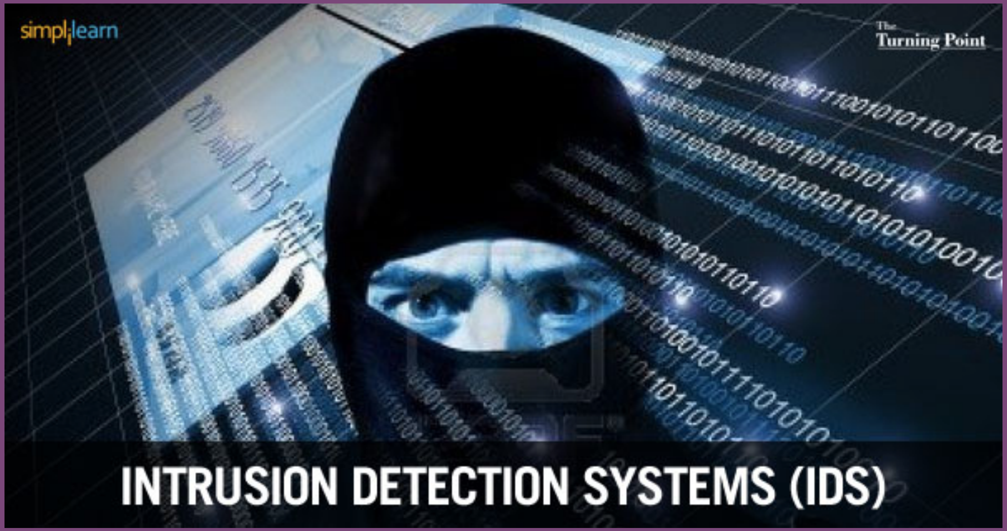
NGINX™

Reverse HTTPS Proxy

- HTTPS proxies are used as encryption accelerators
  - Can reduce the server load by taking care of encryption
  - Can have a hardware accelerator for encrypting data
- The connection between the proxy and the server is HTTP and not HTTPS

Protocol Translation Proxy

- A proxy can use different protocols each side
  - Examples:
  - A Web mail proxy can accept HTTPS requests from the Internet and generate IMAP requests towards the mail server
  - An MPTCP proxy can enable MPTCP-compatible hosts to connect to TCP-only hosts
- Protocol diversity strongly limits the chances of exploiting a vulnerability across a proxy



Intrusion Detection System

Intrusion Detection System (IDS)

Idea: don't wait for the symptoms of an attack before reacting

- An Intrusion Detection System (IDS) monitors
  - Network traffic (Network IDS, NIDS), typically in front of the firewall
  - Events on servers (Host IDS, HIDS)
- When malicious activities are detected, launch an alarm (SMS, mail, etc.)
- Can attempt prevent attacks from succeeding
  - Example: reconfigure firewalls or servers
- Analysis can be done in real-time or by analyzing logs

IDS: Approaches

	Off-line Analysis	Real-time Analysis
Network IDS (NIDS)	Analysis of logs and configuration of firewall, routers	Network sniffer
Host IDS (HIDS)	Examination of system logs	Log/Registry/ Sys-call watcher

IDS with Traffic Characterization

- IDS performs statistical analysis on traffic
  - If a value goes beyond its usual limits then assume there is an attack
- Can recognize new attacks
- May also not recognize them… (false negatives)
- Or see attacks where there aren't (false positives)
- High false positives makes this IDS type unpopular
  - Example: Port Scanning (slow mode to avoid detection)

Signature-based IDS

- Use a database of known attacks
  - Example: Web request with URL of 2000 characters == buffer overflow
- Doesn’t recognize new attacks
  - Must be constantly updated
  - Honeypots: traps to detect attack
- False negatives
  - Manual attacks can have variations that are not detected
  - Signatures are not always precise
- False positives
  - Doesn’t know if an attempted attack was successful
  - Doesn’t know if the target is vulnerable (e.g. Linux attack on Windows server)

Snort: Signature-based

- Lightweight IDS for Linux and Windows
- "Signature, protocol and anomaly based inspection methods"
- Analyze traffic, for example in front of the firewall, to detect possible attacks
- Send mails and/or update filtering rules
- Huge signature database updated by users

Snort: Example of Signatures

```
log tcp any 80 -> any any
```

- Means “Log TCP packets coming from any host, port 80, going to any host, any port”

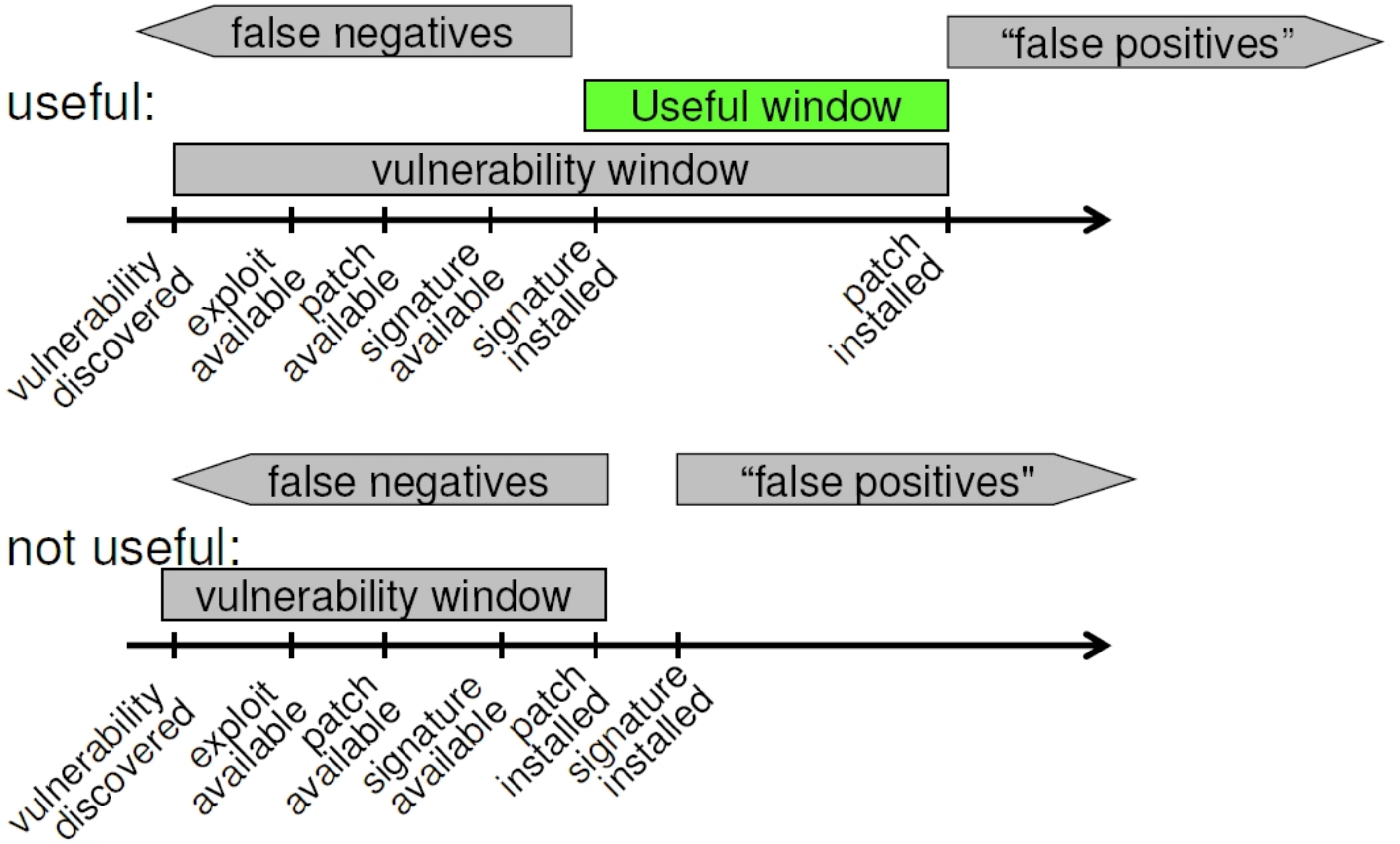
```
alert tcp any any -> 192.168.1.0/24 143
(content: "|90C8 C0FF FFFF|/bin/sh"; msg:
"IMAP buffer overflow!";)
```

- Means “Alert when receiving a packet from any host, any port to port 143 of a computer with IP address in 192.168.1.0/24, when the packet contains the string “|90C8 C0FF FFFF|/bin/sh”

Integrity-based Host IDS: Tripwire

- Typical example of a HIDS with a differed analysis
- Create a digital signature of all files and directories that should not be modified
- Signatures cannot be modified by an attacker
- Regularly compare files and signatures to detect any modification
- Generate an alarm when a modification is detected and can automatically restore the original version of the file

IDS: Efficiency



Intrusion Prevention Systems: IPS

- An IDS that reacts to an attack
  - IP level: Filters the source IP address in the firewall (for a while)
  - TCP level: Sends a spoofed TCP reset packet to the destination to kill the connection
  - Application level: “Corrects” a Web request by removing special characters
- Beware of denial of service through false positives!

IDS: Discussion

- Traffic-characterization IDSes are not yet very efficient
- IDS with signatures work well but:
  - Majority of the attacks for which we have the signature can be blocked by a FW or proxies
  - We should first prevent before trying to detect
- Not sufficient to install an IDS: must also know how to react to attacks and deal with many false positives
- Automatic reactions are usually not advisable (DoS)

IDS: Discussion

- If both traffic characterization and signature-based are possible, it provides a good defense in depth
- IDS deployed in internal networks create less frequent and more critical alarms

Summary

- Proxies
  - Prevent direct connections
  - Application-level filtering
  - Defense in depth and choke point principles
  - Other useful features (e.g., load balancing, caching)
- IDS
  - Monitor network and system activities
  - Raise alarms
  - IPS: react to alarms automatically
  - Challenge: deal with false positives (self DoS)

Any questions?

Stay tuned

+

Next time you will learn about

Cryptography